

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 1

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

2. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;

3. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia(){
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

4. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- Errore a tempo di compilazione
- Errore a tempo di esecuzione
- Programmatore Java con 5 anni di esperienza
- Programmatore Java con 0 anni di esperienza
- Nessuna delle precedenti

5. Date le dichiarazioni:

```

Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- q = (Error []) m;
- q = (Error []) w;
- m = (Object [] []) w;
- m = (Object [] []) q;
- Nessuno dei precedenti

6. Dire quale delle seguenti affermazioni è vera:

- I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- Un attributo **static** non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- Un attributo non può essere contemporaneamente **static** e **final**
- L'operatore **new** non restituisce un riferimento all'oggetto appena creato
- Un oggetto può non esistere dopo la sua dichiarazione

7. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica fl =c1;
        c1.print();
    }
}

```

- Errore a tempo di compilazione
- Errore a tempo di esecuzione
- Cono: 8 - 3 lati per faccia
- Cono: 3 - 3 lati per faccia
- Nessuna delle precedenti

8. Qual è l'output di questo codice?

```

class MyExcl extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {

```

```

        System.out.print(1);
        n();
    }
    catch( MyExc3 s ) {
        System.out.print(2);
        throw( new Exception() );
    }
    finally {
        System.out.print(3);
        throw( new MyExc3() );
    }
}
static void n() throws Exception {
    try {
        System.out.print(4);
        switch(1) {
            case 1: throw( new MyExc2() );
            case 2: throw( new MyExc1() );
            case 3: throw( new MyExc3() );
        }
    }
    catch( MyExc1 t ) {
        throw( new MyExc3() );
    }
    catch( MyExc3 d ) {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception h ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }
}

```

```

public static void main(String [] argv) {
    Sole s = new Sole("bianco");
    System.out.println("Stella: " + s.nome);
    System.out.println("Colore: "
        + s.colore);
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
System.out.println("Pioppo");
int num=n+1;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class Bl {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
System.out.println("Pioppo");
int num=6;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Pioppo
6
 - D. Pioppo
1
 - E. Nessuna delle precedenti
-

15. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 100

- C. 011
 - D. 101
 - E. 010
-

16. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1 = (FiguraGeometrica) f1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Cono: 8 - 3 lati per faccia
 - D. Cono: 3 - 3 lati per faccia
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 2

1. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
System.out.println("Pioppo");
int num=6;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

2. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia(){
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
    }
}
```

```
vinoBianco=spumante;
/*QUI*/
System.out.println("END");
}
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

3. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {
```

```
    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco

E. Nessuna delle precedenti

E. $d = e$;

4. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

5. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. $d = c$;
- B. $d = (\text{Object} [] []) e$;
- C. $c = (\text{String} []) d$;
- D. $d = (\text{Object} [] []) c$;

6. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. $q = (\text{Error} []) m$;
- B. $q = (\text{Error} []) w$;
- C. $m = (\text{Object} [] []) w$;
- D. $m = (\text{Object} [] []) q$;
- E. Nessuno dei precedenti

7. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1 = (FiguraGeometrica) f1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

```

```

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

11. Quale output si ottiene invocando il metodo m?

```

class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
}

```

```

void q(Object a2, String [] a3,
    Boolean b1, Object b2) {
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a3 == a4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
    }
}

```

```

}
catch( MyExc3 d ) {
    System.out.print(5);
    throw( new MyExc2() );
}
catch( Exception h ) {
    System.out.print(6);
    throw( new MyExc2() );
}
}
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

14. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi

- B. Un attributo `static` non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente `static` e `final`
- D. L'operatore `new` non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Cono: 8 - 3 lati per faccia
 - D. Cono: 3 - 3 lati per faccia
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 3

1. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
    super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";
```

```
public Stella(String nome) {
    this.nome = nome;
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

3. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
           Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
```

```

    } else {
        System.out.print(0);
    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

4. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

5. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato

E. Un oggetto può non esistere dopo la sua dichiarazione

6. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {

```

```

        System.out.print(6);
        throw( new MyExc2() );
    }
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

9. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

```

```

Bottiglia() {
    System.out.println(marca);
}

Bottiglia(double litri) {
    this.numlitri=litri;
    System.out.println(litri);
}

public static void main(String args[]) {
    Bottiglia spumante=new Bottiglia();
    Bottiglia vinoRosso=new Bottiglia();
    Bottiglia vinoBianco=new Bottiglia(1.0);
    spumante=null;
    vinoBianco=spumante;
    /*QUI*/
    System.out.println("END");
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```



```
    }  
  }  
}
```

- A. 31
 - B. Errore a tempo di compilazione
 - C. 312Exception in thread main MyExc2
 - D. 312
 - E. Nessuna delle precedenti
-

11. Date le dichiarazioni:

```
Object [] [] m;  
Error [] q;  
Object [] w;  
q = new Error [9];  
m = new Object [5] [2];  
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
 - B. q = (Error []) w;
 - C. m = (Object [] []) w;
 - D. m = (Object [] []) q;
 - E. Nessuno dei precedenti
-

12. Qual è l'output di questo codice?

```
abstract class Programmatore {  
    String nome;  
    String cognome;  
    int anniEsperienza;  
  
    public Programmatore(String n, String c,  
                          int a){  
  
        nome=n;  
        cognome=c;  
        anniEsperienza=a;  
    }  
    abstract void programma();  
}  
public class ProgrammatoreJava  
    extends Programmatore {  
  
    void stampa(){  
  
        System.out.print("Programmatore Java con ");  
        System.out.print(anniEsperienza);  
        System.out.println(" anni di esperienza");  
    }  
  
    ProgrammatoreJava(String n, String c, int a){  
        super(n,c,a);
```

```
    }  
public static void main(String [] argv) {  
    ProgrammatoreJava pj=new  
        ProgrammatoreJava("Franco", "Bianchi",5);  
    pj.stampa();  
}  
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Programmatore Java con 5 anni di esperienza
 - D. Programmatore Java con 0 anni di esperienza
 - E. Nessuna delle precedenti
-

13. Qual è l'output di questo codice?

```
interface FiguraGeometrica {  
    static int dimensioni = 10;  
    void print();  
}
```

```
class Cono implements FiguraGeometrica {  
  
    static final int numLati = 3;  
  
    public void print() {  
        System.out.println("Cono: "  
            + dimensioni + " - " +  
            numLati + " lati per faccia.");  
    }  
  
}
```

```
public static void main(String [] argv) {  
    Cono c1 = new Cono();  
    FiguraGeometrica f1;  
    c1= (FiguraGeometrica) f1;  
    c1.print();  
}  
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Cono: 8 - 3 lati per faccia
 - D. Cono: 3 - 3 lati per faccia
 - E. Nessuna delle precedenti
-

14. Qual è l'output di questo codice?

```
interface FiguraGeometrica {  
    static int dimensioni = 10;  
    void print();  
}
```

```
class Cono implements FiguraGeometrica {
```

```

static final int numLati = 3;

public void print() {
    System.out.println("Cono: "
        + dimensioni + " - " +
        numLati + " lati per faccia.");
}

public static void main(String [] argv) {
    Cono c1 = new Cono();
    FiguraGeometrica fl =c1;
    c1.print();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

15. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");

```

```

        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 4

1. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

2. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

3. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print ();
}

class Cono implements FiguraGeometrica {
    static final int numLati = 3;
```

```
public void print () {
    System.out.println("Cono: "
        + dimensioni + " - " +
        numLati + " lati per faccia.");
}

public static void main(String [] argv) {
    Cono c1 = new Cono();
    FiguraGeometrica f1;
    c1= (FiguraGeometrica) f1;
    c1.print ();
}
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
```

```

try {
    System.out.print(3);
    throw( new Exception() );
}
finally {
    throw( new MyExc3() );
}
}
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

6. Quale output si ottiene invocando il metodo m?

```

class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
          Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

7. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo

D. Pioppo

1

E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

A. Errore a tempo di compilazione

B. Errore a tempo di esecuzione

C. Stella: Sole

Colore: rosa

D. Stella: Sole

Colore: bianco

E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }
}
```

```
public static void main(String [] argv) {
    Sole s = new Sole("bianco");
    System.out.println("Stella: " + s.nome);
    System.out.println("Colore: "
        + s.colore);
}
```

A. Errore a tempo di compilazione

B. Errore a tempo di esecuzione

C. Stella: Sole

Colore: rosa

D. Stella: Sole

Colore: bianco

E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}
```

A. Errore a tempo di compilazione

B. Errore a tempo di esecuzione

C. Cono: 8 - 3 lati per faccia

D. Cono: 3 - 3 lati per faccia

E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
    }
}
```

```

        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
    super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

12. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia(){
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}

```

- A. 0

- B. 1
- C. 2
- D. 4
- E. 5

13. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
    }
}

```

```

    }
    finally {
        throw( new MyExc3() );
    }
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

15. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
            }
        }
    }
}

```

```

        case 3: throw( new MyExc3() );
    }
}
catch( MyExc1 t ) {
    throw( new MyExc3() );
}
catch( MyExc3 d ) {
    System.out.print(5);
    throw( new MyExc2() );
}
catch( Exception h ) {
    System.out.print(6);
    throw( new MyExc2() );
}
}
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 14666666... (ciclo infinito)
- E. Nessuna delle precedenti

16. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo **static** non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente **static** e **final**
- D. L'operatore **new** non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 5

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}
```

}

```
public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo

D. Pioppo

1

E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Cono: 8 - 3 lati per faccia
 - D. Cono: 3 - 3 lati per faccia
 - E. Nessuna delle precedenti
-

5. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
}
```

```
    }
}
static void q() {
    try {
        System.out.print(3);
        throw( new Exception() );
    }
    finally {
        throw( new MyExc3() );
    }
}
}
```

- A. 31
 - B. Errore a tempo di compilazione
 - C. 312Exception in thread main MyExc2
 - D. 312
 - E. Nessuna delle precedenti
-

6. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza

E. Nessuna delle precedenti

7. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
    }
    int num=6;
    return num;
}
public static void main(String args[]) {
    Pioppo p= new Pioppo();
    System.out.println(p.f());
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

8. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

9. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
}
```

```
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

10. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia(){
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}
```

- A. 0
- B. 1
- C. 2

D. 4

E. 5

11. Date le dichiarazioni:

```
Object [] [] m;  
Error [] q;  
Object [] w;  
q = new Error [9];  
m = new Object [5] [2];  
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

12. Qual è l'output di questo codice?

```
interface FiguraGeometrica {  
    static int dimensioni = 10;  
    void print ();  
}  
  
class Cono implements FiguraGeometrica {  
  
    static final int numLati = 3;  
  
    public void print () {  
        System.out.println("Cono: "  
            + dimensioni + " - "  
            + numLati + " lati per faccia.");  
    }  
  
    public static void main(String [] argv) {  
        Cono c1 = new Cono();  
        FiguraGeometrica f1;  
        c1= (FiguraGeometrica) f1;  
        c1.print ();  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```
class Stella {  
    String nome="Stella";  
    final String colore="rosa";  
  
    public Stella(String nome) {  
        this.nome = nome;  
    }  
}  
  
public class Sole extends Stella {  
  
    public Sole(String c) {  
        super("Sole");  
        this.colore = c;  
    }  
  
    public static void main(String [] argv) {  
        Sole s = new Sole("bianco");  
        System.out.println("Stella: " + s.nome);  
        System.out.println("Colore: "  
            + s.colore);  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

14. Quale output si ottiene invocando il metodo m?

```
class F {  
    private String [] a1 = new String [7];  
    private String [] a4 = new String [2];  
    private Boolean b3 = new Boolean(true);  
    void m() {  
        q(a1, new String [2], b3, b3);  
    }  
    void q(Object a2, String [] a3,  
        Boolean b1, Object b2) {  
        if(a2 == a1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(a3 == a4) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(b2 == b1) {  
            System.out.print(1);  
        } else {
```

```

        System.out.print(0);
    }
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

15. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
    }
}

```

```

    }
}
catch( MyExc1 t ) {
    throw( new MyExc3() );
}
catch( MyExc3 d ) {
    System.out.print(5);
    throw( new MyExc2() );
}
catch( Exception h ) {
    System.out.print(6);
    throw( new MyExc2() );
}
}
}
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

16. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 6

1. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
```

```
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```



```
}  
}
```

- A. Errore a tempo di compilazione
 - B. 1111111... (ciclo infinito)
 - C. Exception in thread main MyExc2
 - D. 1111
 - E. Nessuna delle precedenti
-

4. Qual è l'output di questo codice?

```
abstract class Programmatore {  
    String nome;  
    String cognome;  
    int anniEsperienza;  
  
    public Programmatore(String n, String c,  
                          int a){  
        nome=n;  
        cognome=c;  
        anniEsperienza=a;  
    }  
    abstract void programma();  
}  
public class ProgrammatoreJava  
    extends Programmatore {  
    void stampa(){  
  
        System.out.print("Programmatore Java con ");  
        System.out.print (anniEsperienza);  
        System.out.println(" anni di esperienza");  
    }  
    void programma(){}  
    ProgrammatoreJava(String n, String c, int a){  
        super(n,c,a);  
    }  
    public static void main(String [] argv) {  
        ProgrammatoreJava pj=new  
            ProgrammatoreJava("Franco", "Bianchi",5);  
        pj.stampa();  
    }  
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Programmatore Java con 5 anni di esperienza
 - D. Programmatore Java con 0 anni di esperienza
 - E. Nessuna delle precedenti
-

5. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends Exception { }  
public class C1 {
```

```
public static void main(String [] argv)  
    throws Exception {  
    try {  
        System.out.print(1);  
        n();  
    }  
    catch( MyExc3 s ) {  
        System.out.print(2);  
        throw( new Exception() );  
    }  
    finally {  
        System.out.print(3);  
        throw( new MyExc3() );  
    }  
}  
static void n() throws Exception {  
    try {  
        System.out.print(4);  
        switch(1) {  
            case 1: throw( new MyExc2() );  
            case 2: throw( new MyExc1() );  
            case 3: throw( new MyExc3() );  
        }  
    }  
    catch( MyExc1 t ) {  
        throw( new MyExc3() );  
    }  
    catch( MyExc3 d ) {  
        System.out.print(5);  
        throw( new MyExc2() );  
    }  
    catch( Exception h ) {  
        System.out.print(6);  
        throw( new MyExc2() );  
    }  
}
```

- A. Errore a tempo di compilazione
 - B. 1463Exception in thread main MyExc3
 - C. 143Exception in thread main MyExc3
 - D. 146666666... (ciclo infinito)
 - E. Nessuna delle precedenti
-

6. Date le dichiarazioni:

```
String [] c;  
Object [] [] d;  
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

E. Un oggetto può non esistere dopo la sua dichiarazione

7. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n, c, a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

8. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato

9. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[] ) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
    }
}
```

```

        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

11. Quale output si ottiene invocando il metodo m?

```

class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

12. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia () {

```

```

        System.out.println(marca);
    }
}

```

```

Bottiglia(double litri){
    this.numlitri=litri;
    System.out.println(litri);
}

```

```

public static void main(String args[]) {
    Bottiglia spumante=new Bottiglia();
    Bottiglia vinoRosso=new Bottiglia();
    Bottiglia vinoBianco=new Bottiglia(1.0);
    spumante=null;
    vinoBianco=spumante;
    /*QUI*/
    System.out.println("END");
}
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

13. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

14. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

15. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione

- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Stella: Sole
Colore: rosa
 - D. Stella: Sole
Colore: bianco
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 7

1. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
    }
}
```

```
catch( MyExc3 s ) {
    System.out.print(2);
    throw( new Exception() );
}
finally {
    System.out.print(3);
    throw( new MyExc3() );
}
}
static void n() throws Exception {
    try {
        System.out.print(4);
        switch(1) {
            case 1: throw( new MyExc2() );
            case 2: throw( new MyExc1() );
            case 3: throw( new MyExc3() );
        }
    }
    catch( MyExc1 t ) {
        throw( new MyExc3() );
    }
    catch( MyExc3 d ) {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception h ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
}
}
```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

3. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia(){
```

```

    System.out.println(marca);
}

Bottiglia(double litri){
    this.numlitri=litri;
    System.out.println(litri);
}

public static void main(String args[]) {
    Bottiglia spumante=new Bottiglia();
    Bottiglia vinoRosso=new Bottiglia();
    Bottiglia vinoBianco=new Bottiglia(1.0);
    spumante=null;
    vinoBianco=spumante;
    /*QUI*/
    System.out.println("END");
}
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

4. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class Bl {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +

```

```

        numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

7. Quale output si ottiene invocando il metodo m?

```

class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

8. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

9. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

10. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione

- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n, c, a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
```

```
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

13. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

14. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {
    static final int numLati = 3;
    public void print() {
```

```

System.out.println("Cono: "
+ dimensioni + " - " +
numLati + " lati per faccia.");
}

public static void main(String [] argv) {
Cono c1 = new Cono();
FiguraGeometrica f1;
c1= (FiguraGeometrica) f1;
c1.print();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

```

System.out.print (anniEsperienza);
System.out.println(" anni di esperienza");
}
void programma() {}
ProgrammatoreJava (String n, String c, int a){
super (n,c,a);
}
public static void main (String [] argv) {
ProgrammatoreJava pj=new
ProgrammatoreJava ("Franco", "Bianchi",5);
pj.stampa();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

15. Qual è l'output di questo codice?

```

abstract class Programmatore {
String nome;
String cognome;
int anniEsperienza;

public Programmatore (String n, String c,
int a){
nome=n;
cognome=c;
anniEsperienza=a;
}
abstract void programma();
}
public class ProgrammatoreJava
extends Programmatore {
void stampa(){

System.out.print ("Programmatore Java con ");

```

16. Date le dichiarazioni:

```

Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 8

1. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
System.out.println("Pioppo");
int num=6;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
```

```
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

4. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri) {
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

5. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione

- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

7. Quale output si ottiene invocando il metodo m?

```

class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,

```

```

        Boolean b1, Object b2) {
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a3 == a4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

8. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
super(n, c, a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){
```

```
        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
    super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

13. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
 - B. q = (Error []) w;
 - C. m = (Object [] []) w;
 - D. m = (Object [] []) q;
 - E. Nessuno dei precedenti
-

14. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
 - B. d = (Object [] []) e;
 - C. c = (String []) d;
 - D. d = (Object [] []) c;
 - E. d = e;
-

15. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi

- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
 - C. Un attributo non può essere contemporaneamente static e final
 - D. L'operatore new non restituisce un riferimento all'oggetto appena creato
 - E. Un oggetto può non esistere dopo la sua dichiarazione
-

16. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Pioppo
6
 - D. Pioppo
1
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 9

1. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
System.out.println("Pioppo");
int num=n+1;
return num;
    }
    public static void main(String args[] ) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
}
```

```
}
static void n() throws Exception {
    try {
        System.out.print(4);
        switch(1) {
            case 1: throw( new MyExc2() );
            case 2: throw( new MyExc1() );
            case 3: throw( new MyExc3() );
        }
    }
    catch( MyExc1 t ) {
        throw( new MyExc3() );
    }
    catch( MyExc3 d ) {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception h ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
}
}
```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }
}
```

```

}

public static void main(String [] argv) {
    Sole s = new Sole("bianco");
    System.out.println("Stella: " + s.nome);
    System.out.println("Colore: "
        + s.colore);
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
}

ProgrammatoreJava(String n, String c, int a){
    super(n,c,a);
}

public static void main(String [] argv) {
    ProgrammatoreJava pj=new
        ProgrammatoreJava("Franco", "Bianchi",5);
    pj.stampa();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione

C. Pioppo
6

D. Pioppo
1

E. Nessuna delle precedenti

7. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri) {
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}
```

- A. 0
 - B. 1
 - C. 2
 - D. 4
 - E. 5
-

8. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }
}
```

```
}

public static void main(String [] argv) {
    Cono c1 = new Cono();
    FiguraGeometrica f1;
    c1= (FiguraGeometrica) f1;
    c1.print();
}
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Cono: 8 - 3 lati per faccia
 - D. Cono: 3 - 3 lati per faccia
 - E. Nessuna delle precedenti
-

9. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
 - B. q = (Error []) w;
 - C. m = (Object [] []) w;
 - D. m = (Object [] []) q;
 - E. Nessuno dei precedenti
-

10. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
 - B. Un attributo **static** non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
 - C. Un attributo non può essere contemporaneamente **static** e **final**
 - D. L'operatore **new** non restituisce un riferimento all'oggetto appena creato
 - E. Un oggetto può non esistere dopo la sua dichiarazione
-

11. Quale output si ottiene invocando il metodo m?

```

class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {

```

```

            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

13. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

14. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione

- B. Errore a tempo di esecuzione
 - C. Stella: Sole
Colore: rosa
 - D. Stella: Sole
Colore: bianco
 - E. Nessuna delle precedenti
-

15. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {
    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Cono: 8 - 3 lati per faccia
 - D. Cono: 3 - 3 lati per faccia
 - E. Nessuna delle precedenti
-

16. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Programmatore Java con 5 anni di esperienza
 - D. Programmatore Java con 0 anni di esperienza
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 10

1. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

2. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}
```

```
public class Sole extends Stella {
```

```
    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }
```

```
    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1

E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. Errore a tempo di compilazione
 - B. 111111... (ciclo infinito)
 - C. Exception in thread main MyExc2
 - D. 1111
 - E. Nessuna delle precedenti
-

5. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
```

```
void stampa(){

    System.out.print("Programmatore Java con ");
    System.out.print(anniEsperienza);
    System.out.println(" anni di esperienza");
}
void programma(){}
ProgrammatoreJava(String n, String c, int a){
super(n,c,a);
}
public static void main(String [] argv) {
    ProgrammatoreJava pj=new
        ProgrammatoreJava("Franco", "Bianchi",5);
    pj.stampa();
}
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Programmatore Java con 5 anni di esperienza
 - D. Programmatore Java con 0 anni di esperienza
 - E. Nessuna delle precedenti
-

6. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
 - B. q = (Error []) w;
 - C. m = (Object [] []) w;
 - D. m = (Object [] []) q;
 - E. Nessuno dei precedenti
-

7. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }
}
```

```

public static void main(String [] argv) {
    Cono c1 = new Cono();
    FiguraGeometrica f1 =c1;
    c1.print();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
System.out.println("Pioppo");
int num=6;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {

```

```

        throw( new MyExc3() );
    }
    finally {
        System.out.print(2);
        throw( new MyExc2() );
    }
}
static void q() {
    try {
        System.out.print(3);
        throw( new Exception() );
    }
    finally {
        throw( new MyExc3() );
    }
}
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

10. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );

```

```

    }
}
static void n() throws Exception {
    try {
        System.out.print(4);
        switch(1) {
            case 1: throw( new MyExc2() );
            case 2: throw( new MyExc1() );
            case 3: throw( new MyExc3() );
        }
    }
    catch( MyExc1 t ) {
        throw( new MyExc3() );
    }
    catch( MyExc3 d ) {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception h ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print();
    }
}

```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Cono: 8 - 3 lati per faccia
 - D. Cono: 3 - 3 lati per faccia
 - E. Nessuna delle precedenti
-
13. Qual è l'output di questo codice?
- ```

abstract class Programmatore {
 String nome;
 String cognome;
 int anniEsperienza;

 public Programmatore(String n, String c,
 int a){
 nome=n;
 cognome=c;
 anniEsperienza=a;
 }
 abstract void programma();
}

public class ProgrammatoreJava
 extends Programmatore {

 void stampa(){

 System.out.print("Programmatore Java con ");
 System.out.print(anniEsperienza);
 System.out.println(" anni di esperienza");
 }

 ProgrammatoreJava(String n, String c, int a){
 super(n,c,a);
 }
 public static void main(String [] argv) {
 ProgrammatoreJava pj=new
 ProgrammatoreJava("Franco", "Bianchi",5);
 pj.stampa();
 }
}

```
- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. Programmatore Java con 5 anni di esperienza
  - D. Programmatore Java con 0 anni di esperienza
  - E. Nessuna delle precedenti
- 
14. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?
- ```

public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia(){
        System.out.println(marca);
    }
}

```

```

}

Bottiglia(double litri){
    this.numlitri=litri;
    System.out.println(litri);
}

public static void main(String args[]) {
    Bottiglia spumante=new Bottiglia();
    Bottiglia vinoRosso=new Bottiglia();
    Bottiglia vinoBianco=new Bottiglia(1.0);
    spumante=null;
    vinoBianco=spumante;
    /*QUI*/
    System.out.println("END");
}
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

15. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo `static` non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente `static` e `final`
- D. L'operatore `new` non restituisce un riferimento all'oggetto appena creato

E. Un oggetto può non esistere dopo la sua dichiarazione

16. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Stella: Sole
Colore: rosa
 - D. Stella: Sole
Colore: bianco
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 11

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)

E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {
    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

3. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

4. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma() {}
    ProgrammatoreJava(String n, String c, int a){
    super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

5. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

7. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[] ) {
```

```

        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}

```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Pioppo
6
 - D. Pioppo
1
 - E. Nessuna delle precedenti
-

8. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Pioppo
6
 - D. Pioppo
1
 - E. Nessuna delle precedenti
-

9. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava

```

```

    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Programmatore Java con 5 anni di esperienza
 - D. Programmatore Java con 0 anni di esperienza
 - E. Nessuna delle precedenti
-

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```


- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

12. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri) {
        this.numlitri=litri;
        System.out.println(litri);
    }
}
```

```
public static void main(String args[]) {
    Bottiglia spumante=new Bottiglia();
    Bottiglia vinoRosso=new Bottiglia();
    Bottiglia vinoBianco=new Bottiglia(1.0);
    spumante=null;
    vinoBianco=spumante;
    /*QUI*/
    System.out.println("END");
}
}
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

13. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

15. Quale output si ottiene invocando il metodo m?

```

class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,

```

```

        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

16. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
 - B. d = (Object [] []) e;
 - C. c = (String []) d;
 - D. d = (Object [] []) c;
 - E. d = e;
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 12

1. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

D. 4

E. 5

2. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri) {
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}
```

- A. 0
- B. 1
- C. 2

3. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

4. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

5. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
    }
}
```

```
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

7. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

8. Date le dichiarazioni:

```
String [] c;  
Object [] [] d;  
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `d = c;`
 - B. `d = (Object [] []) e;`
 - C. `c = (String []) d;`
 - D. `d = (Object [] []) c;`
 - E. `d = e;`
-

9. Qual è l'output di questo codice?

```
abstract class Programmatore {  
    String nome;  
    String cognome;  
    int anniEsperienza;  
  
    public Programmatore(String n, String c,  
                          int a){  
  
        nome=n;  
        cognome=c;  
        anniEsperienza=a;  
    }  
    abstract void programma();  
}  
public class ProgrammatoreJava  
    extends Programmatore {  
  
    void stampa(){  
  
        System.out.print("Programmatore Java con ");  
        System.out.print (anniEsperienza);  
        System.out.println(" anni di esperienza");  
    }  
  
    ProgrammatoreJava(String n, String c, int a){  
super(n, c, a);  
    }  
    public static void main(String [] argv) {  
        ProgrammatoreJava pj=new  
        ProgrammatoreJava("Franco", "Bianchi",5);  
        pj.stampa();  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza

10. Qual è l'output di questo codice?

```
interface FiguraGeometrica {  
    static int dimensioni = 10;  
    void print();  
}  
  
class Cono implements FiguraGeometrica {  
  
    static final int numLati = 3;  
  
    public void print() {  
        System.out.println("Cono: "  
        + dimensioni + " - " +  
        numLati + " lati per faccia.");  
    }  
  
    public static void main(String [] argv) {  
        Cono cl = new Cono();  
        FiguraGeometrica fl =cl;  
        cl.print();  
    }  
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Cono: 8 - 3 lati per faccia
 - D. Cono: 3 - 3 lati per faccia
 - E. Nessuna delle precedenti
-

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class Bl {  
    public static void main(String [] argv) {  
        try {  
            q();  
        }  
        catch( MyExc3 i ) {  
            System.out.print(1);  
        }  
        catch( MyExc2 w ) {  
        }  
        catch( MyExc1 j ) {  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(2);  
            throw( new MyExc2() );  
        }  
    }  
}
```

```

static void q() {
    try {
        System.out.print(3);
        throw( new Exception() );
    }
    finally {
        throw( new MyExc3() );
    }
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
        }
    }
}

```

```

        throw( new MyExc2() );
    }
    catch( Exception h ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

15. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
    super(n,c,a);
    }
}

```

```

public static void main(String [] argv) {
    ProgrammatoreJava pj=new
        ProgrammatoreJava("Franco", "Bianchi",5);
    pj.stampa();
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 13

1. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

2. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
```

```
private String [] a4 = new String [2];
private Boolean b3 = new Boolean(true);
void m() {
    q(a1, new String [2], b3, b3);
}
void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a3 == a4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

3. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia(){
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }
}
```

```

public static void main(String args[]) {
    Bottiglia spumante=new Bottiglia();
    Bottiglia vinoRosso=new Bottiglia();
    Bottiglia vinoBianco=new Bottiglia(1.0);
    spumante=null;
    vinoBianco=spumante;
    /*QUI*/
    System.out.println("END");
}
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

4. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312

E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

6. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

8. Date le dichiarazioni:

```

Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

9. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print ();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print () {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print ();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {

```

```

    catch( MyExc2 y ) {
        System.out.print(1);
        throw( new MyExc3() );
    }
    catch( MyExc3 g ) {
        System.out.print(2);
    }
    finally {
        throw( new MyExc3() );
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

12. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

13. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
System.out.println("Pioppo");
int num=6;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

15. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
System.out.println("Pioppo");
int num=n+1;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione

- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 14

1. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma() {}
    ProgrammatoreJava(String n, String c, int a){
    super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
        ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

2. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
```

```
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

3. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}
public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

4. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

5. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

6. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";
```

```
public Stella(String nome) {
    this.nome = nome;
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

7. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
System.out.println("Pioppo");
int num=6;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){ }
super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
        ProgrammatoreJava ("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

11. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

12. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

13. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print ();
}
```

```
class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class Bl {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. 31
 - B. Errore a tempo di compilazione
 - C. 312Exception in thread main MyExc2
 - D. 312
 - E. Nessuna delle precedenti
-

15. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. Errore a tempo di compilazione
 - B. 1111111... (ciclo infinito)
 - C. Exception in thread main MyExc2
 - D. 1111
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 15

1. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
    super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
```

```
public static void main(String [] argv)
    throws Exception {
    try {
        System.out.print(1);
        n();
    }
    catch( MyExc3 s ) {
        System.out.print(2);
        throw( new Exception() );
    }
    finally {
        System.out.print(3);
        throw( new MyExc3() );
    }
}
static void n() throws Exception {
    try {
        System.out.print(4);
        switch(1) {
            case 1: throw( new MyExc2() );
            case 2: throw( new MyExc1() );
            case 3: throw( new MyExc3() );
        }
    }
    catch( MyExc1 t ) {
        throw( new MyExc3() );
    }
    catch( MyExc3 d ) {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception h ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
}
}
```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

5. Date le dichiarazioni:

```

Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

6. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

7. Quale output si ottiene invocando il metodo m?

```

class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

8. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print ();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print () {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print ();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

9. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

10. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
    }
}
```

```
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```
interface albero { int f(int i); }
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
    }
}
```

```

    finally {
        System.out.print(2);
        throw( new MyExc2() );
    }
}
static void q() {
    try {
        System.out.print(3);
        throw( new Exception() );
    }
    finally {
        throw( new MyExc3() );
    }
}
}
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

15. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

16. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n, c, a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Programmatore Java con 5 anni di esperienza
 - D. Programmatore Java con 0 anni di esperienza
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 16

1. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

2. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);

```

```
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

3. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
}
```

```

ProgrammatoreJava(String n, String c, int a){
super(n,c,a);
}
public static void main(String [] argv) {
ProgrammatoreJava pj=new
ProgrammatoreJava ("Franco", "Bianchi",5);
pj.stampa();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```

class Stella {
String nome="Stella";
final String colore="rosa";

public Stella(String nome) {
this.nome = nome;
}
}

public class Sole extends Stella {

public Sole(String c) {
super("Sole");
this.colore = c;
}

public static void main(String [] argv) {
Sole s = new Sole("bianco");
System.out.println("Stella: " + s.nome);
System.out.println("Colore: "
+ s.colore);
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
String tipo="Sempre Verde";
public int f(int n)
{
System.out.println("Pioppo");
int num=n+1;
return num;
}
public static void main(String args[]) {
Pioppo p= new Pioppo();
System.out.println(p.f(5));
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
public static void main(String [] argv) {
try {
q();
}
catch( MyExc3 i ) {
System.out.print(1);
}
catch( MyExc2 w ) {
}
catch( MyExc1 j ) {
throw( new MyExc3() );
}
finally {
System.out.print(2);
throw( new MyExc2() );
}
}
static void q() {
try {
System.out.print(3);
throw( new Exception() );
}
finally {
throw( new MyExc3() );
}
}
}
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 14666666... (ciclo infinito)
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione

- C. Pioppo
6
 - D. Pioppo
1
 - E. Nessuna delle precedenti
-

10. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Stella: Sole
Colore: rosa
 - D. Stella: Sole
Colore: bianco
 - E. Nessuna delle precedenti
-

11. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri) {
        this.numlitri=litri;
        System.out.println(litri);
    }
}
```

```
public static void main(String args[]) {
    Bottiglia spumante=new Bottiglia();
    Bottiglia vinoRosso=new Bottiglia();
    Bottiglia vinoBianco=new Bottiglia(1.0);
    spumante=null;
    vinoBianco=spumante;
    /*QUI*/
    System.out.println("END");
}
}
```

- A. 0
 - B. 1
 - C. 2
 - D. 4
 - E. 5
-

12. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
 - B. q = (Error []) w;
 - C. m = (Object [] []) w;
 - D. m = (Object [] []) q;
 - E. Nessuno dei precedenti
-

13. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
 - B. d = (Object [] []) e;
 - C. c = (String []) d;
 - D. d = (Object [] []) c;
 - E. d = e;
-

14. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi

- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

15. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia

E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. Errore a tempo di compilazione
 - B. 111111... (ciclo infinito)
 - C. Exception in thread main MyExc2
 - D. 1111
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 17

1. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
}
```

```
    }
    abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

3. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

4. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
```

```
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

7. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Pioppo
6
 - D. Pioppo
1
 - E. Nessuna delle precedenti
-

8. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
 - B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
 - C. Un attributo non può essere contemporaneamente static e final
 - D. L'operatore new non restituisce un riferimento all'oggetto appena creato
 - E. Un oggetto può non esistere dopo la sua dichiarazione
-

9. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}
```

- A. 0
- B. 1
- C. 2

- D. 4
 - E. 5
-

10. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
 - B. 100
 - C. 011
 - D. 101
 - E. 010
-

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
            }
        }
    }
}
```

```

    catch( MyExc2 y ) {
        System.out.print(1);
        throw( new MyExc3() );
    }
    catch( MyExc3 g ) {
        System.out.print(2);
    }
    finally {
        throw( new MyExc3() );
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

12. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

13. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print ();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print () {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print ();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

15. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {

```

```

            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 18

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

2. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza

- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

3. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

4. Qual è l'output di questo codice?


```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ")
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n, c, a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
        ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }
}

```

```

public static void main(String [] argv) {
    Sole s = new Sole("bianco");
    System.out.println("Stella: " + s.nome);
    System.out.println("Colore: "
        + s.colore);
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

7. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
    }
}

```

```
int num=n+1;
return num;
}
public static void main(String args[]) {
    Pioppo p= new Pioppo();
    System.out.println(p.f(5));
}
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

8. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

9. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

10. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

13. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia(){
        System.out.println(marca);
    }
}

```

```

Bottiglia(double litri){
    this.numlitri=litri;
    System.out.println(litri);
}

public static void main(String args[]) {
    Bottiglia spumante=new Bottiglia();
    Bottiglia vinoRosso=new Bottiglia();
    Bottiglia vinoBianco=new Bottiglia(1.0);
    spumante=null;
    vinoBianco=spumante;
    /*QUI*/
    System.out.println("END");
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

14. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione

- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

A. 31

- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Cono: 8 - 3 lati per faccia
 - D. Cono: 3 - 3 lati per faccia
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 19

1. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

2. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print ();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;
```

```
public void print () {
    System.out.println("Cono: "
        + dimensioni + " - " +
        numLati + " lati per faccia.");
}

public static void main(String [] argv) {
    Cono c1 = new Cono();
    FiguraGeometrica f1 =c1;
    c1.print ();
}
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione

- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

5. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

6. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

7. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print ();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print () {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
```

```

    cl= (FiguraGeometrica) f1;
    cl.print();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

8. Date le dichiarazioni:

```

Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

9. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
}

```

```

ProgrammatoreJava(String n, String c, int a){
super(n,c,a);
}
public static void main(String [] argv) {
    ProgrammatoreJava pj=new
        ProgrammatoreJava("Franco", "Bianchi",5);
    pj.stampa();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
super(n,c,a);
}
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?


```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

12. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final

- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

13. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {

```

```

        throw( new MyExc3() );
    }
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

15. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
System.out.println("Pioppo");
int num=6;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 20

1. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
```

```
Pioppo p= new Pioppo();
System.out.println(p.f());
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1

E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

7. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
```

```

    System.out.println("Cono: "
        + dimensioni + " - " +
        numLati + " lati per faccia.");
}

public static void main(String [] argv) {
    Cono c1 = new Cono();
    FiguraGeometrica f1 =c1;
    c1.print();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

```

        System.out.print(1);
        throw( new MyExc3() );
    }
    catch( MyExc3 g ) {
        System.out.print(2);
    }
    finally {
        throw( new MyExc3() );
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

8. Date le dichiarazioni:

```

Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {

```

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );

```

```

    }
}
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

12. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri) {
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

13. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

14. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;

- B. d = (Object [] []) e;
 - C. c = (String []) d;
 - D. d = (Object [] []) c;
 - E. d = e;
-

15. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
 - B. 100
 - C. 011
 - D. 101
 - E. 010
-

16. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Programmatore Java con 5 anni di esperienza
 - D. Programmatore Java con 0 anni di esperienza
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 21

1. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}
```

}

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
        }
    }
}
```

```

        throw( new MyExc2() );
    }
    catch( Exception h ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {

```

```

        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

6. Date le dichiarazioni:

```

Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

7. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {

```

```

        throw( new MyExc3() );
    }
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

9. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

10. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa

D. Stella: Sole
Colore: bianco

E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

12. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
```

```
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

13. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Programmatore Java con 5 anni di esperienza
 - D. Programmatore Java con 0 anni di esperienza
 - E. Nessuna delle precedenti
-

14. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri) {
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}
```

- A. 0
 - B. 1
 - C. 2
 - D. 4
 - E. 5
-

15. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
 - B. Un attributo `static` non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
 - C. Un attributo non può essere contemporaneamente `static` e `final`
 - D. L'operatore `new` non restituisce un riferimento all'oggetto appena creato
 - E. Un oggetto può non esistere dopo la sua dichiarazione
-

16. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Pioppo
6
 - D. Pioppo
1
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 22

1. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

2. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final

D. L'operatore new non restituisce un riferimento all'oggetto appena creato

E. Un oggetto può non esistere dopo la sua dichiarazione

3. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri) {
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

4. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
```



```

    public int f()
    {
System.out.println("Pioppo");
int num=6;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

5. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
}

```

```

}
static void n() throws Exception {
    try {
        System.out.print(4);
        switch(1) {
            case 1: throw( new MyExc2() );
            case 2: throw( new MyExc1() );
            case 3: throw( new MyExc3() );
        }
    }
    catch( MyExc1 t ) {
        throw( new MyExc3() );
    }
    catch( MyExc3 d ) {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception h ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

7. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
}

```

```
public static void main(String [] argv) {
    ProgrammatoreJava pj=new
        ProgrammatoreJava("Franco", "Bianchi",5);
    pj.stampa();
}
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica fl =c1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
    }
}
```

```
    }
    finally {
        throw( new MyExc2() );
    }
}
static void q() throws Exception {
    try {
    }
    catch( MyExc2 y ) {
        System.out.print(1);
        throw( new MyExc3() );
    }
    catch( MyExc3 g ) {
        System.out.print(2);
    }
    finally {
        throw( new MyExc3() );
    }
}
}
```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

```
}  
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Programmatore Java con 5 anni di esperienza
 - D. Programmatore Java con 0 anni di esperienza
 - E. Nessuna delle precedenti
-

11. Date le dichiarazioni:

```
Object [] [] m;  
Error [] q;  
Object [] w;  
q = new Error [9];  
m = new Object [5] [2];  
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `q = (Error []) m;`
 - B. `q = (Error []) w;`
 - C. `m = (Object [] []) w;`
 - D. `m = (Object [] []) q;`
 - E. Nessuno dei precedenti
-

12. Qual è l'output di questo codice?

```
interface FiguraGeometrica {  
    static int dimensioni = 10;  
    void print ();  
}  
  
class Cono implements FiguraGeometrica {  
  
    static final int numLati = 3;  
  
    public void print () {  
        System.out.println("Cono: "  
            + dimensioni + " - " +  
            numLati + " lati per faccia.");  
    }  
  
    public static void main(String [] argv) {  
        Cono c1 = new Cono();  
        FiguraGeometrica f1;  
        c1= (FiguraGeometrica) f1;  
        c1.print ();  
    }  
}
```

- A. Errore a tempo di compilazione

- B. Errore a tempo di esecuzione
 - C. Cono: 8 - 3 lati per faccia
 - D. Cono: 3 - 3 lati per faccia
 - E. Nessuna delle precedenti
-

13. Qual è l'output di questo codice?

```
class Stella {  
    String nome="Stella";  
    String colore="rosa";  
  
    public Stella(String nome) {  
        this.nome = nome;  
    }  
}  
  
public class Sole extends Stella {  
  
    public Sole(String c) {  
        super("Sole");  
        this.colore = c;  
    }  
  
    public static void main(String [] argv) {  
        Sole s = new Sole("bianco");  
        System.out.println("Stella: " + s.nome);  
        System.out.println("Colore: "  
            + s.colore);  
    }  
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Stella: Sole
Colore: rosa
 - D. Stella: Sole
Colore: bianco
 - E. Nessuna delle precedenti
-

14. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class B1 {  
    public static void main(String [] argv) {  
        try {  
            q();  
        }  
        catch( MyExc3 i ) {  
            System.out.print(1);  
        }  
        catch( MyExc2 w ) {  
        }  
        catch( MyExc1 j ) {  
            throw( new MyExc3() );  
        }  
        finally {
```

```

        System.out.print(2);
        throw( new MyExc2() );
    }
}
static void q() {
    try {
        System.out.print(3);
        throw( new Exception() );
    }
    finally {
        throw( new MyExc3() );
    }
}
}

```

- A. 31
 - B. Errore a tempo di compilazione
 - C. 312Exception in thread main MyExc2
 - D. 312
 - E. Nessuna delle precedenti
-

15. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
    }
}

```

```

        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Stella: Sole
Colore: rosa
 - D. Stella: Sole
Colore: bianco
 - E. Nessuna delle precedenti
-

16. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Pioppo
6
 - D. Pioppo
1
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 23

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;
```

```
public Programmatore(String n, String c,
                    int a){
    nome=n;
    cognome=c;
    anniEsperienza=a;
}
abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

3. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;

- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

4. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
    }
}
```

```
catch( MyExc2 y ) {
    System.out.print(1);
    throw( new MyExc3() );
}
catch( MyExc3 g ) {
    System.out.print(2);
}
finally {
    throw( new MyExc3() );
}
}
```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

6. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

7. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
System.out.println("Pioppo");
int num=6;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono cl = new Cono();
        FiguraGeometrica fl;
        cl= (FiguraGeometrica) fl;
        cl.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
System.out.println("Pioppo");
int num=n+1;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1

11. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia(){
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
    }
}
```



```

vinoBianco=spumante;
/*QUI*/
System.out.println("END");
}
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

12. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

13. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo **static** non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente **static** e **final**
- D. L'operatore **new** non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

14. Quale output si ottiene invocando il metodo m?

```

class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
          Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

15. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}

```

A. Errore a tempo di compilazione

- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 24

1. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
System.out.println("Pioppo");
int num=6;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

void stampa(){
```

System.out.print("Programmatore Java con ");

```
System.out.print (anniEsperienza);
System.out.println(" anni di esperienza");
}

ProgrammatoreJava(String n, String c, int a){
super(n,c,a);
}
public static void main(String [] argv) {
    ProgrammatoreJava pj=new
        ProgrammatoreJava("Franco", "Bianchi",5);
    pj.stampa();
}
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione

- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}
```

```
class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

7. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono cl = new Cono();
        FiguraGeometrica fl;
        cl = (FiguraGeometrica) fl;
        cl.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

8. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
```

```
public static void main(String [] argv) {
    try {
        q();
    }
    catch( MyExc3 i ) {
        System.out.print(1);
    }
    catch( MyExc2 w ) {
    }
    catch( MyExc1 j ) {
        throw( new MyExc3() );
    }
    finally {
        System.out.print(2);
        throw( new MyExc2() );
    }
}
static void q() {
    try {
        System.out.print(3);
        throw( new Exception() );
    }
    finally {
        throw( new MyExc3() );
    }
}
}
```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

10. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

```

```

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

13. Quale output si ottiene invocando il metodo m?

```

class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

14. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

15. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;

- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

16. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia(){
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}
```

- A. 0
 - B. 1
 - C. 2
 - D. 4
 - E. 5
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 25

1. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
System.out.println("Pioppo");
int num=6;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
    }
}
```

```
System.out.println("Colore: "
    + s.colore);
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

3. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

```

static void q() throws Exception {
    try {
    }
    catch( MyExc2 y ) {
        System.out.print(1);
        throw( new MyExc3() );
    }
    catch( MyExc3 g ) {
        System.out.print(2);
    }
    finally {
        throw( new MyExc3() );
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

5. Quale output si ottiene invocando il metodo m?

```

class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

6. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma() {}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

7. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia(){
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
    }
}

```

```

Bottiglia vinoRosso=new Bottiglia();
Bottiglia vinoBianco=new Bottiglia(1.0);
spumante=null;
vinoBianco=spumante;
/*QUI*/
System.out.println("END");
}
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

8. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
System.out.println("Pioppo");
int num=n+1;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {

```

```

        System.out.print(2);
        throw( new Exception() );
    }
    finally {
        System.out.print(3);
        throw( new MyExc3() );
    }
}
static void n() throws Exception {
    try {
        System.out.print(4);
        switch(1) {
            case 1: throw( new MyExc2() );
            case 2: throw( new MyExc1() );
            case 3: throw( new MyExc3() );
        }
    }
    catch( MyExc1 t ) {
        throw( new MyExc3() );
    }
    catch( MyExc3 d ) {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception h ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){
        System.out.print("Programmatore Java con ");

```

```

    System.out.print (anniEsperienza);
    System.out.println(" anni di esperienza");
}

ProgrammatoreJava(String n, String c, int a){
super(n,c,a);
}
public static void main(String [] argv) {
    ProgrammatoreJava pj=new
    ProgrammatoreJava("Franco", "Bianchi",5);
    pj.stampa();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class Bl {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
    }
}

```

```

    }
    finally {
        throw( new MyExc3() );
    }
}
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

```

```

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica fl =c1;
        c1.print();
    }
}

```

- A. Errore a tempo di compilazione

- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

15. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

16. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 26

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)

E. Nessuna delle precedenti

2. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

3. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```


indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

4. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
System.out.println("Pioppo");
int num=6;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

5. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

6. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri) {
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

7. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma (){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
```

```
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

11. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

12. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print ();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print () {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print ();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
```

```
            System.out.print (1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3 () );
        }
        finally {
            System.out.print (2);
            throw( new MyExc2 () );
        }
    }
    static void q() {
        try {
            System.out.print (3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3 () );
        }
    }
}
```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa

- D. Stella: Sole
Colore: bianco
 - E. Nessuna delle precedenti
-

15. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Stella: Sole
Colore: rosa
 - D. Stella: Sole
Colore: bianco
 - E. Nessuna delle precedenti
-

16. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava ("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Programmatore Java con 5 anni di esperienza
 - D. Programmatore Java con 0 anni di esperienza
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 27

1. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
    }
}
```

```
}
finally {
    System.out.print(3);
    throw( new MyExc3() );
}
}
static void n() throws Exception {
    try {
        System.out.print(4);
        switch(1) {
            case 1: throw( new MyExc2() );
            case 2: throw( new MyExc1() );
            case 3: throw( new MyExc3() );
        }
    }
    catch( MyExc1 t ) {
        throw( new MyExc3() );
    }
    catch( MyExc3 d ) {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception h ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
}
}
```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;
```

```

public void print() {
    System.out.println("Cono: "
        + dimensioni + " - " +
        numLati + " lati per faccia.");
}

public static void main(String [] argv) {
    Cono c1 = new Cono();
    FiguraGeometrica f1;
    c1= (FiguraGeometrica) f1;
    c1.print();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

6. Date le dichiarazioni:

```

Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

7. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

8. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
          Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```
if(a3 == a4) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(b2 == b1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

9. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?


```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
System.out.println("Pioppo");
int num=n+1;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
System.out.println("Pioppo");
int num=6;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {

```

```

public static void main(String [] argv)
    throws Exception {
    try {
        q();
    }
    finally {
        throw( new MyExc2() );
    }
}
static void q() throws Exception {
    try {
    }
    catch( MyExc2 y ) {
        System.out.print(1);
        throw( new MyExc3() );
    }
    catch( MyExc3 g ) {
        System.out.print(2);
    }
    finally {
        throw( new MyExc3() );
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

13. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

14. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri) {

```

```

    this.numlitri=litri;
    System.out.println(litri);
}

public static void main(String args[]) {
    Bottiglia spumante=new Bottiglia();
    Bottiglia vinoRosso=new Bottiglia();
    Bottiglia vinoBianco=new Bottiglia(1.0);
    spumante=null;
    vinoBianco=spumante;
    /*QUI*/
    System.out.println("END");
}
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

15. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

```

```

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

16. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
 - B. Un attributo `static` non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
 - C. Un attributo non può essere contemporaneamente `static` e `final`
 - D. L'operatore `new` non restituisce un riferimento all'oggetto appena creato
 - E. Un oggetto può non esistere dopo la sua dichiarazione
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 28

1. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

2. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
}
```

```
public static void main(String [] argv) {
    ProgrammatoreJava pj=new
        ProgrammatoreJava("Franco", "Bianchi",5);
    pj.stampa();
}
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

4. Date le dichiarazioni:

```
String [] c;  
Object [] [] d;  
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

5. Qual è l'output di questo codice?

```
interface FiguraGeometrica {  
    static int dimensioni = 10;  
    void print();  
}  
  
class Cono implements FiguraGeometrica {  
  
    static final int numLati = 3;  
  
    public void print() {  
        System.out.println("Cono: "  
            + dimensioni + " - " +  
            numLati + " lati per faccia.");  
    }  
  
    public static void main(String [] argv) {  
        Cono c1 = new Cono();  
        FiguraGeometrica f1;  
        c1 = (FiguraGeometrica) f1;  
        c1.print();  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends MyExc2 { }  
public class D1 {
```

```
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            q();  
        }  
        finally {  
            throw( new MyExc2() );  
        }  
    }  
    static void q() throws Exception {  
        try {  
        }  
        catch( MyExc2 y ) {  
            System.out.print(1);  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 g ) {  
            System.out.print(2);  
        }  
        finally {  
            throw( new MyExc3() );  
        }  
    }  
}
```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

7. Qual è l'output di questo codice?

```
class Stella {  
    String nome="Stella";  
    String colore="rosa";  
  
    public Stella(String nome) {  
        this.nome = nome;  
    }  
}  
  
public class Sole extends Stella {  
  
    public Sole(String c) {  
        super("Sole");  
        this.colore = c;  
    }  
  
    public static void main(String [] argv) {  
        Sole s = new Sole("bianco");  
        System.out.println("Stella: " + s.nome);  
        System.out.println("Colore: "  
            + s.colore);  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione

- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

8. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri) {
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

9. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}
```

10. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }
}
```

```

}
void programma() {}
ProgrammatoreJava(String n, String c, int a) {
super(n, c, a);
}
public static void main(String [] argv) {
ProgrammatoreJava pj=new
ProgrammatoreJava("Franco", "Bianchi",5);
pj.stampa();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class Bl {
public static void main(String [] argv) {
try {
q();
}
catch( MyExc3 i ) {
System.out.print(1);
}
catch( MyExc2 w ) {
}
catch( MyExc1 j ) {
throw( new MyExc3() );
}
finally {
System.out.print(2);
throw( new MyExc2() );
}
}
static void q() {
try {
System.out.print(3);
throw( new Exception() );
}
finally {
throw( new MyExc3() );
}
}
}
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
static int dimensioni = 10;
void print();
}

class Cono implements FiguraGeometrica {

static final int numLati = 3;

public void print() {
System.out.println("Cono: "
+ dimensioni + " - " +
numLati + " lati per faccia.");
}

public static void main(String [] argv) {
Cono cl = new Cono();
FiguraGeometrica fl =cl;
cl.print();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

14. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

15. Quale output si ottiene invocando il metodo m?

```

class F {
private String [] a1 = new String [7];
private String [] a4 = new String [2];
private Boolean b3 = new Boolean(true);
void m() {
q(a1, new String [2], b3, b3);
}
}

```

```

void q(Object a2, String [] a3,
    Boolean b1, Object b2) {
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a3 == a4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
    }
}

```

```

catch( MyExc3 s ) {
    System.out.print(2);
    throw( new Exception() );
}
finally {
    System.out.print(3);
    throw( new MyExc3() );
}
}
static void n() throws Exception {
    try {
        System.out.print(4);
        switch(1) {
            case 1: throw( new MyExc2() );
            case 2: throw( new MyExc1() );
            case 3: throw( new MyExc3() );
        }
    }
    catch( MyExc1 t ) {
        throw( new MyExc3() );
    }
    catch( MyExc3 d ) {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception h ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
}
}
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 29

1. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

2. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
}
```

```
public static void main(String args[]) {
    Pioppo p= new Pioppo();
    System.out.println(p.f(5));
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
```

```

        throw( new MyExc3() );
    }
    catch( MyExc3 d ) {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception h ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

4. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

5. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

6. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

7. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo `static` non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente `static` e `final`
- D. L'operatore `new` non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

8. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `q = (Error []) m;`
- B. `q = (Error []) w;`
- C. `m = (Object [] []) w;`
- D. `m = (Object [] []) q;`
- E. Nessuno dei precedenti

9. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print ();
}
```

```
class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print () {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print ();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
```

```
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. Errore a tempo di compilazione
 - B. 111111... (ciclo infinito)
 - C. Exception in thread main MyExc2
 - D. 1111
 - E. Nessuna delle precedenti
-

15. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Cono: 8 - 3 lati per faccia
 - D. Cono: 3 - 3 lati per faccia
 - E. Nessuna delle precedenti
-

16. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Programmatore Java con 5 anni di esperienza
 - D. Programmatore Java con 0 anni di esperienza
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 30

1. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `d = c;`
- B. `d = (Object [] []) e;`
- C. `c = (String []) d;`
- D. `d = (Object [] []) c;`
- E. `d = e;`

2. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1 = (FiguraGeometrica) f1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia

- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }
}
```



```

}

public static void main(String [] argv) {
    Sole s = new Sole("bianco");
    System.out.println("Stella: " + s.nome);
    System.out.println("Colore: "
        + s.colore);
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 = c1;
        c1.print();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {

```

```

    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

7. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione

- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

8. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo `static` non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente `static` e `final`
- D. L'operatore `new` non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

9. Quale output si ottiene invocando il metodo `m`?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
           Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

10. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n, c, a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```
class MyExcl extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
    }
}
```

```

    }
    finally {
        System.out.print(3);
        throw( new MyExc3() );
    }
}
static void n() throws Exception {
    try {
        System.out.print(4);
        switch(1) {
            case 1: throw( new MyExc2() );
            case 2: throw( new MyExc1() );
            case 3: throw( new MyExc3() );
        }
    }
    catch( MyExc1 t ) {
        throw( new MyExc3() );
    }
    catch( MyExc3 d ) {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception h ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

12. Date le dichiarazioni:

```

Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

13. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

14. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia(){
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
    }
}

```

```

    spumante=null;
    vinoBianco=spumante;
    /*QUI*/
    System.out.println("END");
}
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

15. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
System.out.println("Pioppo");
int num=n+1;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 31

1. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono cl = new Cono();
        FiguraGeometrica fl =cl;
        cl.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

2. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;

- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

3. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

4. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

5. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri) {
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

6. Qual è l'output di questo codice?

```
interface albero { int f(int i); }
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

7. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
}
```

```

    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

10. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

11. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```


- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
System.out.println("Pioppo");
int num=6;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
```

```
        switch(1) {
            case 1: throw( new MyExc2() );
            case 2: throw( new MyExc1() );
            case 3: throw( new MyExc3() );
        }
    }
    catch( MyExc1 t ) {
        throw( new MyExc3() );
    }
    catch( MyExc3 d ) {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception h ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
}
```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. 31
 - B. Errore a tempo di compilazione
 - C. 312Exception in thread main MyExc2
 - D. 312
 - E. Nessuna delle precedenti
-

15. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco

- E. Nessuna delle precedenti
-

16. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1 = (FiguraGeometrica) f1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Cono: 8 - 3 lati per faccia
 - D. Cono: 3 - 3 lati per faccia
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 32

1. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

3. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
        }
    }
}
```

```

        throw( new MyExc3() );
    }
}
static void n() throws Exception {
    try {
        System.out.print(4);
        switch(1) {
            case 1: throw( new MyExc2() );
            case 2: throw( new MyExc1() );
            case 3: throw( new MyExc3() );
        }
    }
    catch( MyExc1 t ) {
        throw( new MyExc3() );
    }
    catch( MyExc3 d ) {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception h ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

5. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `d = c;`
- B. `d = (Object [] []) e;`
- C. `c = (String []) d;`
- D. `d = (Object [] []) c;`
- E. `d = e;`

6. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

```

```

    }
}
public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

7. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo `static` non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente `static` e `final`
- D. L'operatore `new` non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

8. Quale output si ottiene invocando il metodo `m`?

```

class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {

```

```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class Bl {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }
}

```

```

}

ProgrammatoreJava(String n, String c, int a){
super(n, c, a);
}
public static void main(String [] argv) {
ProgrammatoreJava pj=new
ProgrammatoreJava ("Franco", "Bianchi",5);
pj.stampa ();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

12. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
static String marca="Ferrari";
double numlitri=1.5;

Bottiglia () {
System.out.println(marca);
}

Bottiglia(double litri){
this.numlitri=litri;
System.out.println(litri);
}

public static void main(String args[]) {
Bottiglia spumante=new Bottiglia();
Bottiglia vinoRosso=new Bottiglia();
Bottiglia vinoBianco=new Bottiglia(1.0);
spumante=null;
vinoBianco=spumante;
/*QUI*/
System.out.println("END");
}
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

13. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
String tipo="Sempre Verde";
public int f(int n)
{
System.out.println("Pioppo");
int num=n+1;
return num;
}
public static void main(String args[]) {
Pioppo p= new Pioppo();
System.out.println(p.f(5));
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
public static void main(String [] argv)
throws Exception {
try {
q();
}
finally {
throw( new MyExc2() );
}
}
static void q() throws Exception {
try {
}
catch( MyExc2 y ) {
System.out.print(1);
throw( new MyExc3() );
}
catch( MyExc3 g ) {
System.out.print(2);
}
finally {
throw( new MyExc3() );
}
}
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111

E. Nessuna delle precedenti

15. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono cl = new Cono();
        FiguraGeometrica f1;
        cl = (FiguraGeometrica) f1;
        cl.print();
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Cono: 8 - 3 lati per faccia
 - D. Cono: 3 - 3 lati per faccia
 - E. Nessuna delle precedenti
-

16. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n, c, a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Programmatore Java con 5 anni di esperienza
 - D. Programmatore Java con 0 anni di esperienza
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 33

1. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono cl = new Cono();
        FiguraGeometrica fl = cl;
        cl.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione

- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

4. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

5. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;

6. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

7. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia(){
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
    }
}
```

```

vinoBianco=spumante;
/*QUI*/
System.out.println("END");
}
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

8. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
System.out.println("Pioppo");
int num=n+1;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }
}

```

```

}

public static void main(String [] argv) {
    Sole s = new Sole("bianco");
    System.out.println("Stella: " + s.nome);
    System.out.println("Colore: "
        + s.colore);
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

12. Date le dichiarazioni:

```

Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

14. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza

- C. Un attributo non può essere contemporaneamente `static` e `final`
- D. L'operatore `new` non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

- D. 1111
- E. Nessuna delle precedenti

15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1111111... (ciclo infinito)
- C. Exception in thread main MyExc2

16. Quale output si ottiene invocando il metodo `m`?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
 - B. 100
 - C. 011
 - D. 101
 - E. 010
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 34

1. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

2. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}
```

```
public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
}
```



```

}

ProgrammatoreJava(String n, String c, int a){
super(n,c,a);
}
public static void main(String [] argv) {
ProgrammatoreJava pj=new
ProgrammatoreJava("Franco", "Bianchi",5);
pj.stampa();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

4. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
public static void main(String [] argv)
throws Exception {
try {
q();
}
finally {
throw( new MyExc2() );
}
}
static void q() throws Exception {
try {
}
catch( MyExc2 y ) {
System.out.print(1);
throw( new MyExc3() );
}
}
}

```

```

}
catch( MyExc3 g ) {
System.out.print(2);
}
finally {
throw( new MyExc3() );
}
}
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

6. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

7. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
String tipo="Sempre Verde";
public int f()
{
System.out.println("Pioppo");
int num=6;
return num;
}
public static void main(String args[]) {
Pioppo p= new Pioppo();
System.out.println(p.f());
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
```

```
        case 1: throw( new MyExc2() );
        case 2: throw( new MyExc1() );
        case 3: throw( new MyExc3() );
        }
    }
    catch( MyExc1 t ) {
        throw( new MyExc3() );
    }
    catch( MyExc3 d ) {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception h ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
}
}
```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

```

```

public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n, c, a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");

```

```

        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

14. Date le dichiarazioni:

```

Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;

E. Nessuno dei precedenti

15. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri) {
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

16. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. 31
 - B. Errore a tempo di compilazione
 - C. 312Exception in thread main MyExc2
 - D. 312
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 35

1. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
System.out.println("Pioppo");
int num=6;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
}
```

```
    }
}
static void q() {
    try {
        System.out.print(3);
        throw( new Exception() );
    }
    finally {
        throw( new MyExc3() );
    }
}
```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

3. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

4. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}
```

```

}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)

- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

7. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {
    static final int numLati = 3;
}

```

```

public void print() {
    System.out.println("Cono: "
        + dimensioni + " - " +
        numLati + " lati per faccia.");
}

public static void main(String [] argv) {
    Cono c1 = new Cono();
    FiguraGeometrica f1;
    c1= (FiguraGeometrica) f1;
    c1.print();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

8. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

9. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }
}

```

```

public static void main(String [] argv) {
    Sole s = new Sole("bianco");
    System.out.println("Stella: " + s.nome);
    System.out.println("Colore: "
        + s.colore);
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}
}

```


- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

11. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

12. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia(){
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

13. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

15. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

16. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma() {}
    ProgrammatoreJava(String n, String c, int a){
        super(n, c, a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Programmatore Java con 5 anni di esperienza
 - D. Programmatore Java con 0 anni di esperienza
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 36

1. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono cl = new Cono();
        FiguraGeometrica fl =cl;
        cl.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
```

```
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione

- C. Pioppo
6
 - D. Pioppo
1
 - E. Nessuna delle precedenti
-

4. Date le dichiarazioni:

```
Object [] [] m;  
Error [] q;  
Object [] w;  
q = new Error [9];  
m = new Object [5] [2];  
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
 - B. q = (Error []) w;
 - C. m = (Object [] []) w;
 - D. m = (Object [] []) q;
 - E. Nessuno dei precedenti
-

5. Qual è l'output di questo codice?

```
interface albero { int f(int i);}  
class Pioppo implements albero {  
    String tipo="Sempre Verde";  
    public int f()  
    {  
System.out.println("Pioppo");  
int num=6;  
return num;  
    }  
    public static void main(String args[]) {  
        Pioppo p= new Pioppo();  
        System.out.println(p.f());  
    }  
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Pioppo
6
 - D. Pioppo
1
 - E. Nessuna delle precedenti
-

6. Qual è l'output di questo codice?

```
class Stella {  
    String nome="Stella";  
    final String colore="rosa";  
  
    public Stella(String nome) {  
        this.nome = nome;  
    }  
}  
  
public class Sole extends Stella {  
  
    public Sole(String c) {  
        super("Sole");  
        this.colore = c;  
    }  
  
    public static void main(String [] argv) {  
        Sole s = new Sole("bianco");  
        System.out.println("Stella: " + s.nome);  
        System.out.println("Colore: "  
            + s.colore);  
    }  
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Stella: Sole
Colore: rosa
 - D. Stella: Sole
Colore: bianco
 - E. Nessuna delle precedenti
-

7. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
 - B. Un attributo **static** non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
 - C. Un attributo non può essere contemporaneamente **static** e **final**
 - D. L'operatore **new** non restituisce un riferimento all'oggetto appena creato
 - E. Un oggetto può non esistere dopo la sua dichiarazione
-

8. Qual è l'output di questo codice?

```
class MyExcl extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends MyExc2 { }  
public class D1 {  
    public static void main(String [] argv)  
        throws Exception {
```

```

    try {
        q();
    }
    finally {
        throw( new MyExc2() );
    }
}
static void q() throws Exception {
    try {
    }
    catch( MyExc2 y ) {
        System.out.print(1);
        throw( new MyExc3() );
    }
    catch( MyExc3 g ) {
        System.out.print(2);
    }
    finally {
        throw( new MyExc3() );
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new

```

```

        ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
    }
}

```

```

catch( MyExc3 s ) {
    System.out.print(2);
    throw( new Exception() );
}
finally {
    System.out.print(3);
    throw( new MyExc3() );
}
}
static void n() throws Exception {
    try {
        System.out.print(4);
        switch(1) {
            case 1: throw( new MyExc2() );
            case 2: throw( new MyExc1() );
            case 3: throw( new MyExc3() );
        }
    }
    catch( MyExc1 t ) {
        throw( new MyExc3() );
    }
    catch( MyExc3 d ) {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception h ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
}
}
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

12. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

13. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

14. Quale output si ottiene invocando il metodo m?

```

class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

A. 31

16. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia(){
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}
```

- A. 0
 - B. 1
 - C. 2
 - D. 4
 - E. 5
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 37

1. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

2. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma (){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
```

```
ProgrammatoreJava pj=new
    ProgrammatoreJava("Franco", "Bianchi",5);
pj.stampa();
}
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. 31
- B. Errore a tempo di compilazione

C. 312Exception in thread main MyExc2

D. 312

E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print();
    }
}
```

A. Errore a tempo di compilazione

B. Errore a tempo di esecuzione

C. Cono: 8 - 3 lati per faccia

D. Cono: 3 - 3 lati per faccia

E. Nessuna delle precedenti

5. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
```

```
Bottiglia vinoBianco=new Bottiglia(1.0);
    spumante=null;
    vinoBianco=spumante;
    /*QUI*/
    System.out.println("END");
}
}
```

A. 0

B. 1

C. 2

D. 4

E. 5

6. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}
```

A. Errore a tempo di compilazione

B. Errore a tempo di esecuzione

C. Cono: 8 - 3 lati per faccia

D. Cono: 3 - 3 lati per faccia

E. Nessuna delle precedenti

7. Dire quale delle seguenti affermazioni è vera:

A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi

B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza

C. Un attributo non può essere contemporaneamente static e final

- D. L'operatore `new` non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

8. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `d = c;`
- B. `d = (Object [] []) e;`
- C. `c = (String []) d;`
- D. `d = (Object [] []) c;`
- E. `d = e;`

9. Quale output si ottiene invocando il metodo `m`?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

10. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
    }
}
```

```

    System.out.println(" anni di esperienza");
}

ProgrammatoreJava(String n, String c, int a){
super(n,c,a);
}
public static void main(String [] argv) {
    ProgrammatoreJava pj=new
        ProgrammatoreJava("Franco", "Bianchi",5);
    pj.stampa();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
System.out.println("Pioppo");
int num=n+1;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {

```

```

            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
    }
}

```

```

    catch( MyExc1 t ) {
        throw( new MyExc3() );
    }
    catch( MyExc3 d ) {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception h ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

15. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}

```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Pioppo
- 6

D. Pioppo
1

E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 38

1. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

2. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print ();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;
```

```
public void print () {
    System.out.println("Cono: "
        + dimensioni + " - " +
        numLati + " lati per faccia.");
}

public static void main(String [] argv) {
    Cono c1 = new Cono();
    FiguraGeometrica fl =c1;
    c1.print ();
}
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione

- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

4. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

5. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
    super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. 31
 - B. Errore a tempo di compilazione
 - C. 312Exception in thread main MyExc2
 - D. 312
 - E. Nessuna delle precedenti
-

8. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Pioppo
6
 - D. Pioppo
1
 - E. Nessuna delle precedenti
-

9. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Cono: 8 - 3 lati per faccia
 - D. Cono: 3 - 3 lati per faccia
 - E. Nessuna delle precedenti
-

10. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

public class ProgrammatoreJava
```

```

        extends Programmatore {
void stampa() {
    System.out.print("Programmatore Java con ");
    System.out.print(anniEsperienza);
    System.out.println(" anni di esperienza");
}
ProgrammatoreJava(String n, String c, int a){
super(n,c,a);
}
public static void main(String [] argv) {
    ProgrammatoreJava pj=new
    ProgrammatoreJava("Franco", "Bianchi",5);
    pj.stampa();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

11. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

```

}
static void q() throws Exception {
    try {
    }
    catch( MyExc2 y ) {
        System.out.print(1);
        throw( new MyExc3() );
    }
    catch( MyExc3 g ) {
        System.out.print(2);
    }
    finally {
        throw( new MyExc3() );
    }
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

14. Quale output si ottiene invocando il metodo m?

```

class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,

```

```

        Boolean b1, Object b2) {
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a3 == a4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

15. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo `static` non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente `static` e `final`
- D. L'operatore `new` non restituisce un riferimento all'oggetto appena creato

E. Un oggetto può non esistere dopo la sua dichiarazione

16. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 39

1. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1 = (FiguraGeometrica) f1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
}
```

```
abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }

    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }
}
```

```

public static void main(String [] argv) {
    Sole s = new Sole("bianco");
    System.out.println("Stella: " + s.nome);
    System.out.println("Colore: "
        + s.colore);
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

5. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri) {
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

6. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
}

```

```

public static void main(String [] argv) {
    ProgrammatoreJava pj=new
        ProgrammatoreJava("Franco", "Bianchi",5);
    pj.stampa();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

9. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

10. Qual è l'output di questo codice?


```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
}

```

```

}
public static void main(String args[]) {
    Pioppo p= new Pioppo();
    System.out.println(p.f(5));
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

12. Quale output si ottiene invocando il metodo m?

```

class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

13. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print ();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print () {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print ();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

15. Date le dichiarazioni:

```

Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

16. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 40

1. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

2. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia(){
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }
}
```

```
public static void main(String args[]) {
    Bottiglia spumante=new Bottiglia();
    Bottiglia vinoRosso=new Bottiglia();
    Bottiglia vinoBianco=new Bottiglia(1.0);
    spumante=null;
    vinoBianco=spumante;
    /*QUI*/
    System.out.println("END");
}
}
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

3. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
}
```

```

}
public static void main(String [] argv) {
    ProgrammatoreJava pj=new
        ProgrammatoreJava("Franco", "Bianchi",5);
    pj.stampa();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

6. Date le dichiarazioni:

```

Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;

- B. q = (Error []) w;
 - C. m = (Object [] []) w;
 - D. m = (Object [] []) q;
 - E. Nessuno dei precedenti
-

7. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
 - B. 100
 - C. 011
 - D. 101
 - E. 010
-

8. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

9. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n, c, a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Programmatore Java con 5 anni di esperienza
 - D. Programmatore Java con 0 anni di esperienza
 - E. Nessuna delle precedenti
-

10. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
 - B. d = (Object [] []) e;
 - C. c = (String []) d;
 - D. d = (Object [] []) c;
 - E. d = e;
-

11. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
System.out.println("Pioppo");
int num=n+1;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco

E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
System.out.println("Pioppo");
int num=6;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Pioppo
6
 - D. Pioppo
1
 - E. Nessuna delle precedenti
-

15. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa

- D. Stella: Sole
Colore: bianco
 - E. Nessuna delle precedenti
-

16. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1 = (FiguraGeometrica) f1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Cono: 8 - 3 lati per faccia
 - D. Cono: 3 - 3 lati per faccia
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 41

1. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
System.out.println("Pioppo");
int num=n+1;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){
```

```
System.out.print("Programmatore Java con ");
System.out.print (anniEsperienza);
System.out.println(" anni di esperienza");
}

ProgrammatoreJava(String n, String c, int a){
super(n,c,a);
}
public static void main(String [] argv) {
    ProgrammatoreJava pj=new
        ProgrammatoreJava("Franco", "Bianchi",5);
    pj.stampa();
}
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
```

```

    }
    finally {
        throw( new MyExc3() );
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
System.out.println("Pioppo");
int num=6;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {

```

```

        System.out.print(3);
        throw( new MyExc3() );
    }
}
static void n() throws Exception {
    try {
        System.out.print(4);
        switch(1) {
            case 1: throw( new MyExc2() );
            case 2: throw( new MyExc1() );
            case 3: throw( new MyExc3() );
        }
    }
    catch( MyExc1 t ) {
        throw( new MyExc3() );
    }
    catch( MyExc3 d ) {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception h ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

6. Quale output si ottiene invocando il metodo m?

```

class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```

    }
}
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

7. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print ();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print () {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print ();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

8. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato

E. Un oggetto può non esistere dopo la sua dichiarazione

9. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

10. Date le dichiarazioni:

```

Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

11. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione

- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

13. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

14. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri) {
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

15. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Programmatore Java con 5 anni di esperienza
 - D. Programmatore Java con 0 anni di esperienza
 - E. Nessuna delle precedenti
-

16. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. 31
 - B. Errore a tempo di compilazione
 - C. 312Exception in thread main MyExc2
 - D. 312
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 42

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
    }
}
```

```

}
catch( MyExc3 i ) {
    System.out.print(1);
}
catch( MyExc2 w ) {
}
catch( MyExc1 j ) {
    throw( new MyExc3() );
}
finally {
    System.out.print(2);
    throw( new MyExc2() );
}
}
static void q() {
    try {
        System.out.print(3);
        throw( new Exception() );
    }
    finally {
        throw( new MyExc3() );
    }
}
}
```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}
```

```
class Cono implements FiguraGeometrica {
    static final int numLati = 3;
    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +

```



```

        numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica fl =c1;
        c1.print();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

5. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi

- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

6. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

7. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;

E. d = e;

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

9. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
```

```
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

10. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri) {
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

11. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}
```

```
class Cono implements FiguraGeometrica {
```

```

static final int numLati = 3;

public void print() {
    System.out.println("Cono: "
        + dimensioni + " - " +
        numLati + " lati per faccia.");
}

public static void main(String [] argv) {
    Cono c1 = new Cono();
    FiguraGeometrica f1;
    c1= (FiguraGeometrica) f1;
    c1.print();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {

```

```

System.out.println("Pioppo");
int num=n+1;
return num;
}
public static void main(String args[]) {
    Pioppo p= new Pioppo();
    System.out.println(p.f(5));
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

14. Quale output si ottiene invocando il metodo m?

```

class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

15. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n, c, a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava ("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Programmatore Java con 5 anni di esperienza
 - D. Programmatore Java con 0 anni di esperienza
 - E. Nessuna delle precedenti
-

16. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n, c, a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava ("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Programmatore Java con 5 anni di esperienza
 - D. Programmatore Java con 0 anni di esperienza
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 43

1. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
```

```
String tipo="Sempre Verde";
public int f(int n)
{
    System.out.println("Pioppo");
    int num=n+1;
    return num;
}
public static void main(String args[] ) {
    Pioppo p= new Pioppo();
    System.out.println(p.f(5));
}
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
```

```

        switch(1) {
            case 1: throw( new MyExc2() );
            case 2: throw( new MyExc1() );
            case 3: throw( new MyExc3() );
        }
    }
    catch( MyExc1 t ) {
        throw( new MyExc3() );
    }
    catch( MyExc3 d ) {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception h ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
}
}
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. Errore a tempo di compilazione

- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

6. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato

E. Un oggetto può non esistere dopo la sua dichiarazione

7. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
```

```
+ s.colore);
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

9. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

10. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

11. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print ();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print () {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print ();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

13. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri) {
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

14. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
    }
}
```

```

    this.colore = c;
}

public static void main(String [] argv) {
    Sole s = new Sole("bianco");
    System.out.println("Stella: " + s.nome);
    System.out.println("Colore: "
        + s.colore);
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

15. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {

```

```

        System.out.print(2);
        throw( new MyExc2() );
    }
}
static void q() {
    try {
        System.out.print(3);
        throw( new Exception() );
    }
    finally {
        throw( new MyExc3() );
    }
}
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

16. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 44

1. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma() {}
    ProgrammatoreJava(String n, String c, int a){
    super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;
```

```
public Programmatore(String n, String c,
                      int a){
    nome=n;
    cognome=c;
    anniEsperienza=a;
}
abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
    super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}
```

```
public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

4. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri) {
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

5. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[] ) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

6. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
    }
}
```

```

    }
    catch( MyExc3 s ) {
        System.out.print(2);
        throw( new Exception() );
    }
    finally {
        System.out.print(3);
        throw( new MyExc3() );
    }
}
static void n() throws Exception {
    try {
        System.out.print(4);
        switch(1) {
            case 1: throw( new MyExc2() );
            case 2: throw( new MyExc1() );
            case 3: throw( new MyExc3() );
        }
    }
    catch( MyExc1 t ) {
        throw( new MyExc3() );
    }
    catch( MyExc3 d ) {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception h ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

```

```

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {

```

```

        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

10. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final

- D. L'operatore `new` non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

11. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
System.out.println("Pioppo");
int num=6;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
```

```
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

13. Quale output si ottiene invocando il metodo `m`?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

14. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {
```

```

static final int numLati = 3;

public void print() {
    System.out.println("Cono: "
        + dimensioni + " - " +
        numLati + " lati per faccia.");
}

public static void main(String [] argv) {
    Cono c1 = new Cono();
    FiguraGeometrica f1;
    c1= (FiguraGeometrica) f1;
    c1.print();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

15. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

```

static void q() throws Exception {
    try {
    }
    catch( MyExc2 y ) {
        System.out.print(1);
        throw( new MyExc3() );
    }
    catch( MyExc3 g ) {
        System.out.print(2);
    }
    finally {
        throw( new MyExc3() );
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

16. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 45

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)

E. Nessuna delle precedenti

2. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

3. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia(){
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

5. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;

7. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}
```

```
public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print ();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print () {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print ();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma ();
}

public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma () {}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
    }
}
```

```

    pj.stampa();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

```

```

}
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}
```

```
class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

15. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato

E. Un oggetto può non esistere dopo la sua dichiarazione

16. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 46

1. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo `static` non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente `static` e `final`
- D. L'operatore `new` non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

- C. 011
- D. 101
- E. 010

2. Quale output si ottiene invocando il metodo `m`?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 100

3. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `d = c;`
- B. `d = (Object [] []) e;`
- C. `c = (String []) d;`
- D. `d = (Object [] []) c;`
- E. `d = e;`

4. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {
    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1 = (FiguraGeometrica) f1;
        c1.print();
    }
}
```


- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
```

```
System.out.println("Pioppo");
int num=6;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

7. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia(){
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

8. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. 31
 - B. Errore a tempo di compilazione
 - C. 312Exception in thread main MyExc2
 - D. 312
 - E. Nessuna delle precedenti
-

12. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Pioppo
6
 - D. Pioppo
1
 - E. Nessuna delle precedenti
-

13. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Programmatore Java con 5 anni di esperienza
 - D. Programmatore Java con 0 anni di esperienza
 - E. Nessuna delle precedenti
-

14. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `q = (Error []) m;`
- B. `q = (Error []) w;`
- C. `m = (Object [] []) w;`
- D. `m = (Object [] []) q;`
- E. Nessuno dei precedenti

15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Cono: 8 - 3 lati per faccia
 - D. Cono: 3 - 3 lati per faccia
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 47

1. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
    }
}
```

```
catch( MyExc3 s ) {
    System.out.print(2);
    throw( new Exception() );
}
finally {
    System.out.print(3);
    throw( new MyExc3() );
}
}
static void n() throws Exception {
    try {
        System.out.print(4);
        switch(1) {
            case 1: throw( new MyExc2() );
            case 2: throw( new MyExc1() );
            case 3: throw( new MyExc3() );
        }
    }
    catch( MyExc1 t ) {
        throw( new MyExc3() );
    }
    catch( MyExc3 d ) {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception h ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
}
}
```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
    }
}
```

```

int num=n+1;
return num;
}
public static void main(String args[]) {
    Pioppo p= new Pioppo();
    System.out.println(p.f(5));
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

7. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
```

```

        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

9. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

10. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia(){
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[] ) {
```



```

Bottiglia spumante=new Bottiglia();
Bottiglia vinoRosso=new Bottiglia();
Bottiglia vinoBianco=new Bottiglia(1.0);
spumante=null;
vinoBianco=spumante;
/*QUI*/
System.out.println("END");
}
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

12. Date le dichiarazioni:

```

Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

13. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

14. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

15. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
super(n, c, a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

```
}
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Pioppo
6
 - D. Pioppo
1
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 48

1. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
System.out.println("Pioppo");
int num=6;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

2. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```
if(b2 == b1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

3. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

4. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
```

```

void stampa() {

    System.out.print("Programmatore Java con ");
    System.out.print(anniEsperienza);
    System.out.println(" anni di esperienza");
}

ProgrammatoreJava(String n, String c, int a) {
super(n, c, a);
}
public static void main(String [] argv) {
    ProgrammatoreJava pj=new
        ProgrammatoreJava("Franco", "Bianchi",5);
    pj.stampa();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class Bl {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

6. Date le dichiarazioni:

```

Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

7. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

8. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

9. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

10. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}
```

```
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }

    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
    }
}
```

```

    }
    catch( MyExc3 d ) {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception h ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";
}

```

```

public Stella(String nome) {
    this.nome = nome;
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
    }
}

```

```
    pj.stampa();
}
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Programmatore Java con 5 anni di esperienza
 - D. Programmatore Java con 0 anni di esperienza
 - E. Nessuna delle precedenti
-

15. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia

- D. Cono: 3 - 3 lati per faccia
 - E. Nessuna delle precedenti
-

16. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. Errore a tempo di compilazione
 - B. 111111... (ciclo infinito)
 - C. Exception in thread main MyExc2
 - D. 1111
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 49

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
```

```
{
    System.out.println("Pioppo");
    int num=n+1;
    return num;
}
public static void main(String args[] ) {
    Pioppo p= new Pioppo();
    System.out.println(p.f(5));
}
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
```

```

}
public static void main(String [] argv) {
    ProgrammatoreJava pj=new
        ProgrammatoreJava ("Franco", "Bianchi",5);
    pj.stampa ();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

4. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

5. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

6. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

```

```

}
public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

7. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

8. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

9. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

10. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica fl =c1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;

```

```

        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione

- B. Errore a tempo di esecuzione
 - C. Stella: Sole
Colore: rosa
 - D. Stella: Sole
Colore: bianco
 - E. Nessuna delle precedenti
-

15. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. Errore a tempo di compilazione
 - B. 111111... (ciclo infinito)
 - C. Exception in thread main MyExc2
 - D. 1111
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 50

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)

E. Nessuna delle precedenti

2. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

3. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
```



```

ProgrammatoreJava pj=new
  ProgrammatoreJava("Franco", "Bianchi",5);
pj.stampa();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

5. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

6. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {
    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

7. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

```

```

public Sole(String c) {
    super("Sole");
    this.colore = c;
}

public static void main(String [] argv) {
    Sole s = new Sole("bianco");
    System.out.println("Stella: " + s.nome);
    System.out.println("Colore: "
        + s.colore);
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

```

```

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

10. Quale output si ottiene invocando il metodo m?

```

class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```
}  
}
```

- A. 110
 - B. 100
 - C. 011
 - D. 101
 - E. 010
-

11. Qual è l'output di questo codice?

```
abstract class Programmatore {  
    String nome;  
    String cognome;  
    int anniEsperienza;  
  
    public Programmatore(String n, String c,  
                          int a){  
        nome=n;  
        cognome=c;  
        anniEsperienza=a;  
    }  
    abstract void programma();  
}  
public class ProgrammatoreJava  
    extends Programmatore {  
    void stampa(){  
  
        System.out.print("Programmatore Java con ");  
        System.out.print (anniEsperienza);  
        System.out.println(" anni di esperienza");  
    }  
    void programma(){}  
    ProgrammatoreJava(String n, String c, int a){  
        super(n,c,a);  
    }  
    public static void main(String [] argv) {  
        ProgrammatoreJava pj=new  
            ProgrammatoreJava("Franco", "Bianchi",5);  
        pj.stampa();  
    }  
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Programmatore Java con 5 anni di esperienza
 - D. Programmatore Java con 0 anni di esperienza
 - E. Nessuna delle precedenti
-

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends MyExc2 { }  
public class D1 {
```

```
public static void main(String [] argv)  
    throws Exception {  
    try {  
        q();  
    }  
    finally {  
        throw( new MyExc2() );  
    }  
}  
static void q() throws Exception {  
    try {  
    }  
    catch( MyExc2 y ) {  
        System.out.print(1);  
        throw( new MyExc3() );  
    }  
    catch( MyExc3 g ) {  
        System.out.print(2);  
    }  
    finally {  
        throw( new MyExc3() );  
    }  
}
```

- A. Errore a tempo di compilazione
 - B. 111111... (ciclo infinito)
 - C. Exception in thread main MyExc2
 - D. 1111
 - E. Nessuna delle precedenti
-

13. Qual è l'output di questo codice?

```
interface albero { int f(int i);}  
class Pioppo implements albero{  
    String tipo="Sempre Verde";  
    public int f()  
    {  
        System.out.println("Pioppo");  
        int num=6;  
        return num;  
    }  
    public static void main(String args[]) {  
        Pioppo p= new Pioppo();  
        System.out.println(p.f());  
    }  
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Pioppo
6
 - D. Pioppo
1
 - E. Nessuna delle precedenti
-

14. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri) {
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

15. Date le dichiarazioni:

```

Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

16. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 51

1. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
System.out.println("Pioppo");
int num=n+1;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
```

```
System.out.println("Stella: " + s.nome);
System.out.println("Colore: "
    + s.colore);
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

3. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri) {
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;
}
```

```
public Programmatore(String n, String c,
                    int a){
    nome=n;
    cognome=c;
    anniEsperienza=a;
}
abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
}
```

```

ProgrammatoreJava(String n, String c, int a){
super(n,c,a);
}
public static void main(String [] argv) {
ProgrammatoreJava pj=new
ProgrammatoreJava("Franco", "Bianchi",5);
pj.stampa();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

7. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
String tipo="Sempre Verde";
public int f()
{
System.out.println("Pioppo");
int num=6;
return num;
}
public static void main(String args[]) {
Pioppo p= new Pioppo();
System.out.println(p.f());
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

8. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;

- D. d = (Object [] []) c;
- E. d = e;

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
public static void main(String [] argv)
throws Exception {
try {
q();
}
finally {
throw( new MyExc2() );
}
}
static void q() throws Exception {
try {
}
catch( MyExc2 y ) {
System.out.print(1);
throw( new MyExc3() );
}
catch( MyExc3 g ) {
System.out.print(2);
}
finally {
throw( new MyExc3() );
}
}
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

10. Quale output si ottiene invocando il metodo m?

```

class F {
private String [] a1 = new String [7];
private String [] a4 = new String [2];
private Boolean b3 = new Boolean(true);
void m() {
q(a1, new String [2], b3, b3);
}
void q(Object a2, String [] a3,
Boolean b1, Object b2) {
if(a2 == a1) {
System.out.print(1);
} else {
System.out.print(0);
}
if(a3 == a4) {
System.out.print(1);
}
}
}

```



```

    } else {
        System.out.print(0);
    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

11. Date le dichiarazioni:

```

Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `q = (Error []) m;`
- B. `q = (Error []) w;`
- C. `m = (Object [] []) w;`
- D. `m = (Object [] []) q;`
- E. Nessuno dei precedenti

12. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");

```

```

        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

13. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo `static` non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente `static` e `final`
- D. L'operatore `new` non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

14. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione

- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

A. 31

16. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 52

1. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma () {}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

2. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

3. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione

- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

4. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri) {
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

5. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print ();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print () {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }
}
```

```
}

public static void main(String [] argv) {
    Cono c1 = new Cono();
    FiguraGeometrica f1 =c1;
    c1.print ();
}
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

6. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

7. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print ();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print () {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
    }
}
```

```

        cl.print();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {

```

```

        this.nome = nome;
    }
}

```

```

public class Sole extends Stella {

```

```

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

```

```

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco

E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Pioppo
6
 - D. Pioppo
1
 - E. Nessuna delle precedenti
-

12. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Pioppo
6
 - D. Pioppo
1
 - E. Nessuna delle precedenti
-

13. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
 - B. 100
 - C. 011
 - D. 101
 - E. 010
-

14. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
        }
    }
}
```

```

        throw( new Exception() );
    }
    finally {
        throw( new MyExc3() );
    }
}
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

15. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {

```

```

            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 53

1. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print ();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;
```

```
public void print () {
    System.out.println("Cono: "
        + dimensioni + " - " +
        numLati + " lati per faccia.");
}

public static void main(String [] argv) {
    Cono c1 = new Cono();
    FiguraGeometrica f1 =c1;
    c1.print ();
}
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma ();
}

public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
}
```

```

}
public static void main(String [] argv) {
    ProgrammatoreJava pj=new
        ProgrammatoreJava ("Franco", "Bianchi",5);
    pj.stampa ();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print (1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print (2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print (3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print (1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print (2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print ();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print () {
        System.out.println ("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
    }
}

```

```

    cl= (FiguraGeometrica) f1;
    cl.print();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

7. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `d = c;`
- B. `d = (Object [] []) e;`
- C. `c = (String []) d;`
- D. `d = (Object [] []) c;`
- E. `d = e;`

8. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
System.out.println("Pioppo");
int num=n+1;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

9. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo `static` non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente `static` e `final`
- D. L'operatore `new` non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

10. Quale output si ottiene invocando il metodo `m`?

```

class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

11. Date le dichiarazioni:

```

Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

12. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```
interface albero { int f(int i); }
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
```

```
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

15. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}

```

A. Errore a tempo di compilazione

- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

16. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia(){
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}

```

- A. 0
 - B. 1
 - C. 2
 - D. 4
 - E. 5
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 54

1. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
```

```
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}
```

```
public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }
}
```

```
public static void main(String [] argv) {
    Sole s = new Sole("bianco");
    System.out.println("Stella: " + s.nome);
}
```



```

System.out.println("Colore: "
    + s.colore);
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
System.out.println("Pioppo");
int num=n+1;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
    }
}

```

```

finally {
    System.out.print(3);
    throw( new MyExc3() );
}
}
static void n() throws Exception {
    try {
        System.out.print(4);
        switch(1) {
            case 1: throw( new MyExc2() );
            case 2: throw( new MyExc1() );
            case 3: throw( new MyExc3() );
        }
    }
    catch( MyExc1 t ) {
        throw( new MyExc3() );
    }
    catch( MyExc3 d ) {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception h ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {
    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}

```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Cono: 8 - 3 lati per faccia
 - D. Cono: 3 - 3 lati per faccia
 - E. Nessuna delle precedenti
-

7. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
 - B. Un attributo `static` non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
 - C. Un attributo non può essere contemporaneamente `static` e `final`
 - D. L'operatore `new` non restituisce un riferimento all'oggetto appena creato
 - E. Un oggetto può non esistere dopo la sua dichiarazione
-

8. Date le dichiarazioni:

```
String [] c;  
Object [] [] d;  
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `d = c;`
 - B. `d = (Object [] []) e;`
 - C. `c = (String []) d;`
 - D. `d = (Object [] []) c;`
 - E. `d = e;`
-

9. Qual è l'output di questo codice?

```
class Stella {  
    String nome="Stella";  
    String colore="rosa";  
  
    public Stella(String nome) {  
        this.nome = nome;  
    }  
}  
  
public class Sole extends Stella {  
  
    public Sole(String c) {  
        super("Sole");  
        this.colore = c;  
    }  
}
```

```
public static void main(String [] argv) {  
    Sole s = new Sole("bianco");  
    System.out.println("Stella: " + s.nome);  
    System.out.println("Colore: "  
        + s.colore);  
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Stella: Sole
Colore: rosa
 - D. Stella: Sole
Colore: bianco
 - E. Nessuna delle precedenti
-

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class B1 {  
    public static void main(String [] argv) {  
        try {  
            q();  
        }  
        catch( MyExc3 i ) {  
            System.out.print(1);  
        }  
        catch( MyExc2 w ) {  
        }  
        catch( MyExc1 j ) {  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(2);  
            throw( new MyExc2() );  
        }  
    }  
    static void q() {  
        try {  
            System.out.print(3);  
            throw( new Exception() );  
        }  
        finally {  
            throw( new MyExc3() );  
        }  
    }  
}
```

- A. 31
 - B. Errore a tempo di compilazione
 - C. 312Exception in thread main MyExc2
 - D. 312
 - E. Nessuna delle precedenti
-

11. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Programmatore Java con 5 anni di esperienza
 - D. Programmatore Java con 0 anni di esperienza
 - E. Nessuna delle precedenti
-

12. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }
}
```

```
}

public static void main(String [] argv) {
    Cono c1 = new Cono();
    FiguraGeometrica f1;
    c1= (FiguraGeometrica) f1;
    c1.print();
}
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Cono: 8 - 3 lati per faccia
 - D. Cono: 3 - 3 lati per faccia
 - E. Nessuna delle precedenti
-

13. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
 - B. q = (Error []) w;
 - C. m = (Object [] []) w;
 - D. m = (Object [] []) q;
 - E. Nessuno dei precedenti
-

14. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
          Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
```

```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

15. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 55

1. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
interface albero { int f(int i); }
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[] ) {
```

```
Pioppo p= new Pioppo();
System.out.println(p.f(5));
}
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
    }
}
```

```

    }
    catch( MyExc3 d ) {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception h ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
System.out.println("Pioppo");
int num=6;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

5. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia(){
        System.out.println(marca);
    }

    Bottiglia(double litri){

```

```

        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2

D. 312

E. Nessuna delle precedenti

7. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
 - B. Un attributo `static` non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
 - C. Un attributo non può essere contemporaneamente `static` e `final`
 - D. L'operatore `new` non restituisce un riferimento all'oggetto appena creato
 - E. Un oggetto può non esistere dopo la sua dichiarazione
-

8. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono cl = new Cono();
        FiguraGeometrica fl =cl;
        cl.print();
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Cono: 8 - 3 lati per faccia
 - D. Cono: 3 - 3 lati per faccia
 - E. Nessuna delle precedenti
-

9. Quale output si ottiene invocando il metodo `m`?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
 - B. 100
 - C. 011
 - D. 101
 - E. 010
-

10. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
}
```



```

ProgrammatoreJava(String n, String c, int a){
super(n,c,a);
}
public static void main(String [] argv) {
ProgrammatoreJava pj=new
ProgrammatoreJava("Franco", "Bianchi",5);
pj.stampa();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

11. Date le dichiarazioni:

```

Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

12. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

13. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
static int dimensioni = 10;
void print();
}

class Cono implements FiguraGeometrica {

static final int numLati = 3;

public void print() {
System.out.println("Cono: "
+ dimensioni + " - " +
numLati + " lati per faccia.");
}

public static void main(String [] argv) {
Cono c1 = new Cono();
FiguraGeometrica f1;
c1= (FiguraGeometrica) f1;
c1.print();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```

abstract class Programmatore {
String nome;
String cognome;
int anniEsperienza;

public Programmatore(String n, String c,
int a){
nome=n;
cognome=c;
anniEsperienza=a;
}
abstract void programma();
}

public class ProgrammatoreJava
extends Programmatore {
void stampa(){

System.out.print("Programmatore Java con ");
System.out.print(anniEsperienza);
System.out.println(" anni di esperienza");
}
void programma(){}
ProgrammatoreJava(String n, String c, int a){
super(n,c,a);
}
}

```

```

public static void main(String [] argv) {
    ProgrammatoreJava pj=new
        ProgrammatoreJava("Franco", "Bianchi",5);
    pj.stampa();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

15. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 56

1. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
```

```
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }

    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

3. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia(){
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[] ) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
    }
}
```

```

Bottiglia vinoBianco=new Bottiglia(1.0);
spumante=null;
vinoBianco=spumante;
/*QUI*/
System.out.println("END");
}
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

4. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

5. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

6. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

7. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;

- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

8. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

9. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
    }
}
```

```

catch( MyExc3 i ) {
    System.out.print(1);
}
catch( MyExc2 w ) {
}
catch( MyExc1 j ) {
    throw( new MyExc3() );
}
finally {
    System.out.print(2);
    throw( new MyExc2() );
}
}
static void q() {
    try {
        System.out.print(3);
        throw( new Exception() );
    }
    finally {
        throw( new MyExc3() );
    }
}
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
System.out.println("Pioppo");
int num=6;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

13. Quale output si ottiene invocando il metodo m?

```

class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

14. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
}

```

```

void programma() {}
ProgrammatoreJava(String n, String c, int a){
super(n, c, a);
}
public static void main(String [] argv) {
ProgrammatoreJava pj=new
ProgrammatoreJava("Franco", "Bianchi",5);
pj.stampa();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

15. Qual è l'output di questo codice?

```

class Stella {
String nome="Stella";
String colore="rosa";

public Stella(String nome) {
this.nome = nome;
}
}

public class Sole extends Stella {

public Sole(String c) {
super("Sole");
this.colore = c;
}

public static void main(String [] argv) {
Sole s = new Sole("bianco");
System.out.println("Stella: " + s.nome);
System.out.println("Colore: "
+ s.colore);
}
}

```

- A. Errore a tempo di compilazione

- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
static int dimensioni = 10;
void print();
}

```

```

class Cono implements FiguraGeometrica {

static final int numLati = 3;

public void print() {
System.out.println("Cono: "
+ dimensioni + " - " +
numLati + " lati per faccia.");
}

public static void main(String [] argv) {
Cono c1 = new Cono();
FiguraGeometrica f1;
c1= (FiguraGeometrica) f1;
c1.print();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 57

1. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo `static` non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente `static` e `final`
- D. L'operatore `new` non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

```
    pj.stampa();
}
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
```

3. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono cl = new Cono();
        FiguraGeometrica f1;
        cl= (FiguraGeometrica) f1;
        cl.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
super(n, c, a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
        ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);

```

```

        throw( new MyExc2() );
    }
}
static void q() {
    try {
        System.out.print(3);
        throw( new Exception() );
    }
    finally {
        throw( new MyExc3() );
    }
}
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

8. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia(){
        System.out.println(marca);
    }
}

```

```

Bottiglia(double litri){
    this.numlitri=litri;
    System.out.println(litri);
}

public static void main(String args[]) {
    Bottiglia spumante=new Bottiglia();
    Bottiglia vinoRosso=new Bottiglia();
    Bottiglia vinoBianco=new Bottiglia(1.0);
    spumante=null;
    vinoBianco=spumante;
    /*QUI*/
    System.out.println("END");
}
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

9. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

10. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

12. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
```

```
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
    if(a3 == a4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

13. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

14. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print ();
}
```

```
class Cono implements FiguraGeometrica {
    static final int numLati = 3;
    public void print () {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }
}
```

```

public static void main(String [] argv) {
    Cono c1 = new Cono();
    FiguraGeometrica fl =c1;
    c1.print();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

15. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
System.out.println("Pioppo");
int num=n+1;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1

E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 58

1. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
    }
    int num=6;
    return num;
}
public static void main(String args[]) {
    Pioppo p= new Pioppo();
    System.out.println(p.f());
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

2. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

3. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
```



```

}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa() {

        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma() {}
    ProgrammatoreJava(String n, String c, int a) {
    super(n, c, a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

7. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

```

```

class Cono implements FiguraGeometrica {

```

```

static final int numLati = 3;

public void print() {
    System.out.println("Cono: "
        + dimensioni + " - " +
        numLati + " lati per faccia.");
}

public static void main(String [] argv) {
    Cono c1 = new Cono();
    FiguraGeometrica f1;
    c1= (FiguraGeometrica) f1;
    c1.print();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 31

- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
    }
}

```

```

    this.colore = c;
}

public static void main(String [] argv) {
    Sole s = new Sole("bianco");
    System.out.println("Stella: " + s.nome);
    System.out.println("Colore: "
        + s.colore);
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

11. Quale output si ottiene invocando il metodo m?

```

class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

12. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n, c, a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

13. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

14. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri) {
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

15. Date le dichiarazioni:

```

Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

16. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 59

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

3. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n, c, a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza

2. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
}
```

```
}
static void n() throws Exception {
    try {
        System.out.print(4);
        switch(1) {
            case 1: throw( new MyExc2() );
            case 2: throw( new MyExc1() );
            case 3: throw( new MyExc3() );
        }
    }
    catch( MyExc1 t ) {
        throw( new MyExc3() );
    }
    catch( MyExc3 d ) {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception h ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
}
```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

6. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

7. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia(){
        System.out.println(marca);
    }
}
```

```

Bottiglia(double litri){
    this.numlitri=litri;
    System.out.println(litri);
}

public static void main(String args[]) {
    Bottiglia spumante=new Bottiglia();
    Bottiglia vinoRosso=new Bottiglia();
    Bottiglia vinoBianco=new Bottiglia(1.0);
    spumante=null;
    vinoBianco=spumante;
    /*QUI*/
    System.out.println("END");
}
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

8. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava ("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {

```



```

catch( MyExc2 y ) {
    System.out.print(1);
    throw( new MyExc3() );
}
catch( MyExc3 g ) {
    System.out.print(2);
}
finally {
    throw( new MyExc3() );
}
}
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

11. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo **static** non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente **static** e **final**
- D. L'operatore **new** non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

12. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

```

```

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }
}

```

```

public static void main(String [] argv) {
    Cono c1 = new Cono();
    FiguraGeometrica f1;
}

```

```

c1= (FiguraGeometrica) f1;
c1.print();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

13. Quale output si ottiene invocando il metodo m?

```

class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

14. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
}

```

```

public static void main(String args[]) {
    Pioppo p= new Pioppo();
    System.out.println(p.f(5));
}
}

```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Pioppo
6
 - D. Pioppo
1
 - E. Nessuna delle precedenti
-

15. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
System.out.println("Pioppo");
int num=6;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1

E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Stella: Sole
Colore: rosa
 - D. Stella: Sole
Colore: bianco
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 60

1. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}
```

```
class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
}
```

```

    }
}
static void q() {
    try {
        System.out.print(3);
        throw( new Exception() );
    }
    finally {
        throw( new MyExc3() );
    }
}
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

4. Quale output si ottiene invocando il metodo m?

```

class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

5. Date le dichiarazioni:

```

Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

6. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

7. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

10. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia(){
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[] ) {
```

```

Bottiglia spumante=new Bottiglia();
Bottiglia vinoRosso=new Bottiglia();
Bottiglia vinoBianco=new Bottiglia(1.0);
spumante=null;
vinoBianco=spumante;
/*QUI*/
System.out.println("END");
}
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

11. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

E. d = e;

14. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
System.out.println("Pioppo");
int num=6;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Pioppo
6
 - D. Pioppo
1
 - E. Nessuna delle precedenti
-

15. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;

16. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. Errore a tempo di compilazione
 - B. 111111... (ciclo infinito)
 - C. Exception in thread main MyExc2
 - D. 1111
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 61

1. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
}
```

```
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

3. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

4. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
System.out.println("Pioppo");
int num=6;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
super(n, c, a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
        ProgrammatoreJava ("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

}

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

7. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1 = (FiguraGeometrica) f1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- ```
}

A. Errore a tempo di compilazione
B. Errore a tempo di esecuzione
C. Stella: Sole
 Colore: rosa
D. Stella: Sole
 Colore: bianco
E. Nessuna delle precedenti
```
- 
9. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
 static String marca="Ferrari";
 double numlitri=1.5;

 Bottiglia(){
 System.out.println(marca);
 }

 Bottiglia(double litri){
 this.numlitri=litri;
 System.out.println(litri);
 }

 public static void main(String args[]) {
 Bottiglia spumante=new Bottiglia();
 Bottiglia vinoRosso=new Bottiglia();
 Bottiglia vinoBianco=new Bottiglia(1.0);
 spumante=null;
 vinoBianco=spumante;
 /*QUI*/
 System.out.println("END");
 }
}
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

---

10. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;

- D. `d = (Object [] []) c;`
- E. `d = e;`

---

11. Quale output si ottiene invocando il metodo `m`?

```
class F {
 private String [] a1 = new String [7];
 private String [] a4 = new String [2];
 private Boolean b3 = new Boolean(true);
 void m() {
 q(a1, new String [2], b3, b3);
 }
 void q(Object a2, String [] a3,
 Boolean b1, Object b2) {
 if(a2 == a1) {
 System.out.print(1);
 } else {
 System.out.print(0);
 }
 if(a3 == a4) {
 System.out.print(1);
 } else {
 System.out.print(0);
 }
 if(b2 == b1) {
 System.out.print(1);
 } else {
 System.out.print(0);
 }
 }
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

---

12. Qual è l'output di questo codice?

```
interface albero { int f(int i); }
class Pioppo implements albero {
 String tipo="Sempre Verde";
 public int f(int n)
 {
 System.out.println("Pioppo");
 int num=n+1;
 return num;
 }
 public static void main(String args[]) {
 Pioppo p= new Pioppo();
 System.out.println(p.f(5));
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo  
6
- D. Pioppo  
1
- E. Nessuna delle precedenti

---

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
 public static void main(String [] argv)
 throws Exception {
 try {
 q();
 }
 finally {
 throw(new MyExc2());
 }
 }
 static void q() throws Exception {
 try {
 }
 catch(MyExc2 y) {
 System.out.print(1);
 throw(new MyExc3());
 }
 catch(MyExc3 g) {
 System.out.print(2);
 }
 finally {
 throw(new MyExc3());
 }
 }
}
```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

---

14. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo `static` non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente `static` e `final`
- D. L'operatore `new` non restituisce un riferimento all'oggetto appena creato

E. Un oggetto può non esistere dopo la sua dichiarazione

---

15. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
 static int dimensioni = 10;
 void print();
}

class Cono implements FiguraGeometrica {

 static final int numLati = 3;

 public void print() {
 System.out.println("Cono: "
 + dimensioni + " - " +
 numLati + " lati per faccia.");
 }

 public static void main(String [] argv) {
 Cono c1 = new Cono();
 FiguraGeometrica f1 =c1;
 c1.print();
 }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. Cono: 8 - 3 lati per faccia
  - D. Cono: 3 - 3 lati per faccia
  - E. Nessuna delle precedenti
- 

16. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
 public static void main(String [] argv) {
 try {
 q();
 }
 catch(MyExc3 i) {
 System.out.print(1);
 }
 catch(MyExc2 w) {
 }
 catch(MyExc1 j) {
 throw(new MyExc3());
 }
 finally {
 System.out.print(2);
 throw(new MyExc2());
 }
 }
 static void q() {
 try {
 System.out.print(3);
 throw(new Exception());
 }
 finally {
 throw(new MyExc3());
 }
 }
}
```

- A. 31
  - B. Errore a tempo di compilazione
  - C. 312Exception in thread main MyExc2
  - D. 312
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 62**

1. Qual è l'output di questo codice?

```
abstract class Programmatore {
 String nome;
 String cognome;
 int anniEsperienza;

 public Programmatore(String n, String c,
 int a){
 nome=n;
 cognome=c;
 anniEsperienza=a;
 }
 abstract void programma();
}
public class ProgrammatoreJava
 extends Programmatore {

 void stampa(){

 System.out.print("Programmatore Java con ");
 System.out.print (anniEsperienza);
 System.out.println(" anni di esperienza");
 }

 ProgrammatoreJava(String n, String c, int a){
 super(n,c,a);
 }
 public static void main(String [] argv) {
 ProgrammatoreJava pj=new
 ProgrammatoreJava("Franco", "Bianchi",5);
 pj.stampa();
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
```

```
class MyExc3 extends Exception { }
public class C1 {
 public static void main(String [] argv)
 throws Exception {
 try {
 System.out.print(1);
 n();
 }
 catch(MyExc3 s) {
 System.out.print(2);
 throw(new Exception());
 }
 finally {
 System.out.print(3);
 throw(new MyExc3());
 }
 }
 static void n() throws Exception {
 try {
 System.out.print(4);
 switch(1) {
 case 1: throw(new MyExc2());
 case 2: throw(new MyExc1());
 case 3: throw(new MyExc3());
 }
 }
 catch(MyExc1 t) {
 throw(new MyExc3());
 }
 catch(MyExc3 d) {
 System.out.print(5);
 throw(new MyExc2());
 }
 catch(Exception h) {
 System.out.print(6);
 throw(new MyExc2());
 }
 }
}
```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?



```

interface FiguraGeometrica {
 static int dimensioni = 10;
 void print();
}

class Cono implements FiguraGeometrica {

 static final int numLati = 3;

 public void print() {
 System.out.println("Cono: "
 + dimensioni + " - " +
 numLati + " lati per faccia.");
 }

 public static void main(String [] argv) {
 Cono c1 = new Cono();
 FiguraGeometrica f1 =c1;
 c1.print();
 }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

4. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo **static** non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente **static** e **final**
- D. L'operatore **new** non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
 public static void main(String [] argv)
 throws Exception {
 try {
 q();
 }
 }
}

```

```

 finally {
 throw(new MyExc2());
 }
 }
 static void q() throws Exception {
 try {
 }
 catch(MyExc2 y) {
 System.out.print(1);
 throw(new MyExc3());
 }
 catch(MyExc3 g) {
 System.out.print(2);
 }
 finally {
 throw(new MyExc3());
 }
 }
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

6. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `d = c;`
- B. `d = (Object [] []) e;`
- C. `c = (String []) d;`
- D. `d = (Object [] []) c;`
- E. `d = e;`

7. Date le dichiarazioni:

```

Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `q = (Error []) m;`
- B. `q = (Error []) w;`
- C. `m = (Object [] []) w;`
- D. `m = (Object [] []) q;`

## E. Nessuno dei precedenti

### 8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
 public static void main(String [] argv) {
 try {
 q();
 }
 catch(MyExc3 i) {
 System.out.print(1);
 }
 catch(MyExc2 w) {
 }
 catch(MyExc1 j) {
 throw(new MyExc3());
 }
 finally {
 System.out.print(2);
 throw(new MyExc2());
 }
 }
 static void q() {
 try {
 System.out.print(3);
 throw(new Exception());
 }
 finally {
 throw(new MyExc3());
 }
 }
}
```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

### 9. Qual è l'output di questo codice?

```
abstract class Programmatore {
 String nome;
 String cognome;
 int anniEsperienza;

 public Programmatore(String n, String c,
 int a){
 nome=n;
 cognome=c;
 anniEsperienza=a;
 }
 abstract void programma();
}
```

```
}
public class ProgrammatoreJava
 extends Programmatore {
 void stampa(){

 System.out.print("Programmatore Java con ");
 System.out.print (anniEsperienza);
 System.out.println(" anni di esperienza");
 }
 void programma(){}
 ProgrammatoreJava(String n, String c, int a){
 super(n,c,a);
 }
 public static void main(String [] argv) {
 ProgrammatoreJava pj=new
 ProgrammatoreJava("Franco", "Bianchi",5);
 pj.stampa();
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

### 10. Qual è l'output di questo codice?

```
class Stella {
 String nome="Stella";
 final String colore="rosa";

 public Stella(String nome) {
 this.nome = nome;
 }
}

public class Sole extends Stella {

 public Sole(String c) {
 super("Sole");
 this.colore = c;
 }

 public static void main(String [] argv) {
 Sole s = new Sole("bianco");
 System.out.println("Stella: " + s.nome);
 System.out.println("Colore: "
 + s.colore);
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole  
Colore: rosa
- D. Stella: Sole  
Colore: bianco
- E. Nessuna delle precedenti

- 
11. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
 static String marca="Ferrari";
 double numlitri=1.5;

 Bottiglia() {
 System.out.println(marca);
 }

 Bottiglia(double litri) {
 this.numlitri=litri;
 System.out.println(litri);
 }

 public static void main(String args[]) {
 Bottiglia spumante=new Bottiglia();
 Bottiglia vinoRosso=new Bottiglia();
 Bottiglia vinoBianco=new Bottiglia(1.0);
 spumante=null;
 vinoBianco=spumante;
 /*QUI*/
 System.out.println("END");
 }
}
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

- 
12. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
 String tipo="Sempre Verde";
 public int f()
 {
 System.out.println("Pioppo");
 int num=6;
 return num;
 }
 public static void main(String args[]) {
 Pioppo p= new Pioppo();
 System.out.println(p.f());
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo

6

D. Pioppo

1

E. Nessuna delle precedenti

- 
13. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
 static int dimensioni = 10;
 void print();
}

class Cono implements FiguraGeometrica {

 static final int numLati = 3;

 public void print() {
 System.out.println("Cono: "
 + dimensioni + " - " +
 numLati + " lati per faccia.");
 }

 public static void main(String [] argv) {
 Cono c1 = new Cono();
 FiguraGeometrica f1;
 c1= (FiguraGeometrica) f1;
 c1.print();
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

- 
14. Quale output si ottiene invocando il metodo m?

```
class F {
 private String [] a1 = new String [7];
 private String [] a4 = new String [2];
 private Boolean b3 = new Boolean(true);
 void m() {
 q(a1, new String [2], b3, b3);
 }
 void q(Object a2, String [] a3,
 Boolean b1, Object b2) {
 if(a2 == a1) {
 System.out.print(1);
 } else {
 System.out.print(0);
 }
 }
 if(a3 == a4) {
 System.out.print(1);
 } else {
```

```

 System.out.print(0);
 }
 if(b2 == b1) {
 System.out.print(1);
 } else {
 System.out.print(0);
 }
}
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

15. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
 String tipo="Sempre Verde";
 public int f(int n)
 {
System.out.println("Pioppo");
int num=n+1;
return num;
 }
 public static void main(String args[]) {
 Pioppo p= new Pioppo();
 System.out.println(p.f(5));
 }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo

6

D. Pioppo

1

E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```

class Stella {
 String nome="Stella";
 String colore="rosa";

 public Stella(String nome) {
 this.nome = nome;
 }
}

public class Sole extends Stella {

 public Sole(String c) {
 super("Sole");
 this.colore = c;
 }

 public static void main(String [] argv) {
 Sole s = new Sole("bianco");
 System.out.println("Stella: " + s.nome);
 System.out.println("Colore: "
 + s.colore);
 }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole  
Colore: rosa
- D. Stella: Sole  
Colore: bianco
- E. Nessuna delle precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 63**

1. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

2. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
 static int dimensioni = 10;
 void print();
}

class Cono implements FiguraGeometrica {

 static final int numLati = 3;

 public void print() {
 System.out.println("Cono: "
 + dimensioni + " - " +
 numLati + " lati per faccia.");
 }

 public static void main(String [] argv) {
 Cono cl = new Cono();
 FiguraGeometrica fl;
 cl = (FiguraGeometrica) fl;
 cl.print();
 }
}
```

- A. Errore a tempo di compilazione

- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
 static int dimensioni = 10;
 void print();
}

class Cono implements FiguraGeometrica {

 static final int numLati = 3;

 public void print() {
 System.out.println("Cono: "
 + dimensioni + " - " +
 numLati + " lati per faccia.");
 }

 public static void main(String [] argv) {
 Cono cl = new Cono();
 FiguraGeometrica fl = cl;
 cl.print();
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

4. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final

D. L'operatore `new` non restituisce un riferimento all'oggetto appena creato

E. Un oggetto può non esistere dopo la sua dichiarazione

---

5. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
 public static void main(String [] argv)
 throws Exception {
 try {
 System.out.print(1);
 n();
 }
 catch(MyExc3 s) {
 System.out.print(2);
 throw(new Exception());
 }
 finally {
 System.out.print(3);
 throw(new MyExc3());
 }
 }
 static void n() throws Exception {
 try {
 System.out.print(4);
 switch(1) {
 case 1: throw(new MyExc2());
 case 2: throw(new MyExc1());
 case 3: throw(new MyExc3());
 }
 }
 catch(MyExc1 t) {
 throw(new MyExc3());
 }
 catch(MyExc3 d) {
 System.out.print(5);
 throw(new MyExc2());
 }
 catch(Exception h) {
 System.out.print(6);
 throw(new MyExc2());
 }
 }
}
```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

---

6. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
 String tipo="Sempre Verde";
 public int f(int n)
 {
 System.out.println("Pioppo");
 int num=n+1;
 return num;
 }
 public static void main(String args[]) {
 Pioppo p= new Pioppo();
 System.out.println(p.f(5));
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo  
6
- D. Pioppo  
1
- E. Nessuna delle precedenti

---

7. Quale output si ottiene invocando il metodo `m`?

```
class F {
 private String [] a1 = new String [7];
 private String [] a4 = new String [2];
 private Boolean b3 = new Boolean(true);
 void m() {
 q(a1, new String [2], b3, b3);
 }
 void q(Object a2, String [] a3,
 Boolean b1, Object b2) {
 if(a2 == a1) {
 System.out.print(1);
 } else {
 System.out.print(0);
 }
 if(a3 == a4) {
 System.out.print(1);
 } else {
 System.out.print(0);
 }
 if(b2 == b1) {
 System.out.print(1);
 } else {
 System.out.print(0);
 }
 }
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

---

8. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
 String tipo="Sempre Verde";
 public int f()
 {
System.out.println("Pioppo");
int num=6;
return num;
 }
 public static void main(String args[]) {
 Pioppo p= new Pioppo();
 System.out.println(p.f());
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo  
6
- D. Pioppo  
1
- E. Nessuna delle precedenti

---

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
 public static void main(String [] argv)
 throws Exception {
 try {
 q();
 }
 finally {
 throw(new MyExc2());
 }
 }
 static void q() throws Exception {
 try {
 }
 catch(MyExc2 y) {
 System.out.print(1);
 throw(new MyExc3());
 }
 catch(MyExc3 g) {
 System.out.print(2);
 }
 finally {
 throw(new MyExc3());
 }
 }
}
```

- A. Errore a tempo di compilazione

- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

---

10. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

---

11. Qual è l'output di questo codice?

```
abstract class Programmatore {
 String nome;
 String cognome;
 int anniEsperienza;

 public Programmatore(String n, String c,
 int a){
 nome=n;
 cognome=c;
 anniEsperienza=a;
 }
 abstract void programma();
}
public class ProgrammatoreJava
 extends Programmatore {
 void stampa(){
 System.out.print("Programmatore Java con ");
 System.out.print (anniEsperienza);
 System.out.println(" anni di esperienza");
 }
 void programma(){}
 ProgrammatoreJava(String n, String c, int a){
 super(n,c,a);
 }
 public static void main(String [] argv) {
 ProgrammatoreJava pj=new
 ProgrammatoreJava("Franco", "Bianchi",5);
 pj.stampa();
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione



- C. Programmatore Java con 5 anni di esperienza
  - D. Programmatore Java con 0 anni di esperienza
  - E. Nessuna delle precedenti
- 

12. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
 static String marca="Ferrari";
 double numlitri=1.5;

 Bottiglia() {
 System.out.println(marca);
 }

 Bottiglia(double litri){
 this.numlitri=litri;
 System.out.println(litri);
 }

 public static void main(String args[]) {
 Bottiglia spumante=new Bottiglia();
 Bottiglia vinoRosso=new Bottiglia();
 Bottiglia vinoBianco=new Bottiglia(1.0);
 spumante=null;
 vinoBianco=spumante;
 /*QUI*/
 System.out.println("END");
 }
}
```

- A. 0
  - B. 1
  - C. 2
  - D. 4
  - E. 5
- 

13. Qual è l'output di questo codice?

```
class Stella {
 String nome="Stella";
 String colore="rosa";

 public Stella(String nome) {
 this.nome = nome;
 }
}

public class Sole extends Stella {

 public Sole(String c) {
 super("Sole");
 this.colore = c;
 }
}
```

```
public static void main(String [] argv) {
 Sole s = new Sole("bianco");
 System.out.println("Stella: " + s.nome);
 System.out.println("Colore: "
 + s.colore);
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. Stella: Sole  
Colore: rosa
  - D. Stella: Sole  
Colore: bianco
  - E. Nessuna delle precedenti
- 

14. Qual è l'output di questo codice?

```
class Stella {
 String nome="Stella";
 final String colore="rosa";

 public Stella(String nome) {
 this.nome = nome;
 }
}

public class Sole extends Stella {

 public Sole(String c) {
 super("Sole");
 this.colore = c;
 }

 public static void main(String [] argv) {
 Sole s = new Sole("bianco");
 System.out.println("Stella: " + s.nome);
 System.out.println("Colore: "
 + s.colore);
 }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. Stella: Sole  
Colore: rosa
  - D. Stella: Sole  
Colore: bianco
  - E. Nessuna delle precedenti
- 

15. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
 public static void main(String [] argv) {
 try {
 q();
 }
 catch(MyExc3 i) {
 System.out.print(1);
 }
 catch(MyExc2 w) {
 }
 catch(MyExc1 j) {
 throw(new MyExc3());
 }
 finally {
 System.out.print(2);
 throw(new MyExc2());
 }
 }
 static void q() {
 try {
 System.out.print(3);
 throw(new Exception());
 }
 finally {
 throw(new MyExc3());
 }
 }
}

```

- A. 31
  - B. Errore a tempo di compilazione
  - C. 312Exception in thread main MyExc2
  - D. 312
  - E. Nessuna delle precedenti
- 

16. Qual è l'output di questo codice?

```

abstract class Programmatore {
 String nome;
 String cognome;
 int anniEsperienza;

 public Programmatore(String n, String c,
 int a){
 nome=n;
 cognome=c;
 anniEsperienza=a;
 }
 abstract void programma();
}
public class ProgrammatoreJava
 extends Programmatore {

 void stampa(){

 System.out.print("Programmatore Java con ");
 System.out.print (anniEsperienza);
 System.out.println(" anni di esperienza");
 }

 ProgrammatoreJava(String n, String c, int a){
 super(n,c,a);
 }
 public static void main(String [] argv) {
 ProgrammatoreJava pj=new
 ProgrammatoreJava("Franco", "Bianchi",5);
 pj.stampa();
 }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. Programmatore Java con 5 anni di esperienza
  - D. Programmatore Java con 0 anni di esperienza
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 64**

1. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
 static int dimensioni = 10;
 void print();
}

class Cono implements FiguraGeometrica {

 static final int numLati = 3;

 public void print() {
 System.out.println("Cono: "
 + dimensioni + " - " +
 numLati + " lati per faccia.");
 }

 public static void main(String [] argv) {
 Cono c1 = new Cono();
 FiguraGeometrica f1 =c1;
 c1.print();
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

2. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
 static String marca="Ferrari";
 double numlitri=1.5;

 Bottiglia(){
 System.out.println(marca);
 }

 Bottiglia(double litri){
 this.numlitri=litri;
 System.out.println(litri);
 }
}
```

```
public static void main(String args[]) {
 Bottiglia spumante=new Bottiglia();
 Bottiglia vinoRosso=new Bottiglia();
 Bottiglia vinoBianco=new Bottiglia(1.0);
 spumante=null;
 vinoBianco=spumante;
 /*QUI*/
 System.out.println("END");
}
}
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
 public static void main(String [] argv)
 throws Exception {
 try {
 q();
 }
 finally {
 throw(new MyExc2());
 }
 }
 static void q() throws Exception {
 try {
 }
 catch(MyExc2 y) {
 System.out.print(1);
 throw(new MyExc3());
 }
 catch(MyExc3 g) {
 System.out.print(2);
 }
 finally {
 throw(new MyExc3());
 }
 }
}
```

```

 }
}
}

```

- A. Errore a tempo di compilazione
- B. 1111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
 String tipo="Sempre Verde";
 public int f(int n)
 {
System.out.println("Pioppo");
int num=n+1;
return num;
 }
 public static void main(String args[]) {
 Pioppo p= new Pioppo();
 System.out.println(p.f(5));
 }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo  
6
- D. Pioppo  
1
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
 public static void main(String [] argv) {
 try {
 q();
 }
 catch(MyExc3 i) {
 System.out.print(1);
 }
 catch(MyExc2 w) {
 }
 catch(MyExc1 j) {
 throw(new MyExc3());
 }
 finally {

```

```

 System.out.print(2);
 throw(new MyExc2());
 }
}
static void q() {
 try {
 System.out.print(3);
 throw(new Exception());
 }
 finally {
 throw(new MyExc3());
 }
}
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

6. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

7. Qual è l'output di questo codice?

```

class Stella {
 String nome="Stella";
 final String colore="rosa";

 public Stella(String nome) {
 this.nome = nome;
 }
}

public class Sole extends Stella {

 public Sole(String c) {
 super("Sole");
 this.colore = c;
 }

 public static void main(String [] argv) {

```

```

Sole s = new Sole("bianco");
System.out.println("Stella: " + s.nome);
System.out.println("Colore: "
 + s.colore);
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole  
Colore: rosa
- D. Stella: Sole  
Colore: bianco
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```

abstract class Programmatore {
 String nome;
 String cognome;
 int anniEsperienza;

 public Programmatore(String n, String c,
 int a){
 nome=n;
 cognome=c;
 anniEsperienza=a;
 }
 abstract void programma();
}
public class ProgrammatoreJava
 extends Programmatore {

 void stampa(){

 System.out.print("Programmatore Java con ");
 System.out.print (anniEsperienza);
 System.out.println(" anni di esperienza");
 }

 ProgrammatoreJava(String n, String c, int a){
super(n, c, a);
 }
 public static void main(String [] argv) {
 ProgrammatoreJava pj=new
 ProgrammatoreJava ("Franco", "Bianchi",5);
 pj.stampa();
 }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```

abstract class Programmatore {
 String nome;
 String cognome;
 int anniEsperienza;

 public Programmatore(String n, String c,
 int a){
 nome=n;
 cognome=c;
 anniEsperienza=a;
 }
 abstract void programma();
}
public class ProgrammatoreJava
 extends Programmatore {
 void stampa(){

 System.out.print("Programmatore Java con ");
 System.out.print (anniEsperienza);
 System.out.println(" anni di esperienza");
 }
 void programma(){}
 ProgrammatoreJava(String n, String c, int a){
super(n, c, a);
 }
 public static void main(String [] argv) {
 ProgrammatoreJava pj=new
 ProgrammatoreJava ("Franco", "Bianchi",5);
 pj.stampa();
 }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

10. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo **static** non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente **static** e **final**
- D. L'operatore **new** non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

11. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
 String tipo="Sempre Verde";
 public int f()
 {
System.out.println("Pioppo");
int num=6;
return num;
 }
 public static void main(String args[]) {
 Pioppo p= new Pioppo();
 System.out.println(p.f());
 }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo  
6
- D. Pioppo  
1
- E. Nessuna delle precedenti

12. Date le dichiarazioni:

```

Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
 public static void main(String [] argv)
 throws Exception {
 try {
 System.out.print(1);
 n();
 }
 catch(MyExc3 s) {
 System.out.print(2);

```

```

 throw(new Exception());
 }
 finally {
 System.out.print(3);
 throw(new MyExc3());
 }
}
static void n() throws Exception {
 try {
 System.out.print(4);
 switch(1) {
 case 1: throw(new MyExc2());
 case 2: throw(new MyExc1());
 case 3: throw(new MyExc3());
 }
 }
 catch(MyExc1 t) {
 throw(new MyExc3());
 }
 catch(MyExc3 d) {
 System.out.print(5);
 throw(new MyExc2());
 }
 catch(Exception h) {
 System.out.print(6);
 throw(new MyExc2());
 }
}
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```

class Stella {
 String nome="Stella";
 String colore="rosa";

 public Stella(String nome) {
 this.nome = nome;
 }
}

public class Sole extends Stella {

 public Sole(String c) {
 super("Sole");
 this.colore = c;
 }

 public static void main(String [] argv) {
 Sole s = new Sole("bianco");
 System.out.println("Stella: " + s.nome);
 System.out.println("Colore: "

```

```
 + s.colore);
 }
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. Stella: Sole  
Colore: rosa
  - D. Stella: Sole  
Colore: bianco
  - E. Nessuna delle precedenti
- 

15. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
 static int dimensioni = 10;
 void print();
}

class Cono implements FiguraGeometrica {

 static final int numLati = 3;

 public void print() {
 System.out.println("Cono: "
 + dimensioni + " - "
 + numLati + " lati per faccia.");
 }

 public static void main(String [] argv) {
 Cono c1 = new Cono();
 FiguraGeometrica f1;
 c1= (FiguraGeometrica) f1;
 c1.print();
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione

- C. Cono: 8 - 3 lati per faccia
  - D. Cono: 3 - 3 lati per faccia
  - E. Nessuna delle precedenti
- 

16. Quale output si ottiene invocando il metodo m?

```
class F {
 private String [] a1 = new String [7];
 private String [] a4 = new String [2];
 private Boolean b3 = new Boolean(true);
 void m() {
 q(a1, new String [2], b3, b3);
 }
 void q(Object a2, String [] a3,
 Boolean b1, Object b2) {
 if(a2 == a1) {
 System.out.print(1);
 } else {
 System.out.print(0);
 }
 if(a3 == a4) {
 System.out.print(1);
 } else {
 System.out.print(0);
 }
 if(b2 == b1) {
 System.out.print(1);
 } else {
 System.out.print(0);
 }
 }
}
```

- A. 110
  - B. 100
  - C. 011
  - D. 101
  - E. 010
-





Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 65**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
 public static void main(String [] argv)
 throws Exception {
 try {
 System.out.print(1);
 n();
 }
 catch(MyExc3 s) {
 System.out.print(2);
 throw(new Exception());
 }
 finally {
 System.out.print(3);
 throw(new MyExc3());
 }
 }
 static void n() throws Exception {
 try {
 System.out.print(4);
 switch(1) {
 case 1: throw(new MyExc2());
 case 2: throw(new MyExc1());
 case 3: throw(new MyExc3());
 }
 }
 catch(MyExc1 t) {
 throw(new MyExc3());
 }
 catch(MyExc3 d) {
 System.out.print(5);
 throw(new MyExc2());
 }
 catch(Exception h) {
 System.out.print(6);
 throw(new MyExc2());
 }
 }
}
```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 14666666... (ciclo infinito)

E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
 String tipo="Sempre Verde";
 public int f(int n)
 {
 System.out.println("Pioppo");
 int num=n+1;
 return num;
 }
 public static void main(String args[]) {
 Pioppo p= new Pioppo();
 System.out.println(p.f(5));
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo  
6
- D. Pioppo  
1
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
abstract class Programmatore {
 String nome;
 String cognome;
 int anniEsperienza;

 public Programmatore(String n, String c,
 int a){
 nome=n;
 cognome=c;
 anniEsperienza=a;
 }
 abstract void programma();
}
public class ProgrammatoreJava
 extends Programmatore {
```

```

void stampa() {

 System.out.print("Programmatore Java con ");
 System.out.print (anniEsperienza);
 System.out.println(" anni di esperienza");
}

ProgrammatoreJava(String n, String c, int a){
super(n,c,a);
}
public static void main(String [] argv) {
 ProgrammatoreJava pj=new
 ProgrammatoreJava ("Franco", "Bianchi",5);
 pj.stampa();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

---

4. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
 static int dimensioni = 10;
 void print();
}

class Cono implements FiguraGeometrica {

 static final int numLati = 3;

 public void print() {
 System.out.println("Cono: "
 + dimensioni + " - " +
 numLati + " lati per faccia.");
 }

 public static void main(String [] argv) {
 Cono c1 = new Cono();
 FiguraGeometrica f1;
 c1= (FiguraGeometrica) f1;
 c1.print();
 }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

---

5. Qual è l'output di questo codice?

```

abstract class Programmatore {
 String nome;
 String cognome;
 int anniEsperienza;

 public Programmatore(String n, String c,
 int a){
 nome=n;
 cognome=c;
 anniEsperienza=a;
 }
 abstract void programma();
}

public class ProgrammatoreJava
 extends Programmatore {
 void stampa(){

 System.out.print("Programmatore Java con ");
 System.out.print (anniEsperienza);
 System.out.println(" anni di esperienza");
 }
 void programma(){}
 ProgrammatoreJava(String n, String c, int a){
super(n,c,a);
 }
 public static void main(String [] argv) {
 ProgrammatoreJava pj=new
 ProgrammatoreJava ("Franco", "Bianchi",5);
 pj.stampa();
 }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

---

6. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
 static int dimensioni = 10;
 void print();
}

class Cono implements FiguraGeometrica {

 static final int numLati = 3;

 public void print() {
 System.out.println("Cono: "
 + dimensioni + " - " +
 numLati + " lati per faccia.");
 }
}

```

```

public static void main(String [] argv) {
 Cono c1 = new Cono();
 FiguraGeometrica f1 =c1;
 c1.print();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

7. Date le dichiarazioni:

```

Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

8. Qual è l'output di questo codice?

```

class Stella {
 String nome="Stella";
 final String colore="rosa";

 public Stella(String nome) {
 this.nome = nome;
 }
}

public class Sole extends Stella {

 public Sole(String c) {
 super("Sole");
 this.colore = c;
 }

 public static void main(String [] argv) {
 Sole s = new Sole("bianco");
 System.out.println("Stella: " + s.nome);
 System.out.println("Colore: "
 + s.colore);
 }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole  
Colore: rosa
- D. Stella: Sole  
Colore: bianco
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```

class Stella {
 String nome="Stella";
 String colore="rosa";

 public Stella(String nome) {
 this.nome = nome;
 }
}

public class Sole extends Stella {

 public Sole(String c) {
 super("Sole");
 this.colore = c;
 }

 public static void main(String [] argv) {
 Sole s = new Sole("bianco");
 System.out.println("Stella: " + s.nome);
 System.out.println("Colore: "
 + s.colore);
 }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole  
Colore: rosa
- D. Stella: Sole  
Colore: bianco
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
 String tipo="Sempre Verde";
 public int f()
 {
 System.out.println("Pioppo");
 int num=6;
 return num;
 }
 public static void main(String args[]) {
 Pioppo p= new Pioppo();
 System.out.println(p.f());
 }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. Pioppo  
6
  - D. Pioppo  
1
  - E. Nessuna delle precedenti
- 

11. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `d = c;`
  - B. `d = (Object [] []) e;`
  - C. `c = (String []) d;`
  - D. `d = (Object [] []) c;`
  - E. `d = e;`
- 

12. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - B. Un attributo `static` non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
  - C. Un attributo non può essere contemporaneamente `static` e `final`
  - D. L'operatore `new` non restituisce un riferimento all'oggetto appena creato
  - E. Un oggetto può non esistere dopo la sua dichiarazione
- 

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
 public static void main(String [] argv)
 throws Exception {
 try {
 q();
 }
 finally {
 throw(new MyExc2());
 }
 }
}
```

```
static void q() throws Exception {
 try {
 }
 catch(MyExc2 y) {
 System.out.print(1);
 throw(new MyExc3());
 }
 catch(MyExc3 g) {
 System.out.print(2);
 }
 finally {
 throw(new MyExc3());
 }
}
}
```

- A. Errore a tempo di compilazione
  - B. 1111111... (ciclo infinito)
  - C. Exception in thread main MyExc2
  - D. 1111
  - E. Nessuna delle precedenti
- 

14. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
 public static void main(String [] argv) {
 try {
 q();
 }
 catch(MyExc3 i) {
 System.out.print(1);
 }
 catch(MyExc2 w) {
 }
 catch(MyExc1 j) {
 throw(new MyExc3());
 }
 finally {
 System.out.print(2);
 throw(new MyExc2());
 }
 }
 static void q() {
 try {
 System.out.print(3);
 throw(new Exception());
 }
 finally {
 throw(new MyExc3());
 }
 }
}
```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2

15. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
 static String marca="Ferrari";
 double numlitri=1.5;

 Bottiglia() {
 System.out.println(marca);
 }

 Bottiglia(double litri) {
 this.numlitri=litri;
 System.out.println(litri);
 }

 public static void main(String args[]) {
 Bottiglia spumante=new Bottiglia();
 Bottiglia vinoRosso=new Bottiglia();
 Bottiglia vinoBianco=new Bottiglia(1.0);
 spumante=null;
 vinoBianco=spumante;
 /*QUI*/
 System.out.println("END");
 }
}
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

16. Quale output si ottiene invocando il metodo m?

```
class F {
 private String [] a1 = new String [7];
 private String [] a4 = new String [2];
 private Boolean b3 = new Boolean(true);
 void m() {
 q(a1, new String [2], b3, b3);
 }
 void q(Object a2, String [] a3,
 Boolean b1, Object b2) {
 if(a2 == a1) {
 System.out.print(1);
 } else {
 System.out.print(0);
 }
 if(a3 == a4) {
 System.out.print(1);
 } else {
 System.out.print(0);
 }
 if(b2 == b1) {
 System.out.print(1);
 } else {
 System.out.print(0);
 }
 }
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 66**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
 public static void main(String [] argv) {
 try {
 q();
 }
 catch(MyExc3 i) {
 System.out.print(1);
 }
 catch(MyExc2 w) {
 }
 catch(MyExc1 j) {
 throw(new MyExc3());
 }
 finally {
 System.out.print(2);
 throw(new MyExc2());
 }
 }
 static void q() {
 try {
 System.out.print(3);
 throw(new Exception());
 }
 finally {
 throw(new MyExc3());
 }
 }
}
```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
class Stella {
 String nome="Stella";
 String colore="rosa";
```

```
public Stella(String nome) {
 this.nome = nome;
}

public class Sole extends Stella {

 public Sole(String c) {
 super("Sole");
 this.colore = c;
 }

 public static void main(String [] argv) {
 Sole s = new Sole("bianco");
 System.out.println("Stella: " + s.nome);
 System.out.println("Colore: "
 + s.colore);
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole  
Colore: rosa
- D. Stella: Sole  
Colore: bianco
- E. Nessuna delle precedenti

3. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti



4. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
 static int dimensioni = 10;
 void print();
}

class Cono implements FiguraGeometrica {

 static final int numLati = 3;

 public void print() {
 System.out.println("Cono: "
 + dimensioni + " - " +
 numLati + " lati per faccia.");
 }

 public static void main(String [] argv) {
 Cono c1 = new Cono();
 FiguraGeometrica f1 =c1;
 c1.print();
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

5. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
 static String marca="Ferrari";
 double numlitri=1.5;

 Bottiglia(){
 System.out.println(marca);
 }

 Bottiglia(double litri){
 this.numlitri=litri;
 System.out.println(litri);
 }

 public static void main(String args[]) {
 Bottiglia spumante=new Bottiglia();
 Bottiglia vinoRosso=new Bottiglia();
 Bottiglia vinoBianco=new Bottiglia(1.0);
 spumante=null;
 vinoBianco=spumante;
 /*QUI*/
 System.out.println("END");
 }
}
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

6. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

7. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
 static int dimensioni = 10;
 void print();
}

class Cono implements FiguraGeometrica {

 static final int numLati = 3;

 public void print() {
 System.out.println("Cono: "
 + dimensioni + " - " +
 numLati + " lati per faccia.");
 }

 public static void main(String [] argv) {
 Cono c1 = new Cono();
 FiguraGeometrica f1;
 c1= (FiguraGeometrica) f1;
 c1.print();
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```
class Stella {
 String nome="Stella";
 final String colore="rosa";

 public Stella(String nome) {
 this.nome = nome;
 }
}

public class Sole extends Stella {

 public Sole(String c) {
 super("Sole");
 this.colore = c;
 }

 public static void main(String [] argv) {
 Sole s = new Sole("bianco");
 System.out.println("Stella: " + s.nome);
 System.out.println("Colore: "
 + s.colore);
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole  
Colore: rosa
- D. Stella: Sole  
Colore: bianco
- E. Nessuna delle precedenti

9. Quale output si ottiene invocando il metodo m?

```
class F {
 private String [] a1 = new String [7];
 private String [] a4 = new String [2];
 private Boolean b3 = new Boolean(true);
 void m() {
 q(a1, new String [2], b3, b3);
 }
 void q(Object a2, String [] a3,
 Boolean b1, Object b2) {
 if(a2 == a1) {
 System.out.print(1);
 } else {
 System.out.print(0);
 }
 if(a3 == a4) {
 System.out.print(1);
 } else {
 System.out.print(0);
 }
 if(b2 == b1) {
 System.out.print(1);
 } else {
```

```
 System.out.print(0);
 }
 }
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

10. Qual è l'output di questo codice?

```
abstract class Programmatore {
 String nome;
 String cognome;
 int anniEsperienza;

 public Programmatore(String n, String c,
 int a){
 nome=n;
 cognome=c;
 anniEsperienza=a;
 }
 abstract void programma();
}

public class ProgrammatoreJava
 extends Programmatore {
 void stampa(){

 System.out.print("Programmatore Java con ");
 System.out.print (anniEsperienza);
 System.out.println(" anni di esperienza");
 }
 void programma(){}
 ProgrammatoreJava(String n, String c, int a){
 super(n,c,a);
 }
 public static void main(String [] argv) {
 ProgrammatoreJava pj=new
 ProgrammatoreJava("Franco", "Bianchi",5);
 pj.stampa();
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```

abstract class Programmatore {
 String nome;
 String cognome;
 int anniEsperienza;

 public Programmatore(String n, String c,
 int a){
 nome=n;
 cognome=c;
 anniEsperienza=a;
 }
 abstract void programma();
}
public class ProgrammatoreJava
 extends Programmatore {

 void stampa(){

 System.out.print("Programmatore Java con ");
 System.out.print(anniEsperienza);
 System.out.println(" anni di esperienza");
 }

 ProgrammatoreJava(String n, String c, int a){
 super(n, c, a);
 }
 public static void main(String [] argv) {
 ProgrammatoreJava pj=new
 ProgrammatoreJava("Franco", "Bianchi",5);
 pj.stampa();
 }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

---

12. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
 String tipo="Sempre Verde";
 public int f(int n)
 {
 System.out.println("Pioppo");
 int num=n+1;
 return num;
 }
 public static void main(String args[]) {
 Pioppo p= new Pioppo();
 System.out.println(p.f(5));
 }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo  
6
- D. Pioppo  
1
- E. Nessuna delle precedenti

---

13. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
 String tipo="Sempre Verde";
 public int f()
 {
 System.out.println("Pioppo");
 int num=6;
 return num;
 }
 public static void main(String args[]) {
 Pioppo p= new Pioppo();
 System.out.println(p.f());
 }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo  
6
- D. Pioppo  
1
- E. Nessuna delle precedenti

---

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
 public static void main(String [] argv)
 throws Exception {
 try {
 System.out.print(1);
 n();
 }
 catch(MyExc3 s) {
 System.out.print(2);
 throw(new Exception());
 }
 finally {
 System.out.print(3);
 throw(new MyExc3());
 }
 }
 static void n() throws Exception {
 try {
 System.out.print(4);

```

```

 switch(1) {
 case 1: throw(new MyExc2());
 case 2: throw(new MyExc1());
 case 3: throw(new MyExc3());
 }
 }
 catch(MyExc1 t) {
 throw(new MyExc3());
 }
 catch(MyExc3 d) {
 System.out.print(5);
 throw(new MyExc2());
 }
 catch(Exception h) {
 System.out.print(6);
 throw(new MyExc2());
 }
}
}

```

15. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
 public static void main(String [] argv)
 throws Exception {
 try {
 q();
 }
 finally {
 throw(new MyExc2());
 }
 }
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

```

}
static void q() throws Exception {
 try {
 }
 catch(MyExc2 y) {
 System.out.print(1);
 throw(new MyExc3());
 }
 catch(MyExc3 g) {
 System.out.print(2);
 }
 finally {
 throw(new MyExc3());
 }
}
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

16. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo **static** non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente **static** e **final**
- D. L'operatore **new** non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 67**

1. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `d = c;`
- B. `d = (Object [] []) e;`
- C. `c = (String []) d;`
- D. `d = (Object [] []) c;`
- E. `d = e;`

2. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
 static int dimensioni = 10;
 void print();
}
```

```
class Cono implements FiguraGeometrica {

 static final int numLati = 3;

 public void print() {
 System.out.println("Cono: "
 + dimensioni + " - " +
 numLati + " lati per faccia.");
 }

 public static void main(String [] argv) {
 Cono c1 = new Cono();
 FiguraGeometrica f1;
 c1 = (FiguraGeometrica) f1;
 c1.print();
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia

- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

3. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo `static` non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente `static` e `final`
- D. L'operatore `new` non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

4. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `q = (Error []) m;`
- B. `q = (Error []) w;`
- C. `m = (Object [] []) w;`
- D. `m = (Object [] []) q;`
- E. Nessuno dei precedenti

5. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
 static String marca="Ferrari";
 double numlitri=1.5;

 Bottiglia() {
 System.out.println(marca);
 }

 Bottiglia(double litri) {
 this.numlitri=litri;
 System.out.println(litri);
 }

 public static void main(String args[]) {
 Bottiglia spumante=new Bottiglia();
 Bottiglia vinoRosso=new Bottiglia();
 Bottiglia vinoBianco=new Bottiglia(1.0);
 spumante=null;
 vinoBianco=spumante;
 /*QUI*/
 System.out.println("END");
 }
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
 public static void main(String [] argv) {
 try {
 q();
 }
 catch(MyExc3 i) {
 System.out.print(1);
 }
 catch(MyExc2 w) {
 }
 catch(MyExc1 j) {
 throw(new MyExc3());
 }
 finally {
 System.out.print(2);
 throw(new MyExc2());
 }
 }
 static void q() {
 try {
 System.out.print(3);
 throw(new Exception());
 }
 }
}

```

```

}
finally {
 throw(new MyExc3());
}
}
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

7. Qual è l'output di questo codice?

```

class Stella {
 String nome="Stella";
 final String colore="rosa";

 public Stella(String nome) {
 this.nome = nome;
 }
}

public class Sole extends Stella {

 public Sole(String c) {
 super("Sole");
 this.colore = c;
 }

 public static void main(String [] argv) {
 Sole s = new Sole("bianco");
 System.out.println("Stella: " + s.nome);
 System.out.println("Colore: "
 + s.colore);
 }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole  
Colore: rosa
- D. Stella: Sole  
Colore: bianco
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
 public static void main(String [] argv)
 throws Exception {
 try {
 q();
 }
 }
}

```

```

 }
 finally {
 throw(new MyExc2());
 }
}
static void q() throws Exception {
 try {
 }
 catch(MyExc2 y) {
 System.out.print(1);
 throw(new MyExc3());
 }
 catch(MyExc3 g) {
 System.out.print(2);
 }
 finally {
 throw(new MyExc3());
 }
}
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```

abstract class Programmatore {
 String nome;
 String cognome;
 int anniEsperienza;

 public Programmatore(String n, String c,
 int a){
 nome=n;
 cognome=c;
 anniEsperienza=a;
 }
 abstract void programma();
}
public class ProgrammatoreJava
 extends Programmatore {

 void stampa(){

 System.out.print("Programmatore Java con ");
 System.out.print(anniEsperienza);
 System.out.println(" anni di esperienza");
 }

 ProgrammatoreJava(String n, String c, int a){
super(n,c,a);
 }
 public static void main(String [] argv) {
 ProgrammatoreJava pj=new
 ProgrammatoreJava("Franco", "Bianchi",5);

```

```

 pj.stampa();
 }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
 public static void main(String [] argv)
 throws Exception {
 try {
 System.out.print(1);
 n();
 }
 catch(MyExc3 s) {
 System.out.print(2);
 throw(new Exception());
 }
 finally {
 System.out.print(3);
 throw(new MyExc3());
 }
 }
 static void n() throws Exception {
 try {
 System.out.print(4);
 switch(1) {
 case 1: throw(new MyExc2());
 case 2: throw(new MyExc1());
 case 3: throw(new MyExc3());
 }
 }
 catch(MyExc1 t) {
 throw(new MyExc3());
 }
 catch(MyExc3 d) {
 System.out.print(5);
 throw(new MyExc2());
 }
 catch(Exception h) {
 System.out.print(6);
 throw(new MyExc2());
 }
 }
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)



E. Nessuna delle precedenti

---

11. Qual è l'output di questo codice?

```
abstract class Programmatore {
 String nome;
 String cognome;
 int anniEsperienza;

 public Programmatore(String n, String c,
 int a){
 nome=n;
 cognome=c;
 anniEsperienza=a;
 }
 abstract void programma();
}
public class ProgrammatoreJava
 extends Programmatore {
 void stampa(){

 System.out.print("Programmatore Java con ");
 System.out.print(anniEsperienza);
 System.out.println(" anni di esperienza");
 }
 void programma(){}
 ProgrammatoreJava(String n, String c, int a){
 super(n,c,a);
 }
 public static void main(String [] argv) {
 ProgrammatoreJava pj=new
 ProgrammatoreJava("Franco", "Bianchi",5);
 pj.stampa();
 }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. Programmatore Java con 5 anni di esperienza
  - D. Programmatore Java con 0 anni di esperienza
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
 String tipo="Sempre Verde";
 public int f(int n)
 {
 System.out.println("Pioppo");
 int num=n+1;
 return num;
 }
 public static void main(String args[]) {
 Pioppo p= new Pioppo();
 System.out.println(p.f(5));
 }
}
```

```
}
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. Pioppo  
6
  - D. Pioppo  
1
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
 static int dimensioni = 10;
 void print();
}

class Cono implements FiguraGeometrica {
 static final int numLati = 3;

 public void print() {
 System.out.println("Cono: "
 + dimensioni + " - " +
 numLati + " lati per faccia.");
 }

 public static void main(String [] argv) {
 Cono c1 = new Cono();
 FiguraGeometrica f1 =c1;
 c1.print();
 }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. Cono: 8 - 3 lati per faccia
  - D. Cono: 3 - 3 lati per faccia
  - E. Nessuna delle precedenti
- 

14. Quale output si ottiene invocando il metodo m?

```
class F {
 private String [] a1 = new String [7];
 private String [] a4 = new String [2];
 private Boolean b3 = new Boolean(true);
 void m() {
 q(a1, new String [2], b3, b3);
 }
 void q(Object a2, String [] a3,
 Boolean b1, Object b2) {
 if(a2 == a1) {
 System.out.print(1);
 } else {
 System.out.print(0);
 }
 }
}
```

```

 }
 if(a3 == a4) {
 System.out.print(1);
 } else {
 System.out.print(0);
 }
 if(b2 == b1) {
 System.out.print(1);
 } else {
 System.out.print(0);
 }
}
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

15. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
 String tipo="Sempre Verde";
 public int f()
 {
 System.out.println("Pioppo");
 int num=6;
 return num;
 }
 public static void main(String args[]) {
 Pioppo p= new Pioppo();
 System.out.println(p.f());
 }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. Pioppo
- 6

D. Pioppo  
1

E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```

class Stella {
 String nome="Stella";
 String colore="rosa";

 public Stella(String nome) {
 this.nome = nome;
 }
}

public class Sole extends Stella {

 public Sole(String c) {
 super("Sole");
 this.colore = c;
 }

 public static void main(String [] argv) {
 Sole s = new Sole("bianco");
 System.out.println("Stella: " + s.nome);
 System.out.println("Colore: "
 + s.colore);
 }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole  
Colore: rosa
- D. Stella: Sole  
Colore: bianco
- E. Nessuna delle precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 68**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
 public static void main(String [] argv) {
 try {
 q();
 }
 catch(MyExc3 i) {
 System.out.print(1);
 }
 catch(MyExc2 w) {
 }
 catch(MyExc1 j) {
 throw(new MyExc3());
 }
 finally {
 System.out.print(2);
 throw(new MyExc2());
 }
 }
 static void q() {
 try {
 System.out.print(3);
 throw(new Exception());
 }
 finally {
 throw(new MyExc3());
 }
 }
}
```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

2. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

3. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
 static int dimensioni = 10;
 void print();
}

class Cono implements FiguraGeometrica {

 static final int numLati = 3;

 public void print() {
 System.out.println("Cono: "
 + dimensioni + " - " +
 numLati + " lati per faccia.");
 }

 public static void main(String [] argv) {
 Cono c1 = new Cono();
 FiguraGeometrica f1;
 c1= (FiguraGeometrica) f1;
 c1.print();
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

4. Date le dichiarazioni:

```

Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

5. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
 String tipo="Sempre Verde";
 public int f(int n)
 {
System.out.println("Pioppo");
int num=n+1;
return num;
 }
 public static void main(String args[]) {
 Pioppo p= new Pioppo();
 System.out.println(p.f(5));
 }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo  
6
- D. Pioppo  
1
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```

abstract class Programmatore {
 String nome;
 String cognome;
 int anniEsperienza;

 public Programmatore(String n, String c,
 int a){
 nome=n;
 cognome=c;
 anniEsperienza=a;
 }
}

```

```

 abstract void programma();
 }
 public class ProgrammatoreJava
 extends Programmatore {
 void stampa(){

 System.out.print("Programmatore Java con ");
 System.out.print (anniEsperienza);
 System.out.println(" anni di esperienza");
 }
 void programma(){}
 ProgrammatoreJava(String n, String c, int a){
 super(n,c,a);
 }
 public static void main(String [] argv) {
 ProgrammatoreJava pj=new
 ProgrammatoreJava("Franco", "Bianchi",5);
 pj.stampa();
 }
 }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

7. Qual è l'output di questo codice?

```

abstract class Programmatore {
 String nome;
 String cognome;
 int anniEsperienza;

 public Programmatore(String n, String c,
 int a){
 nome=n;
 cognome=c;
 anniEsperienza=a;
 }
 abstract void programma();
}
public class ProgrammatoreJava
 extends Programmatore {

 void stampa(){

 System.out.print("Programmatore Java con ");
 System.out.print (anniEsperienza);
 System.out.println(" anni di esperienza");
 }

 ProgrammatoreJava(String n, String c, int a){
 super(n,c,a);
 }
 public static void main(String [] argv) {
 ProgrammatoreJava pj=new
 ProgrammatoreJava("Franco", "Bianchi",5);
 }
}

```

```

 pj.stampa();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```

class Stella {
 String nome="Stella";
 String colore="rosa";

 public Stella(String nome) {
 this.nome = nome;
 }
}

public class Sole extends Stella {

 public Sole(String c) {
 super("Sole");
 this.colore = c;
 }

 public static void main(String [] argv) {
 Sole s = new Sole("bianco");
 System.out.println("Stella: " + s.nome);
 System.out.println("Colore: "
 + s.colore);
 }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole  
Colore: rosa
- D. Stella: Sole  
Colore: bianco
- E. Nessuna delle precedenti

9. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final

- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

10. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
 static String marca="Ferrari";
 double numlitri=1.5;

 Bottiglia() {
 System.out.println(marca);
 }

 Bottiglia(double litri) {
 this.numlitri=litri;
 System.out.println(litri);
 }

 public static void main(String args[]) {
 Bottiglia spumante=new Bottiglia();
 Bottiglia vinoRosso=new Bottiglia();
 Bottiglia vinoBianco=new Bottiglia(1.0);
 spumante=null;
 vinoBianco=spumante;
 /*QUI*/
 System.out.println("END");
 }
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
 public static void main(String [] argv)
 throws Exception {
 try {
 System.out.print(1);
 n();
 }
 catch(MyExc3 s) {
 System.out.print(2);
 throw(new Exception());
 }
 finally {
 System.out.print(3);
 throw(new MyExc3());
 }
 }
}

```

```

 }
}
static void n() throws Exception {
 try {
 System.out.print(4);
 switch(1) {
 case 1: throw(new MyExc2());
 case 2: throw(new MyExc1());
 case 3: throw(new MyExc3());
 }
 }
 catch(MyExc1 t) {
 throw(new MyExc3());
 }
 catch(MyExc3 d) {
 System.out.print(5);
 throw(new MyExc2());
 }
 catch(Exception h) {
 System.out.print(6);
 throw(new MyExc2());
 }
}
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

12. Quale output si ottiene invocando il metodo m?

```

class F {
 private String [] a1 = new String [7];
 private String [] a4 = new String [2];
 private Boolean b3 = new Boolean(true);
 void m() {
 q(a1, new String [2], b3, b3);
 }
 void q(Object a2, String [] a3,
 Boolean b1, Object b2) {
 if(a2 == a1) {
 System.out.print(1);
 } else {
 System.out.print(0);
 }
 if(a3 == a4) {
 System.out.print(1);
 } else {
 System.out.print(0);
 }
 if(b2 == b1) {
 System.out.print(1);
 } else {
 System.out.print(0);
 }
 }
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
 public static void main(String [] argv)
 throws Exception {
 try {
 q();
 }
 finally {
 throw(new MyExc2());
 }
 }
 static void q() throws Exception {
 try {
 }
 catch(MyExc2 y) {
 System.out.print(1);
 throw(new MyExc3());
 }
 catch(MyExc3 g) {
 System.out.print(2);
 }
 finally {
 throw(new MyExc3());
 }
 }
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```

class Stella {
 String nome="Stella";
 final String colore="rosa";

 public Stella(String nome) {
 this.nome = nome;
 }
}

public class Sole extends Stella {

```

```

public Sole(String c) {
 super("Sole");
 this.colore = c;
}

public static void main(String [] argv) {
 Sole s = new Sole("bianco");
 System.out.println("Stella: " + s.nome);
 System.out.println("Colore: "
 + s.colore);
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. Stella: Sole  
Colore: rosa
  - D. Stella: Sole  
Colore: bianco
  - E. Nessuna delle precedenti
- 

15. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
 static int dimensioni = 10;
 void print();
}

class Cono implements FiguraGeometrica {

 static final int numLati = 3;

 public void print() {
 System.out.println("Cono: "
 + dimensioni + " - " +
 numLati + " lati per faccia.");
 }
}

```

```

public static void main(String [] argv) {
 Cono c1 = new Cono();
 FiguraGeometrica f1 =c1;
 c1.print();
}
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. Cono: 8 - 3 lati per faccia
  - D. Cono: 3 - 3 lati per faccia
  - E. Nessuna delle precedenti
- 

16. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
 String tipo="Sempre Verde";
 public int f()
 {
 System.out.println("Pioppo");
 int num=6;
 return num;
 }
 public static void main(String args[]) {
 Pioppo p= new Pioppo();
 System.out.println(p.f());
 }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. Pioppo  
6
  - D. Pioppo  
1
  - E. Nessuna delle precedenti
-





Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 69**

1. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
 String tipo="Sempre Verde";
 public int f(int n)
 {
System.out.println("Pioppo");
int num=n+1;
return num;
 }
 public static void main(String args[]) {
 Pioppo p= new Pioppo();
 System.out.println(p.f(5));
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo  
6
- D. Pioppo  
1
- E. Nessuna delle precedenti

2. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
 public static void main(String [] argv)
 throws Exception {
 try {
 System.out.print(1);
 n();
 }
 catch(MyExc3 s) {
 System.out.print(2);
 throw(new Exception());
 }
 finally {
 System.out.print(3);
 throw(new MyExc3());
 }
 }
 static void n() throws Exception {
 try {
 System.out.print(4);
 switch(1) {
 case 1: throw(new MyExc2());
 case 2: throw(new MyExc1());
 case 3: throw(new MyExc3());
 }
 }
 catch(MyExc1 t) {
 throw(new MyExc3());
 }
 catch(MyExc3 d) {
 System.out.print(5);
 throw(new MyExc2());
 }
 catch(Exception h) {
 System.out.print(6);
 throw(new MyExc2());
 }
 }
}
```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

---

4. Qual è l'output di questo codice?

```
abstract class Programmatore {
 String nome;
 String cognome;
 int anniEsperienza;

 public Programmatore(String n, String c,
 int a){
 nome=n;
 cognome=c;
 anniEsperienza=a;
 }
 abstract void programma();
}
public class ProgrammatoreJava
 extends Programmatore {

 void stampa(){

 System.out.print("Programmatore Java con ")
 System.out.print (anniEsperienza);
 System.out.println(" anni di esperienza");
 }

 ProgrammatoreJava(String n, String c, int a){
 super(n,c,a);
 }
 public static void main(String [] argv) {
 ProgrammatoreJava pj=new
 ProgrammatoreJava("Franco", "Bianchi",5);
 pj.stampa();
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

---

5. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
 static String marca="Ferrari";
 double numlitri=1.5;

 Bottiglia(){
 System.out.println(marca);
 }

 Bottiglia(double litri){
 this.numlitri=litri;
 System.out.println(litri);
 }
}
```

```
}
public static void main(String args[]) {
 Bottiglia spumante=new Bottiglia();
 Bottiglia vinoRosso=new Bottiglia();
 Bottiglia vinoBianco=new Bottiglia(1.0);
 spumante=null;
 vinoBianco=spumante;
 /*QUI*/
 System.out.println("END");
}
}
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

---

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
 public static void main(String [] argv) {
 try {
 q();
 }
 catch(MyExc3 i) {
 System.out.print(1);
 }
 catch(MyExc2 w) {
 }
 catch(MyExc1 j) {
 throw(new MyExc3());
 }
 finally {
 System.out.print(2);
 throw(new MyExc2());
 }
 }
 static void q() {
 try {
 System.out.print(3);
 throw(new Exception());
 }
 finally {
 throw(new MyExc3());
 }
 }
}
```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

---

7. Qual è l'output di questo codice?

```
class Stella {
 String nome="Stella";
 final String colore="rosa";

 public Stella(String nome) {
 this.nome = nome;
 }
}

public class Sole extends Stella {

 public Sole(String c) {
 super("Sole");
 this.colore = c;
 }

 public static void main(String [] argv) {
 Sole s = new Sole("bianco");
 System.out.println("Stella: " + s.nome);
 System.out.println("Colore: "
 + s.colore);
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole  
Colore: rosa
- D. Stella: Sole  
Colore: bianco
- E. Nessuna delle precedenti

---

8. Qual è l'output di questo codice?

```
class Stella {
 String nome="Stella";
 String colore="rosa";

 public Stella(String nome) {
 this.nome = nome;
 }
}

public class Sole extends Stella {

 public Sole(String c) {
 super("Sole");
 this.colore = c;
 }

 public static void main(String [] argv) {
 Sole s = new Sole("bianco");
 System.out.println("Stella: " + s.nome);
 System.out.println("Colore: "
```

```
+ s.colore);
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole  
Colore: rosa
- D. Stella: Sole  
Colore: bianco
- E. Nessuna delle precedenti

---

9. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
 String tipo="Sempre Verde";
 public int f()
 {
 System.out.println("Pioppo");
 int num=6;
 return num;
 }
 public static void main(String args[]) {
 Pioppo p= new Pioppo();
 System.out.println(p.f());
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo  
6
- D. Pioppo  
1
- E. Nessuna delle precedenti

---

10. Quale output si ottiene invocando il metodo m?

```
class F {
 private String [] a1 = new String [7];
 private String [] a4 = new String [2];
 private Boolean b3 = new Boolean(true);
 void m() {
 q(a1, new String [2], b3, b3);
 }
 void q(Object a2, String [] a3,
 Boolean b1, Object b2) {
 if(a2 == a1) {
 System.out.print(1);
 } else {
 System.out.print(0);
 }
 if(a3 == a4) {
 System.out.print(1);
 } else {
 System.out.print(0);
 }
 }
}
```

```

 if(b2 == b1) {
 System.out.print(1);
 } else {
 System.out.print(0);
 }
}
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

11. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

12. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
 static int dimensioni = 10;
 void print();
}

class Cono implements FiguraGeometrica {

 static final int numLati = 3;

 public void print() {
 System.out.println("Cono: "
 + dimensioni + " - " +
 numLati + " lati per faccia.");
 }

 public static void main(String [] argv) {
 Cono c1 = new Cono();
 FiguraGeometrica f1 =c1;
 c1.print();
 }
}

```

- A. Errore a tempo di compilazione

- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
 static int dimensioni = 10;
 void print();
}

class Cono implements FiguraGeometrica {

 static final int numLati = 3;

 public void print() {
 System.out.println("Cono: "
 + dimensioni + " - " +
 numLati + " lati per faccia.");
 }

 public static void main(String [] argv) {
 Cono c1 = new Cono();
 FiguraGeometrica f1;
 c1= (FiguraGeometrica) f1;
 c1.print();
 }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```

abstract class Programmatore {
 String nome;
 String cognome;
 int anniEsperienza;

 public Programmatore(String n, String c,
 int a){
 nome=n;
 cognome=c;
 anniEsperienza=a;
 }
 abstract void programma();
}

public class ProgrammatoreJava
 extends Programmatore {
 void stampa(){
 System.out.print("Programmatore Java con ");
 }
}

```

```

 System.out.print (anniEsperienza);
 System.out.println(" anni di esperienza");
}
void programma() {}
ProgrammatoreJava (String n, String c, int a) {
super (n, c, a);
}
public static void main (String [] argv) {
 ProgrammatoreJava pj=new
 ProgrammatoreJava ("Franco", "Bianchi",5);
 pj.stampa ();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

15. Date le dichiarazioni:

```

Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
 public static void main (String [] argv)
 throws Exception {
 try {
 q ();
 }
 finally {
 throw (new MyExc2 ());
 }
 }
 static void q () throws Exception {
 try {
 }
 catch (MyExc2 y) {
 System.out.print (1);
 throw (new MyExc3 ());
 }
 catch (MyExc3 g) {
 System.out.print (2);
 }
 finally {
 throw (new MyExc3 ());
 }
 }
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 70**

1. Qual è l'output di questo codice?

```
abstract class Programmatore {
 String nome;
 String cognome;
 int anniEsperienza;

 public Programmatore(String n, String c,
 int a){
 nome=n;
 cognome=c;
 anniEsperienza=a;
 }
 abstract void programma();
}
public class ProgrammatoreJava
 extends Programmatore {

 void stampa(){

 System.out.print("Programmatore Java con ");
 System.out.print (anniEsperienza);
 System.out.println(" anni di esperienza");
 }

 ProgrammatoreJava(String n, String c, int a){
 super(n,c,a);
 }
 public static void main(String [] argv) {
 ProgrammatoreJava pj=new
 ProgrammatoreJava("Franco", "Bianchi",5);
 pj.stampa();
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
```

```
String tipo="Sempre Verde";
public int f()
{
 System.out.println("Pioppo");
 int num=6;
 return num;
}
public static void main(String args[]) {
 Pioppo p= new Pioppo();
 System.out.println(p.f());
}
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo  
6
- D. Pioppo  
1
- E. Nessuna delle precedenti

3. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

4. Qual è l'output di questo codice?

```
class MyExcl extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
 public static void main(String [] argv)
```



```

 throws Exception {
 try {
 q();
 }
 finally {
 throw(new MyExc2());
 }
}
static void q() throws Exception {
 try {
 }
 catch(MyExc2 y) {
 System.out.print(1);
 throw(new MyExc3());
 }
 catch(MyExc3 g) {
 System.out.print(2);
 }
 finally {
 throw(new MyExc3());
 }
}
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class Bl {
 public static void main(String [] argv) {
 try {
 q();
 }
 catch(MyExc3 i) {
 System.out.print(1);
 }
 catch(MyExc2 w) {
 }
 catch(MyExc1 j) {
 throw(new MyExc3());
 }
 finally {
 System.out.print(2);
 throw(new MyExc2());
 }
 }
 static void q() {
 try {
 System.out.print(3);
 throw(new Exception());
 }
 }
}

```

```

 finally {
 throw(new MyExc3());
 }
 }
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

6. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

7. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
 static String marca="Ferrari";
 double numlitri=1.5;

 Bottiglia(){
 System.out.println(marca);
 }

 Bottiglia(double litri){
 this.numlitri=litri;
 System.out.println(litri);
 }

 public static void main(String args[]) {
 Bottiglia spumante=new Bottiglia();
 Bottiglia vinoRosso=new Bottiglia();
 Bottiglia vinoBianco=new Bottiglia(1.0);
 spumante=null;
 vinoBianco=spumante;
 /*QUI*/
 System.out.println("END");
 }
}

```

- A. 0
- B. 1

- C. 2
- D. 4
- E. 5

8. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
 String tipo="Sempre Verde";
 public int f(int n)
 {
System.out.println("Pioppo");
int num=n+1;
return num;
 }
 public static void main(String args[]) {
 Pioppo p= new Pioppo();
 System.out.println(p.f(5));
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo  
6
- D. Pioppo  
1
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```
abstract class Programmatore {
 String nome;
 String cognome;
 int anniEsperienza;

 public Programmatore(String n, String c,
 int a){
 nome=n;
 cognome=c;
 anniEsperienza=a;
 }
 abstract void programma();
}
public class ProgrammatoreJava
 extends Programmatore {
 void stampa(){
 System.out.print("Programmatore Java con ");
 System.out.print(anniEsperienza);
 System.out.println(" anni di esperienza");
 }
 void programma(){}
 ProgrammatoreJava(String n, String c, int a){
 super(n,c,a);
 }
}
```

```
}
public static void main(String [] argv) {
 ProgrammatoreJava pj=new
 ProgrammatoreJava("Franco", "Bianchi",5);
 pj.stampa();
}
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

10. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

11. Qual è l'output di questo codice?

```
class Stella {
 String nome="Stella";
 final String colore="rosa";

 public Stella(String nome) {
 this.nome = nome;
 }
}

public class Sole extends Stella {

 public Sole(String c) {
 super("Sole");
 this.colore = c;
 }

 public static void main(String [] argv) {
 Sole s = new Sole("bianco");
 System.out.println("Stella: " + s.nome);
 System.out.println("Colore: "
 + s.colore);
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione

- C. Stella: Sole  
Colore: rosa
- D. Stella: Sole  
Colore: bianco
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```
class Stella {
 String nome="Stella";
 String colore="rosa";

 public Stella(String nome) {
 this.nome = nome;
 }
}

public class Sole extends Stella {

 public Sole(String c) {
 super("Sole");
 this.colore = c;
 }

 public static void main(String [] argv) {
 Sole s = new Sole("bianco");
 System.out.println("Stella: " + s.nome);
 System.out.println("Colore: "
 + s.colore);
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole  
Colore: rosa
- D. Stella: Sole  
Colore: bianco
- E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
 public static void main(String [] argv)
 throws Exception {
 try {
 System.out.print(1);
 n();
 }
 catch(MyExc3 s) {
 System.out.print(2);
 throw(new Exception());
 }
 }
}
```

```
finally {
 System.out.print(3);
 throw(new MyExc3());
}
}

static void n() throws Exception {
 try {
 System.out.print(4);
 switch(1) {
 case 1: throw(new MyExc2());
 case 2: throw(new MyExc1());
 case 3: throw(new MyExc3());
 }
 }
 catch(MyExc1 t) {
 throw(new MyExc3());
 }
 catch(MyExc3 d) {
 System.out.print(5);
 throw(new MyExc2());
 }
 catch(Exception h) {
 System.out.print(6);
 throw(new MyExc2());
 }
}
}
```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

14. Quale output si ottiene invocando il metodo m?

```
class F {
 private String [] a1 = new String [7];
 private String [] a4 = new String [2];
 private Boolean b3 = new Boolean(true);
 void m() {
 q(a1, new String [2], b3, b3);
 }
 void q(Object a2, String [] a3,
 Boolean b1, Object b2) {
 if(a2 == a1) {
 System.out.print(1);
 } else {
 System.out.print(0);
 }
 if(a3 == a4) {
 System.out.print(1);
 } else {
 System.out.print(0);
 }
 if(b2 == b1) {
 System.out.print(1);
 } else {
 System.out.print(0);
 }
 }
}
```

```

 System.out.print(0);
 }
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

15. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
 static int dimensioni = 10;
 void print();
}

class Cono implements FiguraGeometrica {

 static final int numLati = 3;

 public void print() {
 System.out.println("Cono: "
 + dimensioni + " - " +
 numLati + " lati per faccia.");
 }

 public static void main(String [] argv) {
 Cono c1 = new Cono();
 FiguraGeometrica f1;
 c1= (FiguraGeometrica) f1;
 c1.print();
 }
}

```

- A. Errore a tempo di compilazione

- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
 static int dimensioni = 10;
 void print();
}

class Cono implements FiguraGeometrica {

 static final int numLati = 3;

 public void print() {
 System.out.println("Cono: "
 + dimensioni + " - " +
 numLati + " lati per faccia.");
 }

 public static void main(String [] argv) {
 Cono c1 = new Cono();
 FiguraGeometrica f1 =c1;
 c1.print();
 }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 71**

1. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
 static int dimensioni = 10;
 void print();
}

class Cono implements FiguraGeometrica {

 static final int numLati = 3;

 public void print() {
 System.out.println("Cono: "
 + dimensioni + " - " +
 numLati + " lati per faccia.");
 }

 public static void main(String [] argv) {
 Cono cl = new Cono();
 FiguraGeometrica fl =cl;
 cl.print();
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

2. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

3. Qual è l'output di questo codice?

```
abstract class Programmatore {
 String nome;
 String cognome;
 int anniEsperienza;

 public Programmatore(String n, String c,
 int a){
 nome=n;
 cognome=c;
 anniEsperienza=a;
 }
 abstract void programma();
}

public class ProgrammatoreJava
 extends Programmatore {

 void stampa(){

 System.out.print("Programmatore Java con ");
 System.out.print(anniEsperienza);
 System.out.println(" anni di esperienza");
 }

 ProgrammatoreJava(String n, String c, int a){
 super(n,c,a);
 }

 public static void main(String [] argv) {
 ProgrammatoreJava pj=new
 ProgrammatoreJava("Franco", "Bianchi",5);
 pj.stampa();
 }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

4. Date le dichiarazioni:

```

Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

5. Qual è l'output di questo codice?

```

abstract class Programmatore {
 String nome;
 String cognome;
 int anniEsperienza;

 public Programmatore(String n, String c,
 int a){
 nome=n;
 cognome=c;
 anniEsperienza=a;
 }
 abstract void programma();
}
public class ProgrammatoreJava
 extends Programmatore {
 void stampa(){

 System.out.print("Programmatore Java con ");
 System.out.print (anniEsperienza);
 System.out.println(" anni di esperienza");
 }
 void programma() {}
 ProgrammatoreJava(String n, String c, int a){
super(n, c, a);
 }
 public static void main(String [] argv) {
 ProgrammatoreJava pj=new
 ProgrammatoreJava("Franco", "Bianchi",5);
 pj.stampa();
 }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

6. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

7. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
 String tipo="Sempre Verde";
 public int f()
 {
 System.out.println("Pioppo");
 int num=6;
 return num;
 }
 public static void main(String args[]) {
 Pioppo p= new Pioppo();
 System.out.println(p.f());
 }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo  
6
- D. Pioppo  
1
- E. Nessuna delle precedenti

8. Quale output si ottiene invocando il metodo m?

```

class F {
 private String [] a1 = new String [7];
 private String [] a4 = new String [2];
 private Boolean b3 = new Boolean(true);
 void m() {
 q(a1, new String [2], b3, b3);
 }
 void q(Object a2, String [] a3,
 Boolean b1, Object b2) {
 if(a2 == a1) {
 System.out.print(1);
 } else {
 System.out.print(0);
 }
 if(a3 == a4) {

```

```

 System.out.print(1);
 } else {
 System.out.print(0);
 }
 if(b2 == b1) {
 System.out.print(1);
 } else {
 System.out.print(0);
 }
}
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

### 9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
 public static void main(String [] argv)
 throws Exception {
 try {
 System.out.print(1);
 n();
 }
 catch(MyExc3 s) {
 System.out.print(2);
 throw(new Exception());
 }
 finally {
 System.out.print(3);
 throw(new MyExc3());
 }
 }
 static void n() throws Exception {
 try {
 System.out.print(4);
 switch(1) {
 case 1: throw(new MyExc2());
 case 2: throw(new MyExc1());
 case 3: throw(new MyExc3());
 }
 }
 catch(MyExc1 t) {
 throw(new MyExc3());
 }
 catch(MyExc3 d) {
 System.out.print(5);
 throw(new MyExc2());
 }
 catch(Exception h) {
 System.out.print(6);
 throw(new MyExc2());
 }
 }
}

```

- ```

    }
}
}

```
- A. Errore a tempo di compilazione
 - B. 1463Exception in thread main MyExc3
 - C. 143Exception in thread main MyExc3
 - D. 146666666... (ciclo infinito)
 - E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {

```



```

System.out.println("Cono: "
    + dimensioni + " - " +
    numLati + " lati per faccia.");
}

public static void main(String [] argv) {
    Cono c1 = new Cono();
    FiguraGeometrica f1;
    c1= (FiguraGeometrica) f1;
    c1.print();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

13. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }
}

```

```

}

Bottiglia(double litri) {
    this.numlitri=litri;
    System.out.println(litri);
}

public static void main(String args[]) {
    Bottiglia spumante=new Bottiglia();
    Bottiglia vinoRosso=new Bottiglia();
    Bottiglia vinoBianco=new Bottiglia(1.0);
    spumante=null;
    vinoBianco=spumante;
    /*QUI*/
    System.out.println("END");
}
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)

16. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Stella: Sole
Colore: rosa
 - D. Stella: Sole
Colore: bianco
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 72

1. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
System.out.println("Pioppo");
int num=n+1;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
        }
    }
}
```

```
        throw( new MyExc2() );
    }
}
static void q() {
    try {
        System.out.print(3);
        throw( new Exception() );
    }
    finally {
        throw( new MyExc3() );
    }
}
}
```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }
}
ProgrammatoreJava(String n, String c, int a){
```

```

super(n, c, a);
}
public static void main(String [] argv) {
    ProgrammatoreJava pj=new
        ProgrammatoreJava("Franco", "Bianchi",5);
    pj.stampa();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

5. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

6. Quale output si ottiene invocando il metodo m?

```

class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

7. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

8. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1 = (FiguraGeometrica) f1;
    }
}
```

```
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

11. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri) {
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

12. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione

- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
}

```

```

public static void main(String args[]) {
    Pioppo p= new Pioppo();
    System.out.println(p.f());
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

15. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
    }
}

```

```

}
}
catch( MyExc1 t ) {
    throw( new MyExc3() );
}
catch( MyExc3 d ) {
    System.out.print(5);
    throw( new MyExc2() );
}
catch( Exception h ) {
    System.out.print(6);
    throw( new MyExc2() );
}
}
}
}
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

16. Date le dichiarazioni:

```

Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 73

1. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

2. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato

E. Un oggetto può non esistere dopo la sua dichiarazione

3. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

4. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

5. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );

```

```

                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

7. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione

- C. Programmatore Java con 5 anni di esperienza
 - D. Programmatore Java con 0 anni di esperienza
 - E. Nessuna delle precedenti
-

8. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
System.out.println("Pioppo");
int num=n+1;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Pioppo
6
 - D. Pioppo
1
 - E. Nessuna delle precedenti
-

9. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
 - B. q = (Error []) w;
 - C. m = (Object [] []) w;
 - D. m = (Object [] []) q;
 - E. Nessuno dei precedenti
-

10. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Programmatore Java con 5 anni di esperienza
 - D. Programmatore Java con 0 anni di esperienza
 - E. Nessuna delle precedenti
-

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
    }
}
```

```

        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111

E. Nessuna delle precedenti

13. Quale output si ottiene invocando il metodo m?

```

class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

14. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {
    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {

```

```

    Cono c1 = new Cono();
    FiguraGeometrica f1 =c1;
    c1.print();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

15. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa

- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

16. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri) {
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 74

1. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
            case 1: throw( new MyExc2() );
            case 2: throw( new MyExc1() );
            case 3: throw( new MyExc3() );
```

```
        }
    }
    catch( MyExc1 t ) {
        throw( new MyExc3() );
    }
    catch( MyExc3 d ) {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception h ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
}
}
```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

3. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
```



```

        System.out.print(0);
    }
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

4. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class Bl {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

6. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia(){
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

7. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

8. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica fl =c1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
}
```

```

}
public static void main(String [] argv) {
    ProgrammatoreJava pj=new
        ProgrammatoreJava ("Franco", "Bianchi", 5);
    pj.stampa ();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

11. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

12. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1

E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione

C. Pioppo
6

D. Pioppo
1

E. Nessuna delle precedenti

15. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono cl = new Cono();
        FiguraGeometrica fl;
        cl= (FiguraGeometrica) fl;
        cl.print();
    }
}
```

A. Errore a tempo di compilazione

B. Errore a tempo di esecuzione

C. Cono: 8 - 3 lati per faccia

D. Cono: 3 - 3 lati per faccia

E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

A. Errore a tempo di compilazione

B. Errore a tempo di esecuzione

C. Programmatore Java con 5 anni di esperienza

D. Programmatore Java con 0 anni di esperienza

E. Nessuna delle precedenti

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 75

1. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print ();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;
```

```
public void print () {
    System.out.println("Cono: "
        + dimensioni + " - " +
        numLati + " lati per faccia.");
}

public static void main(String [] argv) {
    Cono c1 = new Cono();
    FiguraGeometrica f1 =c1;
    c1.print ();
}
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

3. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

4. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
}
```

```

void q(Object a2, String [] a3,
    Boolean b1, Object b2) {
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a3 == a4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

5. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

7. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {
    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
    }
}

```

```

    FiguraGeometrica f1;
    cl= (FiguraGeometrica) f1;
    cl.print ();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma ();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
super(n, c, a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava ("Franco", "Bianchi",5);
        pj.stampa ();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n ();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {

```



```

try {
    q();
}
finally {
    throw( new MyExc2() );
}
}
static void q() throws Exception {
    try {
    }
    catch( MyExc2 y ) {
        System.out.print(1);
        throw( new MyExc3() );
    }
    catch( MyExc3 g ) {
        System.out.print(2);
    }
    finally {
        throw( new MyExc3() );
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

13. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
    }
}

```

```

catch( MyExc3 i ) {
    System.out.print(1);
}
catch( MyExc2 w ) {
}
catch( MyExc1 j ) {
    throw( new MyExc3() );
}
finally {
    System.out.print(2);
    throw( new MyExc2() );
}
}
static void q() {
    try {
        System.out.print(3);
        throw( new Exception() );
    }
    finally {
        throw( new MyExc3() );
    }
}
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

15. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }
}

```

```

Bottiglia(double litri){
    this.numlitri=litri;
    System.out.println(litri);
}

public static void main(String args[]) {
    Bottiglia spumante=new Bottiglia();
    Bottiglia vinoRosso=new Bottiglia();
    Bottiglia vinoBianco=new Bottiglia(1.0);
    spumante=null;
    vinoBianco=spumante;
    /*QUI*/
    System.out.println("END");
}
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

16. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo `static` non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente `static` e `final`
- D. L'operatore `new` non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 76

1. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
```

```
Pioppo p= new Pioppo();
System.out.println(p.f(5));
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

```

```

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

```

}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

6. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

```

```

    Bottiglia() {
        System.out.println(marca);
    }

```

```

    Bottiglia(double litri) {
        this.numlitri=litri;
        System.out.println(litri);
    }

```

```

public static void main(String args[]) {
    Bottiglia spumante=new Bottiglia();
    Bottiglia vinoRosso=new Bottiglia();
    Bottiglia vinoBianco=new Bottiglia(1.0);
    spumante=null;
    vinoBianco=spumante;
    /*QUI*/
    System.out.println("END");
}
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

7. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

```

```

class Cono implements FiguraGeometrica {
    static final int numLati = 3;

```

```

public void print() {
    System.out.println("Cono: "
        + dimensioni + " - " +
        numLati + " lati per faccia.");
}

public static void main(String [] argv) {
    Cono c1 = new Cono();
    FiguraGeometrica f1;
    c1= (FiguraGeometrica) f1;
    c1.print();
}
}

        System.out.print(2);
        throw( new MyExc2() );
    }
}
static void q() {
    try {
        System.out.print(3);
        throw( new Exception() );
    }
    finally {
        throw( new MyExc3() );
    }
}
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

8. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {

```

10. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza

E. Nessuna delle precedenti

11. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

12. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){
```

```
        System.out.print("Programmatore Java con ");
        System.out.print(anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

13. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

14. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

15. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
System.out.println("Pioppo");
int num=6;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Pioppo
6
 - D. Pioppo
1
 - E. Nessuna delle precedenti
-

16. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
    }
}
```

```
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}
```

- A. Errore a tempo di compilazione
 - B. 1463Exception in thread main MyExc3
 - C. 143Exception in thread main MyExc3
 - D. 146666666... (ciclo infinito)
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 77

1. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

2. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print ();
}

class Cono implements FiguraGeometrica {
    static final int numLati = 3;
```

```
public void print () {
    System.out.println("Cono: "
        + dimensioni + " - " +
        numLati + " lati per faccia.");
}

public static void main(String [] argv) {
    Cono c1 = new Cono();
    FiguraGeometrica f1 =c1;
    c1.print ();
}
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione

C. Stella: Sole
Colore: rosa

D. Stella: Sole
Colore: bianco

E. Nessuna delle precedenti

4. Date le dichiarazioni:

```
Object [] [] m;  
Error [] q;  
Object [] w;  
q = new Error [9];  
m = new Object [5] [2];  
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

A. q = (Error []) m;

B. q = (Error []) w;

C. m = (Object [] []) w;

D. m = (Object [] []) q;

E. Nessuno dei precedenti

5. Qual è l'output di questo codice?

```
class Stella {  
    String nome="Stella";  
    String colore="rosa";  
  
    public Stella(String nome) {  
        this.nome = nome;  
    }  
}  
  
public class Sole extends Stella {  
  
    public Sole(String c) {  
        super("Sole");  
        this.colore = c;  
    }  
  
    public static void main(String [] argv) {  
        Sole s = new Sole("bianco");  
        System.out.println("Stella: " + s.nome);  
        System.out.println("Colore: "  
            + s.colore);  
    }  
}
```

A. Errore a tempo di compilazione

B. Errore a tempo di esecuzione

C. Stella: Sole
Colore: rosa

D. Stella: Sole
Colore: bianco

E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends Exception { }  
public class C1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            n();  
        }  
        catch( MyExc3 s ) {  
            System.out.print (2);  
            throw( new Exception() );  
        }  
        finally {  
            System.out.print (3);  
            throw( new MyExc3() );  
        }  
    }  
    static void n() throws Exception {  
        try {  
            System.out.print (4);  
            switch(1) {  
                case 1: throw( new MyExc2() );  
                case 2: throw( new MyExc1() );  
                case 3: throw( new MyExc3() );  
            }  
        }  
        catch( MyExc1 t ) {  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 d ) {  
            System.out.print (5);  
            throw( new MyExc2() );  
        }  
        catch( Exception h ) {  
            System.out.print (6);  
            throw( new MyExc2() );  
        }  
    }  
}
```

A. Errore a tempo di compilazione

B. 1463Exception in thread main MyExc3

C. 143Exception in thread main MyExc3

D. 146666666... (ciclo infinito)

E. Nessuna delle precedenti

7. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `d = c;`
- B. `d = (Object [] []) e;`
- C. `c = (String []) d;`
- D. `d = (Object [] []) c;`
- E. `d = e;`

8. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
super(n, c, a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava ("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

9. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo `static` non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente `static` e `final`
- D. L'operatore `new` non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

10. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    void programma(){}
    ProgrammatoreJava(String n, String c, int a){
super(n, c, a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava ("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
System.out.println("Pioppo");
int num=n+1;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

14. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia(){
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
    }
}

```

```

Bottiglia vinoRosso=new Bottiglia();
Bottiglia vinoBianco=new Bottiglia(1.0);
spumante=null;
vinoBianco=spumante;
/*QUI*/
System.out.println("END");
}
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

15. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
System.out.println("Pioppo");
int num=6;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1

E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 78

1. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo `static` non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente `static` e `final`
- D. L'operatore `new` non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

```
Bottiglia(double litri){
    this.numlitri=litri;
    System.out.println(litri);
}

public static void main(String args[]){
    Bottiglia spumante=new Bottiglia();
    Bottiglia vinoRosso=new Bottiglia();
    Bottiglia vinoBianco=new Bottiglia(1.0);
    spumante=null;
    vinoBianco=spumante;
    /*QUI*/
    System.out.println("END");
}
}
```

2. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];
```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

4. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]){
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

3. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia(){
        System.out.println(marca);
    }
}
```

5. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
    }
}
```

```
finally {
    System.out.print(3);
    throw( new MyExc3() );
}
}
static void n() throws Exception {
    try {
        System.out.print(4);
        switch(1) {
            case 1: throw( new MyExc2() );
            case 2: throw( new MyExc1() );
            case 3: throw( new MyExc3() );
        }
    }
    catch( MyExc1 t ) {
        throw( new MyExc3() );
    }
    catch( MyExc3 d ) {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception h ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
}
}
```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

7. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

8. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
```

```

    nome=n;
    cognome=c;
    anniEsperienza=a;
}
abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
void stampa(){

    System.out.print("Programmatore Java con ");
    System.out.print (anniEsperienza);
    System.out.println(" anni di esperienza");
}
void programma() {}
ProgrammatoreJava(String n, String c, int a){
super(n,c,a);
}
public static void main(String [] argv) {
    ProgrammatoreJava pj=new
        ProgrammatoreJava("Franco", "Bianchi",5);
    pj.stampa();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){

        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

void stampa(){

    System.out.print("Programmatore Java con ");
    System.out.print (anniEsperienza);
    System.out.println(" anni di esperienza");
}

ProgrammatoreJava(String n, String c, int a){
super(n,c,a);

```

```

}
public static void main(String [] argv) {
    ProgrammatoreJava pj=new
        ProgrammatoreJava("Franco", "Bianchi",5);
    pj.stampa();
}
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
System.out.println("Pioppo");
int num=n+1;
return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

12. Quale output si ottiene invocando il metodo m?

```

class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 100
- C. 011
- D. 101

E. 010

13. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print();
    }
}

```

```
}  
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Cono: 8 - 3 lati per faccia
 - D. Cono: 3 - 3 lati per faccia
 - E. Nessuna delle precedenti
-

15. Qual è l'output di questo codice?

```
class Stella {  
    String nome="Stella";  
    String colore="rosa";  
  
    public Stella(String nome) {  
        this.nome = nome;  
    }  
}  
  
public class Sole extends Stella {  
  
    public Sole(String c) {  
        super("Sole");  
        this.colore = c;  
    }  
  
    public static void main(String [] argv) {  
        Sole s = new Sole("bianco");  
        System.out.println("Stella: " + s.nome);  
        System.out.println("Colore: "  
            + s.colore);  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa

D. Stella: Sole
Colore: bianco

E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```
class Stella {  
    String nome="Stella";  
    final String colore="rosa";  
  
    public Stella(String nome) {  
        this.nome = nome;  
    }  
}  
  
public class Sole extends Stella {  
  
    public Sole(String c) {  
        super("Sole");  
        this.colore = c;  
    }  
  
    public static void main(String [] argv) {  
        Sole s = new Sole("bianco");  
        System.out.println("Stella: " + s.nome);  
        System.out.println("Colore: "  
            + s.colore);  
    }  
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Stella: Sole
Colore: rosa
 - D. Stella: Sole
Colore: bianco
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 79

1. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma() {}
    ProgrammatoreJava(String n, String c, int a){
    super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
```

```
public static void main(String [] argv)
    throws Exception {
    try {
        q();
    }
    finally {
        throw( new MyExc2() );
    }
}
static void q() throws Exception {
    try {
    }
    catch( MyExc2 y ) {
        System.out.print(1);
        throw( new MyExc3() );
    }
    catch( MyExc3 g ) {
        System.out.print(2);
    }
    finally {
        throw( new MyExc3() );
    }
}
}
```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
    }
}
```

```

        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 31
- B. Errore a tempo di compilazione
- C. 312Exception in thread main MyExc2
- D. 312
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco

E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
                          int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){

        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

7. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}

```

- A. Errore a tempo di compilazione

- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {
    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

9. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C. Un attributo non può essere contemporaneamente static e final
- D. L'operatore new non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

10. Qual è l'output di questo codice?


```

interface FiguraGeometrica {
    static int dimensioni = 10;
    void print ();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print () {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print ();
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

11. Date le dichiarazioni:

```

Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
m = new Object [5] [2];
w = new Object [4] [5];

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

12. Qual è l'output di questo codice?

```

class Stella {
    String nome="Stella";
    String colore="rosa";

```

```

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

13. Date le dichiarazioni:

```

String [] c;
Object [] [] d;
Object [] e;

```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = (Object [] []) e;
- C. c = (String []) d;
- D. d = (Object [] []) c;
- E. d = e;

14. Quale output si ottiene invocando il metodo m?

```

class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```

    }
    if(a3 == a4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

15. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia(){
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
    }
}

```

```

        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

16. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 80

1. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }

    public static void main(String [] argv) {
        Sole s = new Sole("bianco");
        System.out.println("Stella: " + s.nome);
        System.out.println("Colore: "
            + s.colore);
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
}
```

```
}
abstract void programma();
}
public class ProgrammatoreJava
    extends Programmatore {

    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }

    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Programmatore Java con 5 anni di esperienza
- D. Programmatore Java con 0 anni di esperienza
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class Stella {
    String nome="Stella";
    final String colore="rosa";

    public Stella(String nome) {
        this.nome = nome;
    }
}

public class Sole extends Stella {

    public Sole(String c) {
        super("Sole");
        this.colore = c;
    }
}
```

```

}

public static void main(String [] argv) {
    Sole s = new Sole("bianco");
    System.out.println("Stella: " + s.nome);
    System.out.println("Colore: "
        + s.colore);
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Stella: Sole
Colore: rosa
- D. Stella: Sole
Colore: bianco
- E. Nessuna delle precedenti

4. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Bottiglia {
    static String marca="Ferrari";
    double numlitri=1.5;

    Bottiglia() {
        System.out.println(marca);
    }

    Bottiglia(double litri){
        this.numlitri=litri;
        System.out.println(litri);
    }

    public static void main(String args[]) {
        Bottiglia spumante=new Bottiglia();
        Bottiglia vinoRosso=new Bottiglia();
        Bottiglia vinoBianco=new Bottiglia(1.0);
        spumante=null;
        vinoBianco=spumante;
        /*QUI*/
        System.out.println("END");
    }
}

```

- A. 0
- B. 1
- C. 2
- D. 4
- E. 5

5. Qual è l'output di questo codice?

```

interface albero { int f(int i);}
class Pioppo implements albero {
    String tipo="Sempre Verde";
    public int f(int n)
    {
        System.out.println("Pioppo");
        int num=n+1;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f(5));
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Pioppo
6
- D. Pioppo
1
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```

abstract class Programmatore {
    String nome;
    String cognome;
    int anniEsperienza;

    public Programmatore(String n, String c,
        int a){
        nome=n;
        cognome=c;
        anniEsperienza=a;
    }
    abstract void programma();
}

public class ProgrammatoreJava
    extends Programmatore {
    void stampa(){
        System.out.print("Programmatore Java con ");
        System.out.print (anniEsperienza);
        System.out.println(" anni di esperienza");
    }
    void programma() {}
    ProgrammatoreJava(String n, String c, int a){
        super(n,c,a);
    }
    public static void main(String [] argv) {
        ProgrammatoreJava pj=new
            ProgrammatoreJava("Franco", "Bianchi",5);
        pj.stampa();
    }
}

```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Programmatore Java con 5 anni di esperienza
 - D. Programmatore Java con 0 anni di esperienza
 - E. Nessuna delle precedenti
-

7. Qual è l'output di questo codice?

```
interface albero { int f(int i);}
class Pioppo implements albero{
    String tipo="Sempre Verde";
    public int f()
    {
        System.out.println("Pioppo");
        int num=6;
        return num;
    }
    public static void main(String args[]) {
        Pioppo p= new Pioppo();
        System.out.println(p.f());
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Pioppo
6
 - D. Pioppo
1
 - E. Nessuna delle precedenti
-

8. Date le dichiarazioni:

```
String [] c;
Object [] [] d;
Object [] e;
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
 - B. d = (Object [] []) e;
 - C. c = (String []) d;
 - D. d = (Object [] []) c;
 - E. d = e;
-

9. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}
```

```
class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1 =c1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
 - B. Errore a tempo di esecuzione
 - C. Cono: 8 - 3 lati per faccia
 - D. Cono: 3 - 3 lati per faccia
 - E. Nessuna delle precedenti
-

10. Quale output si ottiene invocando il metodo m?

```
class F {
    private String [] a1 = new String [7];
    private String [] a4 = new String [2];
    private Boolean b3 = new Boolean(true);
    void m() {
        q(a1, new String [2], b3, b3);
    }
    void q(Object a2, String [] a3,
        Boolean b1, Object b2) {
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 100
- C. 011
- D. 101
- E. 010

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
        }
        catch( MyExc3 s ) {
            System.out.print(2);
            throw( new Exception() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(4);
            switch(1) {
                case 1: throw( new MyExc2() );
                case 2: throw( new MyExc1() );
                case 3: throw( new MyExc3() );
            }
        }
        catch( MyExc1 t ) {
            throw( new MyExc3() );
        }
        catch( MyExc3 d ) {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception h ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1463Exception in thread main MyExc3
- C. 143Exception in thread main MyExc3
- D. 146666666... (ciclo infinito)
- E. Nessuna delle precedenti

12. Date le dichiarazioni:

```
Object [] [] m;
Error [] q;
Object [] w;
q = new Error [9];
```

```
m = new Object [5] [2];
w = new Object [4] [5];
```

indicare quale dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error []) m;
- B. q = (Error []) w;
- C. m = (Object [] []) w;
- D. m = (Object [] []) q;
- E. Nessuno dei precedenti

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 g ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 111111... (ciclo infinito)
- C. Exception in thread main MyExc2
- D. 1111
- E. Nessuna delle precedenti

14. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza

- C. Un attributo non può essere contemporaneamente **static** e **final**
- D. L'operatore **new** non restituisce un riferimento all'oggetto appena creato
- E. Un oggetto può non esistere dopo la sua dichiarazione

- B. Errore a tempo di compilazione
- C. `312Exception` in thread main `MyExc2`
- D. `312`
- E. Nessuna delle precedenti

15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            q();
        }
        catch( MyExc3 i ) {
            System.out.print(1);
        }
        catch( MyExc2 w ) {
        }
        catch( MyExc1 j ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

A. 31

16. Qual è l'output di questo codice?

```
interface FiguraGeometrica {
    static int dimensioni = 10;
    void print();
}

class Cono implements FiguraGeometrica {

    static final int numLati = 3;

    public void print() {
        System.out.println("Cono: "
            + dimensioni + " - " +
            numLati + " lati per faccia.");
    }

    public static void main(String [] argv) {
        Cono c1 = new Cono();
        FiguraGeometrica f1;
        c1= (FiguraGeometrica) f1;
        c1.print();
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Cono: 8 - 3 lati per faccia
- D. Cono: 3 - 3 lati per faccia
- E. Nessuna delle precedenti

Prova n. 1

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1

Prova d'esame

prof. Piero A. Bonatti

17-07-2014

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

1	■				
2		■			
3			■		
4	■				
5			■		
6					■
7					■
8		■			
9				■	
10			■		
11			■		
12		■			
13	■				
14	■				
15				■	
16	■				

Risultato prova n. 1:

1	■				
2			■		
3				■	
4			■		
5		■			
6			■		
7	■				
8		■			
9	■				
10			■		
11				■	
12		■			
13	■				
14					■
15	■				
16					■

Risultato prova n. 2:

1			■		
2				■	
3				■	
4			■		
5					■
6		■			
7		■			
8	■				
9			■		
10		■			
11			■		
12	■				
13	■				
14					■
15	■				
16	■				

Risultato prova n. 3:

1		■		
2			■	
3	■			
4		■		
5	■			
6			■	
7	■			
8	■			
9			■	
10				■
11			■	
12			■	
13			■	
14	■			
15		■		
16				■

Risultato prova n. 4:

1	■				
2				■	
3			■		
4					■
5		■			
6	■				
7	■				
8					■
9			■		
10			■		
11			■		
12	■				
13	■				
14				■	
15		■			
16		■			

Risultato prova n. 5:

1	■				
2					■
3	■				
4			■		
5		■			
6		■			
7	■				
8					■
9	■				
10		■			
11				■	
12			■		
13			■		
14			■		
15	■				
16				■	

Risultato prova n. 6:

1				■	
2		■			
3			■		
4		■			
5	■				
6					■
7				■	
8			■		
9		■			
10	■				
11	■				
12	■				
13					■
14	■				
15			■		
16			■		

Risultato prova n. 7:

1	■				
2					■
3	■				
4			■		
5				■	
6		■			
7				■	
8	■				
9		■			
10	■				
11			■		
12	■				
13			■		
14		■			
15					■
16			■		

Risultato prova n. 8:

1			■		
2		■			
3				■	
4			■		
5	■				
6	■				
7			■		
8	■				
9			■		
10					■
11				■	
12		■			
13		■			
14	■				
15					■
16	■				

Risultato prova n. 9:

1				■	
2				■	
3			■		
4	■				
5			■		
6			■		
7					■
8	■				
9		■			
10		■			
11		■			
12	■				
13	■				
14			■		
15					■
16	■				

Risultato prova n. 10:

1		■			
2	■				
3			■		
4			■		
5					■
6	■				
7	■				
8			■		
9	■				
10		■			
11	■				
12			■		
13					■
14				■	
15				■	
16		■			

Risultato prova n. 11:

1					■
2			■		
3				■	
4			■		
5				■	
6	■				
7	■				
8		■			
9	■				
10					■
11		■			
12	■				
13	■				
14		■			
15			■		
16			■		

Risultato prova n. 12:

1	■				
2				■	
3			■		
4		■			
5				■	
6		■			
7		■			
8			■		
9					■
10	■				
11	■				
12					■
13			■		
14	■				
15			■		
16	■				

Risultato prova n. 13:

1			■		
2			■		
3	■				
4				■	
5		■			
6				■	
7			■		
8	■				
9	■				
10		■			
11					■
12			■		
13					■
14		■			
15	■				
16	■				

Risultato prova n. 14:

1			■		
2		■			
3					■
4			■		
5			■		
6			■		
7				■	
8	■				
9		■			
10	■				
11				■	
12	■				
13	■				
14		■			
15					■
16	■				

Risultato prova n. 15:

1	■				
2				■	
3	■				
4	■				
5			■		
6		■			
7		■			
8			■		
9	■				
10				■	
11			■		
12			■		
13		■			
14					■
15					■
16	■				

Risultato prova n. 16:

1	■				
2			■		
3			■		
4	■				
5		■			
6			■		
7	■				
8					■
9			■		
10				■	
11	■				
12		■			
13					■
14	■				
15				■	
16		■			

Risultato prova n. 17:

1	■				
2					■
3				■	
4	■				
5	■				
6	■				
7			■		
8			■		
9		■			
10	■				
11				■	
12		■			
13			■		
14			■		
15		■			
16					■

Risultato prova n. 18:

1			■		
2					■
3				■	
4	■				
5		■			
6				■	
7	■				
8			■		
9	■				
10			■		
11		■			
12					■
13			■		
14	■				
15	■				
16		■			

Risultato prova n. 19:

1	■				
2	■				
3	■				
4				■	
5			■		
6	■				
7					■
8			■		
9	■				
10		■			
11		■			
12			■		
13					■
14		■			
15				■	
16			■		

Risultato prova n. 20:

1					■
2			■		
3		■			
4	■				
5		■			
6			■		
7				■	
8	■				
9		■			
10	■				
11	■				
12				■	
13			■		
14			■		
15					■
16	■				

Risultato prova n. 21:

1				■	
2					■
3			■		
4	■				
5		■			
6		■			
7	■				
8					■
9	■				
10			■		
11			■		
12	■				
13				■	
14		■			
15	■				
16			■		

Risultato prova n. 22:

1		■			
2			■		
3			■		
4	■				
5	■				
6		■			
7					■
8	■				
9			■		
10	■				
11			■		
12	■				
13					■
14				■	
15		■			
16				■	

Risultato prova n. 23:

1	■				
2	■				
3	■				
4			■		
5					■
6	■				
7	■				
8		■			
9		■			
10					■
11		■			
12				■	
13				■	
14			■		
15			■		
16			■		

Risultato prova n. 24:

1	■				
2				■	
3			■		
4	■				
5				■	
6			■		
7			■		
8			■		
9		■			
10	■				
11	■				
12	■				
13		■			
14					■
15					■
16		■			

Risultato prova n. 25:

1		■			
2				■	
3			■		
4	■				
5		■			
6			■		
7	■				
8			■		
9	■				
10			■		
11					■
12					■
13		■			
14	■				
15				■	
16	■				

Risultato prova n. 26:

1				■
2		■		
3	■			
4			■	
5		■		
6			■	
7			■	
8			■	
9	■			
10			■	
11	■			
12	■			
13		■		
14			■	
15	■			
16				■

Risultato prova n. 27:

1			■		
2	■				
3	■				
4		■			
5	■				
6	■				
7				■	
8			■		
9			■		
10	■				
11			■		
12		■			
13					■
14					■
15				■	
16		■			

Risultato prova n. 28:

1				■	
2			■		
3		■			
4			■		
5	■				
6		■			
7					■
8			■		
9	■				
10				■	
11	■				
12		■			
13	■				
14	■				
15					■
16			■		

Risultato prova n. 29:

1		■			
2	■				
3	■				
4				■	
5					■
6	■				
7	■				
8					■
9				■	
10	■				
11		■			
12			■		
13			■		
14			■		
15			■		
16		■			

Risultato prova n. 30:

1					■
2			■		
3					■
4				■	
5			■		
6			■		
7	■				
8	■				
9	■				
10		■			
11			■		
12	■				
13		■			
14		■			
15				■	
16	■				

Risultato prova n. 31:

1					■
2	■				
3			■		
4		■			
5		■			
6	■				
7					■
8				■	
9		■			
10				■	
11	■				
12			■		
13			■		
14	■				
15	■				
16			■		

Risultato prova n. 32:

1					■
2	■				
3	■				
4			■		
5		■			
6	■				
7			■		
8			■		
9				■	
10	■				
11		■			
12			■		
13		■			
14					■
15	■				
16				■	

Risultato prova n. 33:

1				■	
2				■	
3	■				
4		■			
5	■				
6					■
7	■				
8	■				
9		■			
10					■
11			■		
12	■				
13			■		
14			■		
15			■		
16		■			

Risultato prova n. 34:

1	■				
2		■			
3			■		
4				■	
5	■				
6	■				
7	■				
8		■			
9	■				
10		■			
11					■
12			■		
13			■		
14					■
15				■	
16			■		

Risultato prova n. 35:

1					■
2	■				
3			■		
4			■		
5	■				
6	■				
7					■
8	■				
9			■		
10				■	
11		■			
12		■			
13	■				
14				■	
15		■			
16			■		

Risultato prova n. 36:

1			■		
2			■		
3		■			
4	■				
5			■		
6					■
7					■
8		■			
9				■	
10	■				
11	■				
12			■		
13	■				
14		■			
15	■				
16				■	

Risultato prova n. 37:

1			■		
2					■
3	■				
4			■		
5			■		
6		■			
7		■			
8			■		
9	■				
10	■				
11		■			
12	■				
13	■				
14				■	
15					■
16				■	

Risultato prova n. 38:

1	■				
2	■				
3	■				
4				■	
5			■		
6			■		
7		■			
8	■				
9					■
10		■			
11			■		
12				■	
13					■
14	■				
15			■		
16		■			

Risultato prova n. 39:

1					■
2			■		
3	■				
4		■			
5	■				
6			■		
7				■	
8					■
9			■		
10		■			
11			■		
12	■				
13		■			
14	■				
15				■	
16	■				

Risultato prova n. 40:

1			■		
2	■				
3	■				
4	■				
5		■			
6				■	
7	■				
8					■
9				■	
10			■		
11					■
12	■				
13		■			
14			■		
15			■		
16		■			

Risultato prova n. 41:

1	■			
2		■		
3				■
4			■	
5				■
6	■			
7		■		
8		■		
9			■	
10			■	
11	■			
12	■			
13			■	
14				■
15	■			
16			■	

Risultato prova n. 42:

1	■				
2			■		
3		■			
4	■				
5			■		
6					■
7					■
8	■				
9				■	
10			■		
11	■				
12	■				
13			■		
14				■	
15		■			
16		■			

Risultato prova n. 43:

1			■		
2	■				
3				■	
4			■		
5			■		
6			■		
7		■			
8					■
9	■				
10					■
11	■				
12		■			
13				■	
14	■				
15	■				
16		■			

Risultato prova n. 44:

1		■			
2			■		
3			■		
4		■			
5		■			
6	■				
7				■	
8	■				
9			■		
10			■		
11	■				
12	■				
13	■				
14					■
15					■
16				■	

Risultato prova n. 45:

1					■
2				■	
3		■			
4	■				
5	■				
6	■				
7			■		
8				■	
9	■				
10		■			
11		■			
12			■		
13			■		
14			■		
15	■				
16					■

Risultato prova n. 46:

1				■	
2		■			
3			■		
4			■		
5	■				
6					■
7				■	
8		■			
9					■
10			■		
11	■				
12			■		
13		■			
14	■				
15	■				
16	■				

Risultato prova n. 47:

1	■				
2				■	
3					■
4	■				
5		■			
6			■		
7			■		
8		■			
9			■		
10				■	
11		■			
12					■
13	■				
14			■		
15	■				
16	■				

Risultato prova n. 48:

1		■			
2			■		
3			■		
4					■
5		■			
6				■	
7			■		
8				■	
9			■		
10	■				
11					■
12		■			
13	■				
14	■				
15	■				
16	■				

Risultato prova n. 49:

1		■			
2		■			
3	■				
4		■			
5					■
6	■				
7				■	
8					■
9	■				
10				■	
11			■		
12	■				
13	■				
14			■		
15			■		
16			■		

Risultato prova n. 50:

1			■		
2	■				
3			■		
4		■			
5			■		
6	■				
7	■				
8		■			
9	■				
10				■	
11			■		
12				■	
13					■
14	■				
15		■			
16					■

Risultato prova n. 51:

1			■		
2		■			
3	■				
4			■		
5					■
6			■		
7	■				
8	■				
9	■				
10				■	
11	■				
12			■		
13				■	
14		■			
15					■
16		■			

Risultato prova n. 52:

1				■	
2					■
3	■				
4		■			
5	■				
6	■				
7		■			
8			■		
9					■
10				■	
11			■		
12	■				
13	■				
14			■		
15		■			
16			■		

Risultato prova n. 53:

1			■		
2	■				
3	■				
4			■		
5		■			
6					■
7					■
8		■			
9				■	
10		■			
11	■				
12	■				
13			■		
14				■	
15	■				
16			■		

Risultato prova n. 54:

1				■	
2			■		
3		■			
4	■				
5			■		
6		■			
7					■
8					■
9				■	
10	■				
11			■		
12		■			
13	■				
14			■		
15	■				
16	■				

Risultato prova n. 55:

1					■
2	■				
3			■		
4	■				
5					■
6	■				
7		■			
8			■		
9			■		
10		■			
11		■			
12	■				
13				■	
14			■		
15				■	
16	■				

Risultato prova n. 56:

1					■
2	■				
3	■				
4			■		
5		■			
6	■				
7		■			
8			■		
9	■				
10		■			
11	■				
12				■	
13			■		
14					■
15			■		
16				■	

Risultato prova n. 57:

1	■				
2		■			
3	■				
4			■		
5			■		
6		■			
7	■				
8		■			
9	■				
10				■	
11				■	
12	■				
13					■
14			■		
15			■		
16					■

Risultato prova n. 58:

1		■			
2		■			
3			■		
4	■				
5		■			
6			■		
7			■		
8	■				
9					■
10	■				
11					■
12	■				
13				■	
14			■		
15	■				
16				■	

Risultato prova n. 59:

1			■		
2					■
3		■			
4				■	
5			■		
6					■
7	■				
8		■			
9	■				
10			■		
11				■	
12	■				
13			■		
14	■				
15		■			
16	■				

Risultato prova n. 60:

1	■				
2			■		
3			■		
4	■				
5	■				
6		■			
7	■				
8				■	
9			■		
10		■			
11				■	
12			■		
13	■				
14					■
15					■
16		■			

Risultato prova n. 61:

1	■				
2		■			
3					■
4					■
5	■				
6		■			
7			■		
8		■			
9			■		
10	■				
11			■		
12	■				
13	■				
14				■	
15			■		
16				■	

Risultato prova n. 62:

1			■		
2	■				
3					■
4					■
5		■			
6			■		
7				■	
8	■				
9	■				
10		■			
11			■		
12			■		
13				■	
14	■				
15		■			
16	■				

Risultato prova n. 63:

1					■
2			■		
3	■				
4			■		
5		■			
6		■			
7	■				
8	■				
9			■		
10					■
11	■				
12			■		
13		■			
14				■	
15	■				
16				■	

Risultato prova n. 64:

1		■			
2			■		
3	■				
4	■				
5			■		
6					■
7			■		
8	■				
9				■	
10	■				
11		■			
12					■
13	■				
14		■			
15			■		
16				■	

Risultato prova n. 65:

1		■			
2				■	
3			■		
4					■
5			■		
6		■			
7	■				
8	■				
9				■	
10			■		
11	■				
12			■		
13	■				
14		■			
15	■				
16					■

Risultato prova n. 66:

1		■			
2	■				
3					■
4			■		
5			■		
6		■			
7	■				
8	■				
9	■				
10		■			
11			■		
12			■		
13					■
14				■	
15	■				
16				■	

Risultato prova n. 67:

1		■			
2		■			
3	■				
4			■		
5			■		
6			■		
7	■				
8				■	
9					■
10			■		
11		■			
12				■	
13	■				
14	■				
15					■
16	■				

Risultato prova n. 68:

1			■		
2		■			
3		■			
4	■				
5			■		
6		■			
7	■				
8				■	
9	■				
10				■	
11					■
12					■
13	■				
14			■		
15			■		
16	■				

Risultato prova n. 69:

1	■				
2	■				
3			■		
4	■				
5		■			
6		■			
7			■		
8			■		
9			■		
10					■
11	■				
12				■	
13		■			
14				■	
15	■				
16					■

Risultato prova n. 70:

1					■
2		■			
3	■				
4			■		
5			■		
6					■
7	■				
8				■	
9		■			
10				■	
11	■				
12			■		
13			■		
14		■			
15	■				
16	■				

Risultato prova n. 71:

1			■		
2		■			
3	■				
4					■
5		■			
6				■	
7					■
8	■				
9	■				
10			■		
11			■		
12				■	
13	■				
14	■				
15		■			
16			■		

Risultato prova n. 72:

1	■				
2					■
3	■				
4		■			
5	■				
6		■			
7			■		
8			■		
9			■		
10	■				
11		■			
12	■				
13				■	
14					■
15				■	
16			■		

Risultato prova n. 73:

1			■		
2		■			
3				■	
4		■			
5	■				
6			■		
7					■
8				■	
9					■
10	■				
11		■			
12			■		
13	■				
14	■				
15	■				
16			■		

Risultato prova n. 74:

1				■	
2					■
3			■		
4				■	
5			■		
6			■		
7	■				
8	■				
9		■			
10	■				
11	■				
12	■				
13		■			
14		■			
15			■		
16					■

Risultato prova n. 75:

1				■	
2			■		
3					■
4	■				
5	■				
6			■		
7	■				
8		■			
9		■			
10			■		
11				■	
12	■				
13			■		
14					■
15	■				
16		■			

Risultato prova n. 76:

1				■	
2					■
3	■				
4			■		
5				■	
6		■			
7		■			
8	■				
9					■
10			■		
11			■		
12		■			
13	■				
14			■		
15	■				
16	■				

Risultato prova n. 77:

1					■
2			■		
3			■		
4	■				
5		■			
6		■			
7		■			
8			■		
9	■				
10	■				
11			■		
12				■	
13					■
14	■				
15				■	
16	■				

Risultato prova n. 78:

1			■		
2	■				
3		■			
4	■				
5		■			
6	■				
7	■				
8					■
9					■
10	■				
11			■		
12				■	
13		■			
14				■	
15			■		
16			■		

Risultato prova n. 79:

1				■	
2	■				
3	■				
4			■		
5			■		
6			■		
7	■				
8		■			
9					■
10				■	
11		■			
12			■		
13	■				
14					■
15		■			
16	■				

Risultato prova n. 80: