

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. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

2. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }
    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}
class Tree extends Plant {
    public Tree(String f){
        super(f);
    }
    public Tree(){}
    public void stampa() {
        System.out.println("Tree: "
            + family);
    }
    public static void main(String[] args){
        Plant a = new Tree(Olive);
        a.stampa();
    }
}
```

- A. Plant: Olive

B. Tree: Olive

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

3. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}
class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea(){}

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle", 3);
        ((Triangle)t).print();
    }
}
```

A. Geometric figure with 0 angles.

B. Geometric figure with 3 angles.

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

4. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. s = (B1) y;
- B. s = (B1) g;
- C. s = y;
- D. g = y;
- E. s = g;

5. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

6. Quale output si ottiene invocando il metodo q?

```

class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

7. Qual'è l'output di questo codice?

```

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

```

```

    finally {
        System.out.print(3);
        throw( new MyExc2() );
    }
}
}

```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

8. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```

public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}

```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

9. Date le dichiarazioni:

```

class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();

```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

10. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}

```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

11. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    public static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
- D. Un'istanza della classe I può essere costruita con new C.I().
- E. Nessuna delle precedenti.

12. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

13. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
    }
}
```

```
    angles = a;
}

abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

15. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
    }
}
```

```

        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
 - B. 1342
 - C. 134
 - D. 134Exception in thread "main" java.lang.Exception
 - E. Nessuna delle precedenti
-

16. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

```

```

    }
}
class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
 - B. XYZ
 - C. ZZZ
 - D. Errore a tempo di compilazione.
 - 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 programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
}
```

```
}
static void p() throws Exception {
    try {
        throw( new MyExc2() );
    }
    catch( MyExc2 f ) {
        System.out.print(1);
        throw( new MyExc3() );
    }
    catch( Exception d ) {
        System.out.print(2);
    }
    finally {
        System.out.print(3);
        throw( new MyExc2() );
    }
}
```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

3. Qual'è l'output di questo codice?

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

```

        throw( new Exception() );
    }
    finally {
        System.out.print (4);
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

4. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    public static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
- D. Un'istanza della classe I può essere costruita con new C.I().
- E. Nessuna delle precedenti.

5. Quale output si ottiene invocando il metodo q?

```

class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print (1);
        } else {
            System.out.print (0);
        }
        if(a3 == a1) {

```

```

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

```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

6. Qual'è l'output di questo programma?

```

class Plant {

    String family;
    protected Plant (String f) {
        family = f;
    }

    protected void stampa() {
        System.out.println ("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree (String f) {
        super (f);
    }
    public Tree () {}

    public void stampa() {
        System.out.println ("Tree: "
            + family);
    }

    public static void main (String[] args) {
        Plant a = new Tree (Olive);
        a.stampa ();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

8. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}
```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

9. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
```

```
int angles;

protected Figure(String t, int a) {
    type = t;
    angles = a;
}

abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

10. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
```

```

    Plant a = new Tree("Olive");
    a.stampa();
}
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

12. Date le dichiarazioni:

```

class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();

```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

13. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. s = (B1) y;
- B. s = (B1) g;
- C. s = y;
- D. g = y;
- E. s = g;

14. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    void getArea(){}

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

15. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ

D. Errore a tempo di compilazione.

E. Nessuna delle precedenti.

16. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}
```

A. 4

B. 5

C. 6

D. 7

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 programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

```
abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea(){}

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }
}
```

3. Date le dichiarazioni:

```
class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Sub1 g;
B1 s;
Super y;
```

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

- A. s = (B1) y;
- B. s = (B1) g;
- C. s = y;
- D. g = y;
- E. s = g;

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

5. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();
```

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

- A. s = (B1) z;
- B. z = (B2) s;

- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

6. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}
```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

7. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }
    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}
class Tree extends Plant {
    public Tree(String f){
```

```

    super(f);
}
public Tree() {}

public void stampa() {
    System.out.println("Tree: "
        + family);
}

public static void main(String[] args) {
    Plant a = new Tree(Olive);
    a.stampa();
}
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

9. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    Plant(String f) {
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {
    public Tree(String f) {
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args) {
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

10. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

11. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    public static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
- D. Un'istanza della classe I può essere costruita con new C.I().
- E. Nessuna delle precedenti.

12. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

13. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
```

```
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}
```

- A. 4
- B. 5
- C. 6
- D. 7
- 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 B1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }

    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}
```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

15. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

A. 101

B. 110

C. 100

D. 011

E. 111

16. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

A. Errore a tempo di compilazione per errata dichiarazione di un array.

B. Viene lanciata una NullPointerException.

C. Viene lanciata una ArrayIndexOutOfBoundsException.

D. null

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. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}
```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

2. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print() {
        System.out.print("Geometric figure with ");
```

```
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
- D. Un'istanza della classe I può essere costruita con new C.I().
- E. Nessuna delle precedenti.

4. Date le dichiarazioni:

```
class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Sub1 g;
B1 s;
Super y;
```

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

- A. `s = (B1) y;`
- B. `s = (B1) g;`
- C. `s = y;`
- D. `g = y;`
- E. `s = g;`

5. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2 ();
t = new B1 ();
s = new B1 ();
```

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

- A. `s = (B1) z;`
- B. `z = (B2) s;`
- C. `z = (B2) t;`
- D. `s = (B1) t;`
- E. Nessuno dei precedenti

6. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea() {}

    public void print() {
        System.out.print("Geometric figure with ");
```

```
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

8. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX

- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

9. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}
```

- A. 12312
- B. Errore a tempo di compilazione

11. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}
```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

12. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}
class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}
```

```

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

13. Quale output si ottiene invocando il metodo q?

```

class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

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

14. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    Plant(String f){
        family = f;
    }
    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {
    public Tree(String f){
        super(f);
    }
    public void stampa() {
        System.out.println("Tree: "
            + family);
    }
    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

15. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }
    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {
    public Tree(String f){
        super(f);
    }
}

```

```

}
public Tree() {}

public void stampa() {
    System.out.println("Tree: "
        + family);
}

public static void main(String[] args) {
    Plant a = new Tree(Olive);
    a.stampa();
}
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

16. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {

```

```

        System.out.print(1);
        m();
        System.out.print(2);
    }
    catch( Exception w ) {
    }
    finally {
        throw( new Exception() );
    }
}
static void m() {
    try {
        System.out.print(3);
        throw( new Exception() );
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. Errore a tempo di compilazione
 - B. 1342
 - C. 134
 - D. 134Exception in thread "main" java.lang.Exception
 - 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. 5

1. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

A. Errore a tempo di compilazione per errata dichiarazione di un array.

B. Viene lanciata una NullPointerException.

C. Viene lanciata una ArrayIndexOutOfBoundsException.

D. null

E. Nessuna delle precedenti.

2. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}
```

```
class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}
```

```
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
```

```
class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

4. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea() {}

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```

public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}

```

- A. 4
- B. 5
- C. 6

D. 7

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 D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}

```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

7. Quale output si ottiene invocando il metodo q?

```

class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {

```

```

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

```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

8. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. s = (B1) y;
- B. s = (B1) g;
- C. s = y;
- D. g = y;
- E. s = g;

9. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

10. Date le dichiarazioni:

```

class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();

```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

11. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
- D. Un'istanza della classe I può essere costruita con new C.I().
- 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 B1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            System.out.print (1);
            m();
            System.out.print (2);
        }
        catch( MyExc3 d ) {
            System.out.print (3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print (4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print (5);
        }
    }
}
```

- A. 14... (ciclo infinito)

B. Errore a tempo di compilazione

C. 143

D. 1432

E. Nessuna delle precedenti

14. Qual'è l'output di questo programma?

```
class Plant {

    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }
    public Tree(){ }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree(Olive);
        a.stampa();
    }
}
```

A. Plant: Olive

B. Tree: Olive

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

15. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print ("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print ("Y");
    }
}
```

```

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

16. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {

```

```

        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- 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. 6

1. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }
    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Date le dichiarazioni:

```
class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Sub1 g;
B1 s;
Super y;
```

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

- A. s = (B1) y;
- B. s = (B1) g;
- C. s = y;
- D. g = y;
- E. s = g;

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

4. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

5. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

6. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    protected Plant(String f) {
        family = f;
    }
}
```

```
    }
    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {
    public Tree(String f) {
        super(f);
    }
    public Tree() {}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args) {
        Plant a = new Tree(Olive);
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
- D. Un'istanza della classe I può essere costruita con new C.I().
- E. Nessuna delle precedenti.

8. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]) {
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

9. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}
```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

10. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}
```

```
class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea() {}

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- 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 B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}
```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

13. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
}
```

```
}

public void print() {
    System.out.print("Geometric figure with ");
    System.out.print(angles + " angles.");
}

public static void main(String[] args) {
    Triangle t = new Triangle("triangle" , 3);
    t.print();
}
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}
```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

15. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();
```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

16. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
```

```
void q() {
    float [] a2 = new float [4];
    m(f1, a2, b3);
}
void m(Float f2, float [] a1, Boolean b2){
    a3 = new float [4];
    if(f1 == f2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a3 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b1 == b2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}
```

- A. 101
 - B. 110
 - C. 100
 - D. 011
 - E. 111
-

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

        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}
```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

2. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.

- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

3. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }
    public Tree(){ }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree(Olive);
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- 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 B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
    }
}
```

```
catch( MyExc3 c ) {
    throw( new MyExc3() );
}
catch( MyExc1 j ) {
    System.out.print(4);
    throw( new MyExc3() );
}
catch( Exception z ) {
    System.out.print(5);
}
}
```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

6. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }
    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

8. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```

public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}

```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

9. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    public static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
- D. Un'istanza della classe I può essere costruita con new C.I().
- E. Nessuna delle precedenti.

10. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    void getArea() {}

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle", 3);
        ((Triangle)t).print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

11. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

12. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

13. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();
```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

14. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

15. Quale output si ottiene invocando il metodo q?

```

class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

A. 101

B. 110

C. 100

D. 011

E. 111

16. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

A. s = (B1) y;

B. s = (B1) g;

C. s = y;

D. g = y;

E. s = g;

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. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

2. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.

E. Nessuna delle precedenti.

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

4. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
    }
}
```

```

    m(f1, a2, b3);
}
void m(Float f2, float [] a1, Boolean b2){
    a3 = new float [4];
    if(f1 == f2) {
        System.out.print (1);
    } else {
        System.out.print (0);
    }
    if(a3 == a1) {
        System.out.print (1);
    } else {
        System.out.print (0);
    }
    if(b1 == b2) {
        System.out.print (1);
    } else {
        System.out.print (0);
    }
}
}
}

```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

5. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

6. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    Plant(String f){
        family = f;
    }
}

```

```

protected void stampa() {
    System.out.println("Plant: "
        + family);
}
}
class Tree extends Plant {
    public Tree(String f){
        super(f);
    }
    public void stampa() {
        System.out.println("Tree: "
            + family);
    }
    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```

public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}

```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

8. Dato un file contenente il seguente codice:

```
package pk;
```

```
public class C {  
    public static class I {  
    }  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
 - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
 - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con `new I()`.
 - D. Un'istanza della classe I può essere costruita con `new C.I()`.
 - E. Nessuna delle precedenti.
-

9. Date le dichiarazioni:

```
class Super extends Object {...}  
class Sub1 extends Super {...}  
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Sub1 g;  
B1 s;  
Super y;
```

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

- A. `s = (B1) y;`
 - B. `s = (B1) g;`
 - C. `s = y;`
 - D. `g = y;`
 - E. `s = g;`
-

10. Date le dichiarazioni:

```
class B1 extends C0 {...}  
class C0 extends Object {...}  
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;  
C0 t;  
B2 z;  
z = new B2();  
t = new B1();  
s = new B1();
```

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

- A. `s = (B1) z;`
 - B. `z = (B2) s;`
 - C. `z = (B2) t;`
 - D. `s = (B1) t;`
 - E. Nessuno dei precedenti
-

11. Qual'è l'output di questo programma?

```
class Plant {  
  
    String family;  
    protected Plant(String f){  
        family = f;  
    }  
  
    protected void stampa() {  
        System.out.println("Plant: "  
            + family);  
    }  
}  
  
class Tree extends Plant {  
  
    public Tree(String f){  
        super(f);  
    }  
    public Tree(){  
  
    }  
  
    public void stampa() {  
        System.out.println("Tree: "  
            + family);  
    }  
  
    public static void main(String[] args){  
        Plant a = new Tree(Olive);  
        a.stampa();  
    }  
}
```

- A. Plant: Olive
 - B. Tree: Olive
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

12. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- 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 B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
    }
}

```

```

}
catch( MyExc1 j ) {
    System.out.print(4);
    throw( new MyExc3() );
}
catch( Exception z ) {
    System.out.print(5);
}
}
}

```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}

```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

15. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
 - B. XYZ
 - C. ZZZ
 - D. Errore a tempo di compilazione.
 - E. Nessuna delle precedenti.
-

16. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea() {}

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle", 3);
        ((Triangle)t).print();
    }
}

```

- A. Geometric figure with 0 angles.
 - B. Geometric figure with 3 angles.
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - 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?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}
```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

2. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
```

```
private Boolean b1 = new Boolean(false);
private Boolean b3 = b1;
void q() {
    float [] a2 = new float [4];
    m(f1, a2, b3);
}
void m(Float f2, float [] a1, Boolean b2){
    a3 = new float [4];
    if(f1 == f2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a3 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b1 == b2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

3. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    Plant(String f){
        family = f;
    }
    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}
```

```

class Tree extends Plant {

    public Tree(String f) {
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args) {
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    void getArea() {}

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle", 3);
        ((Triangle)t).print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.

- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Qual'è l'output di questo programma?

```

class Plant {

    String family;
    protected Plant(String f) {
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f) {
        super(f);
    }

    public Tree() {}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args) {
        Plant a = new Tree(Olive);
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

6. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. s = (B1) y;

- B. s = (B1) g;
- C. s = y;
- D. g = y;
- E. s = g;

7. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}
```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

8. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
    }
}
```

```
Stella.makeStella("Venere"); /* QUI */
Stella.makeStella("Krypton");
}
}
```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

9. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();
```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

10. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
```

```

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

11. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class Bl {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

12. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    public static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
- D. Un'istanza della classe I può essere costruita con new C.I().
- E. Nessuna delle precedenti.

13. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

14. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.**
- B. Viene lanciata una NullPointerException.**
- C. Viene lanciata una ArrayIndexOutOfBoundsException.**
- D. null**
- E. Nessuna delle precedenti.**

15. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
}
```

```
public void print(){
    System.out.print("Geometric figure with ");
    System.out.print(angles + " angles.");
}

public static void main(String[] args) {
    Triangle t = new Triangle("triangle" , 3);
    t.print();
}
}
```

- A. Geometric figure with 0 angles.**
- B. Geometric figure with 3 angles.**
- C. Errore a tempo di compilazione.**
- D. Errore a tempo di esecuzione.**
- E. Nessuna delle precedenti.**

16. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array**
 - B. Una classe non può essere dichiarata final**
 - C. Il minimo numero di elementi che un array può contenere è 0**
 - D. In Java tutti i tipi numerici sono con segno**
 - E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente**
-

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. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }
    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}
class Tree extends Plant {
    public Tree(String f){
        super(f);
    }
    public Tree(){}
    public void stampa() {
        System.out.println("Tree: "
            + family);
    }
    public static void main(String[] args){
        Plant a = new Tree(Olive);
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;
    protected Figure(String t, int a) {
```

```
        type = t;
        angles = a;
    }
    abstract void getArea();
}
class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }
    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```

    }
    if(a3 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b1 == b2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

4. Date le dichiarazioni:

```

class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B1 s;
C0 t;
B2 z;
z = new B2 ();
t = new B1 ();
s = new B1 ();

```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

5. Qual'è l'output di questo codice?

```

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

```

```

}
static void p() throws Exception {
    try {
        throw( new MyExc2() );
    }
    catch( MyExc2 f ) {
        System.out.print(1);
        throw( new MyExc3() );
    }
    catch( Exception d ) {
        System.out.print(2);
    }
    finally {
        System.out.print(3);
        throw( new MyExc2() );
    }
}
}

```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- 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 B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}
}

```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

7. Qual'è l'output di questo programma?

```
class Plant {  
  
    String family;  
    Plant(String f){  
        family = f;  
    }  
  
    protected void stampa() {  
        System.out.println("Plant: "  
            + family);  
    }  
}  
  
class Tree extends Plant {  
  
    public Tree(String f){  
        super(f);  
    }  
  
    public void stampa() {  
        System.out.println("Tree: "  
            + family);  
    }  
  
    public static void main(String[] args){  
        Plant a = new Tree("Olive");  
        a.stampa();  
    }  
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Qual'è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}
```

```
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

9. Qual'è l'output di questo programma?

```
public class C {  
    public static void main(String args[]){  
        Object[] o = new Object[100];  
        System.out.println(o[10]);  
    }  
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

10. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {  
    static String pianetaVicino = new String();  
    String nome = new String();  
  
    static void makeStella(String pianeta) {  
        Stella s = new Stella();  
        s.pianetaVicino = pianeta;  
    }  
  
    public static void main(String [] arg) {  
        Stella.makeStella("Marte");  
    }  
}
```

```

    Stella.makeStella("Venere"); /* QUI */
    Stella.makeStella("Krypton");
}
}

```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

11. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. `s = (B1) y;`
- B. `s = (B1) g;`
- C. `s = y;`
- D. `g = y;`
- E. `s = g;`

12. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
    }
}

```

```

Y y = new Z();
Z z = new Z();
z.f(x); z.f(y); z.f(z);
}
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

13. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

14. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134

D. 134Exception in thread "main" java.lang.Exception

E. Nessuna delle precedenti

15. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea(){

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}
```

}

A. Geometric figure with 0 angles.

B. Geometric figure with 3 angles.

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

16. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. Un'istanza della classe I può essere costruita solo all'interno della classe C.

B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.

C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().

D. Un'istanza della classe I può essere costruita con new C.I().

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 programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

A. Errore a tempo di compilazione per errata dichiarazione di un array.

B. Viene lanciata una NullPointerException.

C. Viene lanciata una ArrayIndexOutOfBoundsException.

D. null

E. Nessuna delle precedenti.

2. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

3. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```

public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}

```

- A. 4
 - B. 5
 - C. 6
 - D. 7
 - E. Nessuna delle precedenti.
-

5. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. s = (B1) y;
 - B. s = (B1) g;
 - C. s = y;
 - D. g = y;
 - E. s = g;
-

6. Date le dichiarazioni:

```

class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();

```

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

- A. s = (B1) z;
 - B. z = (B2) s;
 - C. z = (B2) t;
 - D. s = (B1) t;
 - E. Nessuno dei precedenti
-

7. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    public static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
 - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
 - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
 - D. Un'istanza della classe I può essere costruita con new C.I().
 - E. Nessuna delle precedenti.
-

8. Quale output si ottiene invocando il metodo q?

```

class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        }
    }
}

```

```

    } else {
        System.out.print(0);
    }
}

```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

9. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }
    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- 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 B1 {

```

```

    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}

```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

11. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

12. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {

```

```

        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f) {
        super(f);
    }
    public Tree() {}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args) {
        Plant a = new Tree(Olive);
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

13. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

14. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
    }
}

```

```

    }
    catch( MyExc2 f ) {
        System.out.print(1);
        throw( new MyExc3() );
    }
    catch( Exception d ) {
        System.out.print(2);
    }
    finally {
        System.out.print(3);
        throw( new MyExc2() );
    }
}

```

- A. 12312
 - B. Errore a tempo di compilazione
 - C. 1Exception in thread "main" MyExc3
 - D. 13Exception in thread "main" MyExc2
 - E. Nessuna delle precedenti
-

16. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;

```

```

        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea(){}

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}

```

- A. Geometric figure with 0 angles.
 - B. Geometric figure with 3 angles.
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - 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. 12

1. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea() {}

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
```

```
private void f(Y y) {
    System.out.print("Y");
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

3. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

4. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}
```

```

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

5. Date le dichiarazioni:

```

class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();

```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

6. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}

```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

7. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {
    public Tree(String f){
        super(f);
    }
}

```

```

}
public Tree() {}

public void stampa() {
    System.out.println("Tree: "
        + family);
}

public static void main(String[] args) {
    Plant a = new Tree(Olive);
    a.stampa();
}
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Qual'è l'output di questo programma?

```

class Plant {

    String family;
    Plant(String f) {
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f) {
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args) {
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

10. Quale output si ottiene invocando il metodo q?

```

class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2) {
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

11. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Qual'è l'output di questo codice?

```

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

```

```

        System.out.print(4);
    }
}

```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

13. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. s = (B1) y;
- B. s = (B1) g;
- C. s = y;
- D. g = y;
- E. s = g;

14. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```

public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}

```

- A. 4
- B. 5
- C. 6
- D. 7

E. Nessuna delle precedenti.

15. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
 - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
 - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
 - D. Un'istanza della classe I può essere costruita con new C.I().
 - 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 D1 {
```

```
public static void main(String [] argv)
    throws Exception {
    try {
        p();
    }
}
static void p() throws Exception {
    try {
        throw( new MyExc2() );
    }
    catch( MyExc2 f ) {
        System.out.print(1);
        throw( new MyExc3() );
    }
    catch( Exception d ) {
        System.out.print(2);
    }
    finally {
        System.out.print(3);
        throw( new MyExc2() );
    }
}
}
```

- A. 12312
 - B. Errore a tempo di compilazione
 - C. 1Exception in thread "main" MyExc3
 - D. 13Exception in thread "main" MyExc2
 - 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 programma?

```
class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

4. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}
```

- A. 4
- B. 5
- C. 6
- D. 7
- 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 Bl {
    public static void main(String [] argv)
        throws Exception {

        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}
```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

6. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

7. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

9. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }
    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}
class Tree extends Plant {
    public Tree(String f){
        super(f);
    }
}
```

```
    }
    public Tree() {}
    public void stampa() {
        System.out.println("Tree: "
            + family);
    }
    public static void main(String[] args){
        Plant a = new Tree(Olive);
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

10. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}
class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

11. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea() {}

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    public static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
- D. Un'istanza della classe I può essere costruita con new C.I().
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}

```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

14. Date le dichiarazioni:

```

class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();

```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;

E. Nessuno dei precedenti

15. Date le dichiarazioni:

```
class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Sub1 g;
B1 s;
Super y;
```

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

- A. `s = (B1) y;`
 - B. `s = (B1) g;`
 - C. `s = y;`
 - D. `g = y;`
 - E. `s = g;`
-

16. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}
```

```
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}
```

```
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
```

```
class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
 - B. XYZ
 - C. ZZZ
 - D. Errore a tempo di compilazione.
 - 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. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

2. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
    }
}
```

```
s.pianetaVicino = pianeta;
}

public static void main(String [] arg) {
    Stella.makeStella("Marte");
    Stella.makeStella("Venere"); /* QUI */
    Stella.makeStella("Krypton");
}
}
```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

3. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.

- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

- D. null
- E. Nessuna delle precedenti.

4. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea() {}

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.

6. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

7. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();
```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

8. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
```

```

protected void f(Z z) {
    System.out.print("Z");
}
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}

```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

10. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. s = (B1) y;
- B. s = (B1) g;
- C. s = y;
- D. g = y;
- E. s = g;

11. Qual'è l'output di questo programma?

```

class Plant {

    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

- A. Plant: Olive
 - B. Tree: Olive
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

12. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.**
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.**
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().**
- D. Un'istanza della classe I può essere costruita con new C.I().**
- E. Nessuna delle precedenti.**

13. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }
    public Tree(){}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree(Olive);
        a.stampa();
    }
}
```

- A. Plant: Olive**
- B. Tree: Olive**
- C. Errore a tempo di compilazione.**

D. Errore a tempo di esecuzione.

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 B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}
```

- A. 14... (ciclo infinito)**
- B. Errore a tempo di compilazione**
- C. 143**
- D. 1432**
- E. Nessuna delle precedenti**

15. Qual'è l'output di questo codice?

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

```

        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
 - B. 1342
 - C. 134
 - D. 134Exception in thread "main" java.lang.Exception
 - E. Nessuna delle precedenti
-

16. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

```

```

    }
class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
 - B. XYZ
 - C. ZZZ
 - D. Errore a tempo di compilazione.
 - 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. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

```
B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();
```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
```

2. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```

        System.out.print(5);
    }
}

```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

4. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una `NullPointerException`.
- C. Viene lanciata una `ArrayIndexOutOfBoundsException`.
- D. `null`
- E. Nessuna delle precedenti.

5. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

6. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    public static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe `I` può essere costruita solo all'interno della classe `C`.
- B. Un'istanza della classe `I` può essere costruita solo all'interno del pacchetto `pk`.
- C. All'interno del pacchetto `pk`, un'istanza della classe `I` può essere costruita con `new I()`.
- D. Un'istanza della classe `I` può essere costruita con `new C.I()`.
- E. Nessuna delle precedenti.

7. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```

public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}

```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

8. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
 - B. XYZ
 - C. ZZZ
 - D. Errore a tempo di compilazione.
 - E. Nessuna delle precedenti.
-

9. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea(){}

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }
}
```

```
}

public static void main(String[] args) {
    Figure t = new Triangle("triangle" , 3);
    ((Triangle)t).print();
}
}
```

- A. Geometric figure with 0 angles.
 - B. Geometric figure with 3 angles.
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

10. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {
    public Tree(String f){
        super(f);
    }
    public Tree(){}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree(Olive);
        a.stampa();
    }
}
```

- A. Plant: Olive
 - B. Tree: Olive
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

11. Qual'è l'output di questo programma?

```

class Plant {

    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. s = (B1) y;
- B. s = (B1) g;
- C. s = y;
- D. g = y;
- E. s = g;

13. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

15. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
 - B. 1342
 - C. 134
 - D. 134Exception in thread "main" java.lang.Exception
 - 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 D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}

```

- A. 12312
 - B. Errore a tempo di compilazione
 - C. 1Exception in thread "main" MyExc3
 - D. 13Exception in thread "main" MyExc2
 - 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. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

2. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
- D. Un'istanza della classe I può essere costruita con new C.I().
- E. Nessuna delle precedenti.

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

E. Nessuna delle precedenti.

4. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }
    public Tree(){}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree(Olive);
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null

6. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
}
```

```

}
void getArea() {}

public void print() {
    System.out.print("Geometric figure with ");
    System.out.print(angles + " angles.");
}

public static void main(String[] args) {
    Figure t = new Triangle("triangle" , 3);
    ((Triangle)t).print();
}
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```

public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}

```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

9. Date le dichiarazioni:

```

class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();

```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

10. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```
Sub1 g;  
Bl s;  
Super y;
```

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

- A. `s = (Bl) y;`
- B. `s = (Bl) g;`
- C. `s = y;`
- D. `g = y;`
- E. `s = g;`

12. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

13. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

A. XXX

- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- 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 D1 {  
    public static void main(String [] argv)  
        throws Exception {  
  
        try {  
            p();  
        }  
    }  
    static void p() throws Exception {  
        try {  
            throw( new MyExc2() );  
        }  
        catch( MyExc2 f ) {  
            System.out.print(1);  
            throw( new MyExc3() );  
        }  
        catch( Exception d ) {  
            System.out.print(2);  
        }  
        finally {  
            System.out.print(3);  
            throw( new MyExc2() );  
        }  
    }  
}
```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- 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 B1 {  
    public static void main(String [] argv)  
        throws Exception {  
  
        try {  
            System.out.print(1);  
            m();  
            System.out.print(2);  
        }  
        catch( MyExc3 d ) {  
            System.out.print(3);  
        }  
    }  
}
```

```

}
static void m() throws Exception {
    try {
        if(true) throw( new MyExc1() );
        else throw( new MyExc3() );
    }
    catch( MyExc3 c ) {
        throw( new MyExc3() );
    }
    catch( MyExc1 j ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( Exception z ) {
        System.out.print(5);
    }
}
}

```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

16. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

```

```

}
class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- 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 MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

2. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }
}
```

```
abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea(){}

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}
```

- A. 4
- B. 5

C. 6

D. 7

E. Nessuna delle precedenti.

4. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

A. XXX

B. XYZ

C. ZZZ

D. Errore a tempo di compilazione.

E. Nessuna delle precedenti.

5. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]) {
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

A. Errore a tempo di compilazione per errata dichiarazione di un array.

B. Viene lanciata una `NullPointerException`.

C. Viene lanciata una `ArrayIndexOutOfBoundsException`.

D. null

E. Nessuna delle precedenti.

6. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    protected Plant(String f) {
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {
    public Tree(String f) {
        super(f);
    }
    public Tree() {}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args) {
        Plant a = new Tree(Olive);
        a.stampa();
    }
}
```

A. Plant: Olive

B. Tree: Olive

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

7. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. Un'istanza della classe `I` può essere costruita solo all'interno della classe `C`.

B. Un'istanza della classe `I` può essere costruita solo all'interno del pacchetto `pk`.

C. All'interno del pacchetto `pk`, un'istanza della classe `I` può essere costruita con `new I()`.

D. Un'istanza della classe I può essere costruita con new C.I().

E. Nessuna delle precedenti.

8. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

A. 101

B. 110

C. 100

D. 011

E. 111

9. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
```

```
protected void f(Z z) {
    System.out.print("Z");
}
}
```

```
class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

A. XXX

B. XYZ

C. ZZZ

D. Errore a tempo di compilazione.

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 B1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}
```

A. 14... (ciclo infinito)

B. Errore a tempo di compilazione

C. 143

D. 1432

E. Nessuna delle precedenti

11. Date le dichiarazioni:

```
class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Sub1 g;
B1 s;
Super y;
```

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

- A. `s = (B1) y;`
- B. `s = (B1) g;`
- C. `s = y;`
- D. `g = y;`
- E. `s = g;`

12. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

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 D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}
```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

14. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();
```

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

- A. `s = (B1) z;`

- B. z = (B2) s;
 - C. z = (B2) t;
 - D. s = (B1) t;
 - E. Nessuno dei precedenti
-

15. Qual'è l'output di questo programma?

```
class Plant {  
  
    String family;  
    Plant(String f) {  
        family = f;  
    }  
  
    protected void stampa() {  
        System.out.println("Plant: "  
            + family);  
    }  
}  
  
class Tree extends Plant {  
  
    public Tree(String f) {  
        super(f);  
    }  
  
    public void stampa() {  
        System.out.println("Tree: "  
            + family);  
    }  
}
```

```
}  
public static void main(String[] args) {  
    Plant a = new Tree("Olive");  
    a.stampa();  
}  
}
```

- A. Plant: Olive
 - B. Tree: Olive
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

16. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
 - B. Una classe non può essere dichiarata final
 - C. Il minimo numero di elementi che un array può contenere è 0
 - D. In Java tutti i tipi numerici sono con segno
 - E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente
-

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. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}
```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

2. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();
```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;

D. s = (B1) t;

E. Nessuno dei precedenti

3. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}
```

```
class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}
```

```
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
```

```
class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

4. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

5. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

6. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }
}
```

```
}

abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea(){}

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            p();
        }
    }
}
```

```
static void p() throws Exception {
    try {
        throw( new MyExc2() );
    }
    catch( MyExc2 f ) {
        System.out.print(1);
        throw( new MyExc3() );
    }
    catch( Exception d ) {
        System.out.print(2);
    }
    finally {
        System.out.print(3);
        throw( new MyExc2() );
    }
}
```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

10. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

11. Qual'è l'output di questo programma?

```
class Plant {

    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }

    public Tree(){}
}
```

```

public void stampa() {
    System.out.println("Tree: "
        + family);
}

public static void main(String[] args) {
    Plant a = new Tree(Olive);
    a.stampa();
}
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

13. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    public static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
- D. Un'istanza della classe I può essere costruita con new C.I().
- E. Nessuna delle precedenti.

14. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- 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 B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}

```

- A. 14... (ciclo infinito)
 - B. Errore a tempo di compilazione
 - C. 143
 - D. 1432
 - E. Nessuna delle precedenti
-

16. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. s = (B1) y;
 - B. s = (B1) g;
 - C. s = y;
 - D. g = y;
 - E. s = g;
-

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. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

2. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    protected Plant(String f) {
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
```

```
        + family);
    }
}

class Tree extends Plant {
    public Tree(String f) {
        super(f);
    }
    public Tree() {}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args) {
        Plant a = new Tree(Olive);
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    Plant(String f) {
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {
    public Tree(String f) {
```

```

    super(f);
}

public void stampa() {
    System.out.println("Tree: "
        + family);
}

public static void main(String[] args) {
    Plant a = new Tree("Olive");
    a.stampa();
}
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. `s = (B1) y;`
- B. `s = (B1) g;`
- C. `s = y;`
- D. `g = y;`
- E. `s = g;`

5. Qual'è l'output di questo codice?

```

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

```

```

        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}

```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

6. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una `NullPointerException`.
- C. Viene lanciata una `ArrayIndexOutOfBoundsException`.
- D. null
- E. Nessuna delle precedenti.

7. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
}

```

```

public void print() {
    System.out.print("Geometric figure with ");
    System.out.print(angles + " angles.");
}

public static void main(String[] args) {
    Triangle t = new Triangle("triangle" , 3);
    t.print();
}
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    void getArea() {}

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

10. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    public static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con `new I()`.
- D. Un'istanza della classe I può essere costruita con `new C.I()`.
- E. Nessuna delle precedenti.

11. Date le dichiarazioni:

```

class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B1 s;
C0 t;
B2 z;
z = new B2 ();
t = new B1 ();
s = new B1 ();

```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

12. Quale output si ottiene invocando il metodo q?

```

class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

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

E. 111

13. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```

public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}

```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

14. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

15. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
}

```

```

}
static void m() throws Exception {
    try {
        if(true) throw( new MyExc1() );
        else throw( new MyExc3() );
    }
    catch( MyExc3 c ) {
        throw( new MyExc3() );
    }
    catch( MyExc1 j ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( Exception z ) {
        System.out.print(5);
    }
}
}

```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

16. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }

```

```

public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
 - B. 1342
 - C. 134
 - D. 134Exception in thread "main" java.lang.Exception
 - 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. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2 ();
t = new B1 ();
s = new B1 ();
```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

2. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea(){}

    public void print(){
        System.out.print("Geometric figure with ");
```

```
System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}
```

- A. 12312

- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

4. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

5. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}
```

```
class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}
```

```
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
```

```
class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ

6. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}
```

```
class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}
```

```
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
```

```
class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

7. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}
```

```
class Tree extends Plant {
    public Tree(String f){
        super(f);
    }
}
```

```

public void stampa() {
    System.out.println("Tree: "
        + family);
}
public static void main(String[] args) {
    Plant a = new Tree("Olive");
    a.stampa();
}
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

9. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. s = (B1) y;
- B. s = (B1) g;
- C. s = y;
- D. g = y;
- E. s = g;

10. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}

```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

11. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    public static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
- D. Un'istanza della classe I può essere costruita con new C.I().
- E. Nessuna delle precedenti.

12. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

13. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}
```

A. 4

B. 5

C. 6

D. 7

E. Nessuna delle precedenti.

14. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

A. 101

B. 110

C. 100

D. 011

E. 111

15. Qual'è l'output di questo codice?

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

```

        throw( new Exception() );
    }
}
static void m() {
    try {
        System.out.print(3);
        throw( new Exception() );
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. Errore a tempo di compilazione
 - B. 1342
 - C. 134
 - D. 134Exception in thread "main" java.lang.Exception
 - E. Nessuna delle precedenti
-

16. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }
}

```

```

protected void stampa() {
    System.out.println("Plant: "
        + family);
}
}
class Tree extends Plant {
    public Tree(String f){
        super(f);
    }
    public Tree(){}
    public void stampa() {
        System.out.println("Tree: "
            + family);
    }
    public static void main(String[] args){
        Plant a = new Tree(Olive);
        a.stampa();
    }
}

```

- A. Plant: Olive
 - B. Tree: Olive
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - 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. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}
```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

2. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {
    public Tree(String f){
        super(f);
    }
    public Tree() {}
}
```

```
public void stampa() {
    System.out.println("Tree: "
        + family);
}

public static void main(String[] args){
    Plant a = new Tree(Olive);
    a.stampa();
}
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```
}  
}
```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv){  
        try {  
            System.out.print(1);  
            m();  
            System.out.print(2);  
        }  
        catch( Exception w ) { }  
        finally {  
            throw( new Exception() );  
        }  
    }  
    static void m() {  
        try {  
            System.out.print(3);  
            throw( new Exception() );  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

5. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {
```

```
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- 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 D1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            p();  
        }  
    }  
    static void p() throws Exception {  
        try {  
            throw( new MyExc2() );  
        }  
        catch( MyExc2 f ) {  
            System.out.print(1);  
            throw( new MyExc3() );  
        }  
        catch( Exception d ) {  
            System.out.print(2);  
        }  
        finally {  
            System.out.print(3);  
            throw( new MyExc2() );  
        }  
    }  
}
```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2

E. Nessuna delle precedenti

7. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

8. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}
```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

9. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

10. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print() {
        System.out.print("Geometric figure with ");
    }
}
```

```

    System.out.print(angles + " angles.");
}

public static void main(String[] args) {
    Triangle t = new Triangle("triangle" , 3);
    t.print();
}
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

12. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];

```

```

        System.out.println(o[10]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una `NullPointerException`.
- C. Viene lanciata una `ArrayIndexOutOfBoundsException`.
- D. null
- E. Nessuna delle precedenti.

13. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    public static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe `I` può essere costruita solo all'interno della classe `C`.
- B. Un'istanza della classe `I` può essere costruita solo all'interno del pacchetto `pk`.
- C. All'interno del pacchetto `pk`, un'istanza della classe `I` può essere costruita con `new I()`.
- D. Un'istanza della classe `I` può essere costruita con `new C.I()`.
- E. Nessuna delle precedenti.

14. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea(){

    }

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }
}

```

```
public static void main(String[] args) {
    Figure t = new Triangle("triangle" , 3);
    ((Triangle)t).print();
}
}
```

- A. Geometric figure with 0 angles.
 - B. Geometric figure with 3 angles.
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

15. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();
```

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

- A. s = (B1) z;
 - B. z = (B2) s;
 - C. z = (B2) t;
 - D. s = (B1) t;
 - E. Nessuno dei precedenti
-

16. Date le dichiarazioni:

```
class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Sub1 g;
B1 s;
Super y;
```

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

- A. s = (B1) y;
 - B. s = (B1) g;
 - C. s = y;
 - D. g = y;
 - E. s = g;
-

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 q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
```

```
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}
```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

3. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();
```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

4. Date le dichiarazioni:

```
class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Sub1 g;
B1 s;
Super y;
```

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

- A. s = (B1) y;
- B. s = (B1) g;
- C. s = y;
- D. g = y;
- E. s = g;

5. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

6. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea(){}

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
    static void p() throws Exception {
```

```

try {
    throw( new MyExc2() );
}
catch( MyExc2 f ) {
    System.out.print(1);
    throw( new MyExc3() );
}
catch( Exception d ) {
    System.out.print(2);
}
finally {
    System.out.print(3);
    throw( new MyExc2() );
}
}
}

```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

8. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.

- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

10. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }
}

```

```

public Tree() {}

public void stampa() {
    System.out.println("Tree: "
        + family);
}

public static void main(String[] args) {
    Plant a = new Tree(Olive);
    a.stampa();
}
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```

public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}

```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

12. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.

- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

13. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

14. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    public static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.

- C. All'interno del pacchetto `pk`, un'istanza della classe `I` può essere costruita con `new I()`.
 - D. Un'istanza della classe `I` può essere costruita con `new C.I()`.
 - E. Nessuna delle precedenti.
-

15. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }
}
```

```
public static void main(String[] args) {
    Triangle t = new Triangle("triangle" , 3);
    t.print();
}
}
```

- A. Geometric figure with 0 angles.
 - B. Geometric figure with 3 angles.
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

16. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
 - B. Una classe non può essere dichiarata `final`
 - C. Il minimo numero di elementi che un array può contenere è 0
 - D. In Java tutti i tipi numerici sono con segno
 - E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente
-

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 MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

2. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }
}
```

```
abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea(){}

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();
```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;

- D. `s = (B1) t;`
- E. Nessuno dei precedenti

4. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

5. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}
```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

6. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

7. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
```

```

public Triangle(String t, int a) {
    super(t, a);
}

public void print() {
    System.out.print("Geometric figure with ");
    System.out.print(angles + " angles.");
}

public static void main(String[] args) {
    Triangle t = new Triangle("triangle", 3);
    t.print();
}
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    public static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con `new I()`.
- D. Un'istanza della classe I può essere costruita con `new C.I()`.
- E. Nessuna delle precedenti.

9. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

10. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una `NullPointerException`.
- C. Viene lanciata una `ArrayIndexOutOfBoundsException`.
- D. null
- E. Nessuna delle precedenti.

11. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. `s = (B1) y;`
- B. `s = (B1) g;`
- C. `s = y;`
- D. `g = y;`
- E. `s = g;`

12. Quale output si ottiene invocando il metodo q?

```

class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```

}
if(a3 == a1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(b1 == b2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
}

```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

13. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}

```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione

- C. 143
- D. 1432
- E. Nessuna delle precedenti

14. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {

```

```

        System.out.print(1);
        throw( new MyExc3() );
    }
    catch( Exception d ) {
        System.out.print(2);
    }
    finally {
        System.out.print(3);
        throw( new MyExc2() );
    }
}
}

```

- A. 12312
 - B. Errore a tempo di compilazione
 - C. 1Exception in thread "main" MyExc3
 - D. 13Exception in thread "main" MyExc2
 - E. Nessuna delle precedenti
-

16. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {

```

```

        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }
    public Tree(){}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree(Olive);
        a.stampa();
    }
}

```

- A. Plant: Olive
 - B. Tree: Olive
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - 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. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

2. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }
}
```

```
}
abstract void getArea();
}
class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea(){}

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

4. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {
    public Tree(String f){
        super(f);
    }

    public void stampa() {
```

```
        System.out.println("Tree: "
            + family);
    }
    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

6. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}
```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

7. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

8. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2 ();
t = new B1 ();
s = new B1 ();
```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

9. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
- D. Un'istanza della classe I può essere costruita con new C.I().
- E. Nessuna delle precedenti.

10. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}
```

```
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }
    public Tree(){}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree(Olive);
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Date le dichiarazioni:

```
class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Sub1 g;
B1 s;
Super y;
```

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

- A. s = (B1) y;
- B. s = (B1) g;
- C. s = y;
- D. g = y;
- E. s = g;

12. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
        }
    }
}

```

```

        throw( new MyExc2() );
    }
}

```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

14. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

15. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {

```

```

        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- 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 B1 {
    public static void main(String [] argv)
        throws Exception {

```

```

        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}

```

- A. 14... (ciclo infinito)
 - B. Errore a tempo di compilazione
 - C. 143
 - D. 1432
 - 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. 25

1. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

2. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
```

```
        + family);
    }
}

class Tree extends Plant {
    public Tree(String f){
        super(f);
    }
    public Tree(){}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree(Olive);
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con `new I()`.
- D. Un'istanza della classe I può essere costruita con `new C.I()`.

E. Nessuna delle precedenti.

4. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

5. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print (1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print (2);
        }
    }
}
```

```
}
finally {
    System.out.print (3);
    throw( new MyExc2() );
}
}
```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

7. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2 ();
t = new B1 ();
s = new B1 ();
```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

8. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print () {
```

```

    System.out.print("Geometric figure with ");
    System.out.print(angles + " angles.");
}

public static void main(String[] args) {
    Triangle t = new Triangle("triangle" , 3);
    t.print();
}
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

```

```

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

10. Quale output si ottiene invocando il metodo q?

```

class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

11. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "

```

```

        + family);
    }
    public static void main(String[] args) {
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

13. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;
}

```

```

protected Figure(String t, int a) {
    type = t;
    angles = a;
}

abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea() {}

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- 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 B1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {

```

```

        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( Exception z ) {
        System.out.print(5);
    }
}
}

```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

15. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. s = (B1) y;
- B. s = (B1) g;

- C. s = y;
- D. g = y;
- E. s = g;

16. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```

public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}

```

- A. 4
- B. 5
- C. 6
- D. 7
- 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. 26

1. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
```

```
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

3. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
```

```

public static void main(String[] args) {
    X x = new Z();
    Y y = new Z();
    Z z = new Z();
    z.f(x); z.f(y); z.f(z);
}
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

4. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    public static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con `new I()`.
- D. Un'istanza della classe I può essere costruita con `new C.I()`.
- E. Nessuna delle precedenti.

5. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

6. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una `NullPointerException`.
- C. Viene lanciata una `ArrayIndexOutOfBoundsException`.
- D. null
- E. Nessuna delle precedenti.

7. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```

public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}

```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

8. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. `s = (B1) y;`
- B. `s = (B1) g;`
- C. `s = y;`
- D. `g = y;`
- E. `s = g;`

9. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }
    public Tree(){}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree(Olive);
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

10. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
}
```

```
}
void getArea(){}

public void print(){
    System.out.print("Geometric figure with ");
    System.out.print(angles + " angles.");
}

public static void main(String[] args) {
    Figure t = new Triangle("triangle" , 3);
    ((Triangle)t).print();
}
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}
}
```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

12. Date le dichiarazioni:

```

class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B1 s;
C0 t;
B2 z;
z = new B2 ();
t = new B1 ();
s = new B1 ();

```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- 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 B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}

```

- A. 14... (ciclo infinito)

B. Errore a tempo di compilazione

C. 143

D. 1432

E. Nessuna delle precedenti

14. Quale output si ottiene invocando il metodo q?

```

class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

A. 101

B. 110

C. 100

D. 011

E. 111

15. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    Plant(String f){
        family = f;
    }
    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

```

```

class Tree extends Plant {
    public Tree(String f) {
        super(f);
    }
    public void stampa() {
        System.out.println("Tree: "
            + family);
    }
    public static void main(String[] args) {
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- 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 D1 {
    public static void main(String [] argv)

```

```

        throws Exception {
            try {
                p();
            }
        }
        static void p() throws Exception {
            try {
                throw( new MyExc2() );
            }
            catch( MyExc2 f ) {
                System.out.print(1);
                throw( new MyExc3() );
            }
            catch( Exception d ) {
                System.out.print(2);
            }
            finally {
                System.out.print(3);
                throw( new MyExc2() );
            }
        }
    }
}

```

- A. 12312
 - B. Errore a tempo di compilazione
 - C. 1Exception in thread "main" MyExc3
 - D. 13Exception in thread "main" MyExc2
 - 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 programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

2. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
}
```

```
void m(Float f2, float [] a1, Boolean b2){
    a3 = new float [4];
    if(f1 == f2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a3 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b1 == b2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

3. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}
```

- A. 4

- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

4. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2 ();
t = new B1 ();
s = new B1 ();
```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

5. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}
```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

7. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
- D. Un'istanza della classe I può essere costruita con new C.I().
- E. Nessuna delle precedenti.

8. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea(){}

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- 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 B1 {
    public static void main(String [] argv)
        throws Exception {
```

```
try {
    System.out.print(1);
    m();
    System.out.print(2);
}
catch( MyExc3 d ) {
    System.out.print(3);
}
}
static void m() throws Exception {
    try {
        if(true) throw( new MyExc1() );
        else throw( new MyExc3() );
    }
    catch( MyExc3 c ) {
        throw( new MyExc3() );
    }
    catch( MyExc1 j ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( Exception z ) {
        System.out.print(5);
    }
}
}
```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

10. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
```

```

    Triangle t = new Triangle("triangle" , 3);
    t.print();
}
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }
    public Tree(){}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree(Olive);
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

```

```

}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

```

```

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

```

```

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

13. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

14. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.

- C. Viene lanciata una `ArrayIndexOutOfBoundsException`.
 - D. null
 - E. Nessuna delle precedenti.
-

15. Date le dichiarazioni:

```
class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Sub1 g;
B1 s;
Super y;
```

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

- A. `s = (B1) y;`
 - B. `s = (B1) g;`
 - C. `s = y;`
 - D. `g = y;`
 - E. `s = g;`
-

16. Qual'è l'output di questo programma?

```
class Plant {
    String family;
```

```
Plant(String f) {
    family = f;
}

protected void stampa() {
    System.out.println("Plant: "
        + family);
}

class Tree extends Plant {

    public Tree(String f) {
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args) {
        Plant a = new Tree("Olive");
        a.stampa();
    }
}
```

- A. Plant: Olive
 - B. Tree: Olive
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - 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. 28

1. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }
    public Tree() {}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree(Olive);
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();
```

```
static void makeStella(String pianeta) {
    Stella s = new Stella();
    s.pianetaVicino = pianeta;
}

public static void main(String [] arg) {
    Stella.makeStella("Marte");
    Stella.makeStella("Venere"); /* QUI */
    Stella.makeStella("Krypton");
}
}
```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

3. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();
```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

4. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
    }
}

```

```

Z z = (Z)y;
z.f(x); z.f(y); z.f(z);
}
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}

```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

7. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno

E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

8. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
 - B. Viene lanciata una `NullPointerException`.
 - C. Viene lanciata una `ArrayIndexOutOfBoundsException`.
 - D. null
 - E. Nessuna delle precedenti.
-

9. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe `I` può essere costruita solo all'interno della classe `C`.
 - B. Un'istanza della classe `I` può essere costruita solo all'interno del pacchetto `pk`.
 - C. All'interno del pacchetto `pk`, un'istanza della classe `I` può essere costruita con `new I()`.
 - D. Un'istanza della classe `I` può essere costruita con `new C.I()`.
 - E. Nessuna delle precedenti.
-

10. Date le dichiarazioni:

```
class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Sub1 g;
B1 s;
Super y;
```

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

- A. `s = (B1) y;`
 - B. `s = (B1) g;`
 - C. `s = y;`
 - D. `g = y;`
 - E. `s = g;`
-

11. Qual'è l'output di questo programma?

```
class Plant {

    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}
```

- A. Plant: Olive
 - B. Tree: Olive
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - 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 B1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            System.out.print(1);
            m();
        }
    }
}
```

```

        System.out.print(2);
    }
    catch( MyExc3 d ) {
        System.out.print(3);
    }
}
static void m() throws Exception {
    try {
        if(true) throw( new MyExc1() );
        else throw( new MyExc3() );
    }
    catch( MyExc3 c ) {
        throw( new MyExc3() );
    }
    catch( MyExc1 j ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( Exception z ) {
        System.out.print(5);
    }
}
}

```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

13. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

14. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea(){}

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

15. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

```

```

}
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

16. Quale output si ottiene invocando il metodo q?

```

class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
}

```

```

void q() {
    float [] a2 = new float [4];
    m(f1, a2, b3);
}
void m(Float f2, float [] a1, Boolean b2){
    a3 = new float [4];
    if(f1 == f2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a3 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b1 == b2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 101
 - B. 110
 - C. 100
 - D. 011
 - E. 111
-

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. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

2. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}
```

```

}
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

3. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2) {
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```

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

```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

4. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}

```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

5. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```

public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] argv) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}

```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

6. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. s = (B1) y;
- B. s = (B1) g;
- C. s = y;
- D. g = y;
- E. s = g;

7. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }
}

```

```

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

9. Date le dichiarazioni:

```

class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();

```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

10. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

11. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno

E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

12. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
- D. Un'istanza della classe I può essere costruita con new C.I().
- E. Nessuna delle precedenti.

13. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    Plant(String f) {
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f) {
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args) {
        Plant a = new Tree("Olive");
        a.stampa();
    }
}
```

A. Plant: Olive

B. Tree: Olive

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

14. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    void getArea() {}

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle", 3);
        ((Triangle)t).print();
    }
}
```

A. Geometric figure with 0 angles.

B. Geometric figure with 3 angles.

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

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 D1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            p();
        }
    }

    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
    }
}
```

```

    }
    catch( MyExc2 f ) {
        System.out.print(1);
        throw( new MyExc3() );
    }
    catch( Exception d ) {
        System.out.print(2);
    }
    finally {
        System.out.print(3);
        throw( new MyExc2() );
    }
}
}

```

- A. 12312
 - B. Errore a tempo di compilazione
 - C. 1Exception in thread "main" MyExc3
 - D. 13Exception in thread "main" MyExc2
 - E. Nessuna delle precedenti
-

16. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }
}

```

```

protected void stampa() {
    System.out.println("Plant: "
        + family);
}
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }
    public Tree(){
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree(Olive);
        a.stampa();
    }
}

```

- A. Plant: Olive
 - B. Tree: Olive
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - 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. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea(){}

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- 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 B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
```

```
System.out.print(1);
m();
System.out.print(2);
}
catch( MyExc3 d ) {
    System.out.print(3);
}
}
static void m() throws Exception {
    try {
        if(true) throw( new MyExc1() );
        else throw( new MyExc3() );
    }
    catch( MyExc3 c ) {
        throw( new MyExc3() );
    }
    catch( MyExc1 j ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( Exception z ) {
        System.out.print(5);
    }
}
}
```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

3. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}
```

```

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}

```

- A. Geometric figure with 0 angles.
 B. Geometric figure with 3 angles.
 C. Errore a tempo di compilazione.
 D. Errore a tempo di esecuzione.
 E. Nessuna delle precedenti.

4. Quale output si ottiene invocando il metodo q?

```

class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

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

5. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. `s = (B1) y;`
 B. `s = (B1) g;`
 C. `s = y;`
 D. `g = y;`
 E. `s = g;`

6. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    public static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
 B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
 C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con `new I()`.
 D. Un'istanza della classe I può essere costruita con `new C.I()`.
 E. Nessuna delle precedenti.

7. Qual'è l'output di questo programma?

```

class Plant {

    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

```

```

    }
}

class Tree extends Plant {

    public Tree(String f) {
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args) {
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134

D. 134Exception in thread "main" java.lang.Exception

E. Nessuna delle precedenti

9. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

10. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

11. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```

public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}

```

- A. 4
- B. 5
- C. 6
- D. 7
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}

```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

13. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

14. Date le dichiarazioni:

```

class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();

```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;

E. Nessuno dei precedenti

15. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }
    public Tree(){}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree(Olive);
```

```
        a.stampa();
    }
}
```

- A. Plant: Olive
 - B. Tree: Olive
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

16. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
 - B. Viene lanciata una NullPointerException.
 - C. Viene lanciata una ArrayIndexOutOfBoundsException.
 - D. null
 - 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 programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
```

```
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- 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 Bl {
    public static void main(String [] argv)
        throws Exception {

        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }

    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
```

```

    }
    catch( MyExc3 c ) {
        throw( new MyExc3() );
    }
    catch( MyExc1 j ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( Exception z ) {
        System.out.print(5);
    }
}
}

```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

4. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```

public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}

```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

5. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.

- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}

```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

7. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. `s = (B1) y;`
- B. `s = (B1) g;`
- C. `s = y;`
- D. `g = y;`
- E. `s = g;`

- A. **Plant: Olive**
- B. **Tree: Olive**
- C. **Errore a tempo di compilazione.**
- D. **Errore a tempo di esecuzione.**
- E. **Nessuna delle precedenti.**

8. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe `I` può essere costruita solo all'interno della classe `C`.
- B. Un'istanza della classe `I` può essere costruita solo all'interno del pacchetto `pk`.
- C. All'interno del pacchetto `pk`, un'istanza della classe `I` può essere costruita con `new I()`.
- D. Un'istanza della classe `I` può essere costruita con `new C.I()`.
- E. Nessuna delle precedenti.

9. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}
```

10. Quale output si ottiene invocando il metodo `q`?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

11. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}
```

```

}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea() {}

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle", 3);
        ((Triangle)t).print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134

D. 134Exception in thread "main" java.lang.Exception

E. Nessuna delle precedenti

13. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }
    public Tree(){}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree(Olive);
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

15. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2 ();
t = new B1 ();
s = new B1 ();
```

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

- A. `s = (B1) z;`
 - B. `z = (B2) s;`
 - C. `z = (B2) t;`
 - D. `s = (B1) t;`
 - E. Nessuno dei precedenti
-

16. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}
```

```
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z ();
        Y y = new Z ();
        Z z = new Z ();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
 - B. XYZ
 - C. ZZZ
 - D. Errore a tempo di compilazione.
 - 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 programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

2. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.

- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

3. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea(){

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {
    public Tree(String f){
        super(f);
    }
    public Tree(){}

    public void stampa() {
        System.out.println("Tree: "

```

```

        + family);
    }

    public static void main(String[] args){
        Plant a = new Tree(Olive);
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

6. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

7. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    public static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
- D. Un'istanza della classe I può essere costruita con new C.I().
- E. Nessuna delle precedenti.

8. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

9. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```

public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}

```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

10. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {

```

```

        System.out.print (2);
    }
    finally {
        System.out.print (3);
        throw( new MyExc2() );
    }
}
}

```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

12. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. s = (B1) y;
- B. s = (B1) g;
- C. s = y;
- D. g = y;
- E. s = g;

13. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    Plant(String f){
        family = f;
    }
    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}
class Tree extends Plant {
    public Tree(String f){
        super(f);
    }
}

```

```

}
public void stampa() {
    System.out.println("Tree: "
        + family);
}
public static void main(String[] args){
    Plant a = new Tree("Olive");
    a.stampa();
}
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- 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 B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print (1);
            m();
            System.out.print (2);
        }
        catch( MyExc3 d ) {
            System.out.print (3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print (4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print (5);
        }
    }
}

```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432

E. Nessuna delle precedenti

15. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();
```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;

E. Nessuno dei precedenti

16. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
```

```
private Boolean b1 = new Boolean(false);
private Boolean b3 = b1;
void q() {
    float [] a2 = new float [4];
    m(f1, a2, b3);
}
void m(Float f2, float [] a1, Boolean b2){
    a3 = new float [4];
    if(f1 == f2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a3 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b1 == b2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}
```

- A. 101
 - B. 110
 - C. 100
 - D. 011
 - E. 111
-

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?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print (1);
            m();
            System.out.print (2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print (3);
            throw( new Exception() );
        }
        finally {
            System.out.print (4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

2. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    Plant (String f){
        family = f;
    }
    protected void stampa() {
```

```
        System.out.println("Plant: "
            + family);
    }
}
class Tree extends Plant {
    public Tree(String f){
        super(f);
    }
    public void stampa() {
        System.out.println("Tree: "
            + family);
    }
    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
```

```

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

```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

4. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

5. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

6. Qual'è l'output di questo programma?

```

class Plant {

    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }
    public Tree(){}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree(Olive);
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {

```

```

void f(Y y) {
    System.out.print("Y");
}
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

8. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea() {}

    public void print() {
        System.out.print("Geometric figure with " + angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle", 3);
        ((Triangle)t).print();
    }
}

```

- A. Geometric figure with 0 angles.

- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    public static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con `new I()`.
- D. Un'istanza della classe I può essere costruita con `new C.I()`.
- E. Nessuna delle precedenti.

10. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una `NullPointerException`.
- C. Viene lanciata una `ArrayIndexOutOfBoundsException`.
- D. null
- 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 B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
    }
}

```

```

        System.out.print(2);
    }
    catch( MyExc3 d ) {
        System.out.print(3);
    }
}
static void m() throws Exception {
    try {
        if(true) throw( new MyExc1() );
        else throw( new MyExc3() );
    }
    catch( MyExc3 c ) {
        throw( new MyExc3() );
    }
    catch( MyExc1 j ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( Exception z ) {
        System.out.print(5);
    }
}
}

```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

12. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. s = (B1) y;
- B. s = (B1) g;
- C. s = y;
- D. g = y;
- E. s = g;

13. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```

public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}

```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

14. Date le dichiarazioni:

```

class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();

```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

15. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

```

```

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- 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 D1 {

```

```

    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}

```

- A. 12312
 - B. Errore a tempo di compilazione
 - C. 1Exception in thread "main" MyExc3
 - D. 13Exception in thread "main" MyExc2
 - 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. 34

1. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}
```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

2. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {
    public Tree(String f){
        super(f);
    }
}
```

```
public void stampa() {
    System.out.println("Tree: "
        + family);
}

public static void main(String[] args){
    Plant a = new Tree("Olive");
    a.stampa();
}
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- 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 Bl {
    public static void main(String [] argv)
        throws Exception {

        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }

    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
```

```

        System.out.print(5);
    }
}

```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

4. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea(){}

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. s = (B1) y;
- B. s = (B1) g;
- C. s = y;
- D. g = y;
- E. s = g;

6. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print ("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print ("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print ("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z ();
        Y y = new Z ();
        Z z = new Z ();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

8. Date le dichiarazioni:

```

class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B1 s;
C0 t;
B2 z;
z = new B2 ();
t = new B1 ();
s = new B1 ();

```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

9. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    public static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
- D. Un'istanza della classe I può essere costruita con new C.I().
- E. Nessuna delle precedenti.

10. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print ("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print ("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print ("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z ();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

11. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

12. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

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

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

13. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}
```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

14. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

15. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }
}

```

```

}
public Tree(){}

public void stampa() {
    System.out.println("Tree: "
        + family);
}

public static void main(String[] args){
    Plant a = new Tree(Olive);
    a.stampa();
}
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

16. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
 - B. Una classe non può essere dichiarata final
 - C. Il minimo numero di elementi che un array può contenere è 0
 - D. In Java tutti i tipi numerici sono con segno
 - E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente
-

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?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}
```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

2. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
```

```
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

3. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 110

- C. 100
 - D. 011
 - E. 111
-

4. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2 ();
t = new B1 ();
s = new B1 ();
```

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

- A. `s = (B1) z;`
 - B. `z = (B2) s;`
 - C. `z = (B2) t;`
 - D. `s = (B1) t;`
 - E. Nessuno dei precedenti
-

5. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
 - B. Una classe non può essere dichiarata `final`
 - C. Il minimo numero di elementi che un array può contenere è 0
 - D. In Java tutti i tipi numerici sono con segno
 - E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente
-

6. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}
```

```
class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle", 3);
        t.print();
    }
}
```

- A. Geometric figure with 0 angles.
 - B. Geometric figure with 3 angles.
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - 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 D1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}
```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2

E. Nessuna delle precedenti

8. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

9. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}
```

A. 4

B. 5

C. 6

D. 7

E. Nessuna delle precedenti.

10. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea(){}

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}
```

A. Geometric figure with 0 angles.

B. Geometric figure with 3 angles.

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

11. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}
```

```

class Tree extends Plant {

    public Tree(String f) {
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args) {
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Qual'è l'output di questo programma?

```

class Plant {

    String family;
    protected Plant(String f) {
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f) {
        super(f);
    }
    public Tree() {}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args) {
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive

- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

13. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. s = (B1) y;
- B. s = (B1) g;
- C. s = y;
- D. g = y;
- E. s = g;

14. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ

D. Errore a tempo di compilazione.

E. Nessuna delle precedenti.

15. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. Un'istanza della classe I può essere costruita solo all'interno della classe C.

B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.

C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().

D. Un'istanza della classe I può essere costruita con new C.I().

E. Nessuna delle precedenti.

16. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}
```

```
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}
```

```
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
```

```
class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

A. XXX

B. XYZ

C. ZZZ

D. Errore a tempo di compilazione.

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. Date le dichiarazioni:

```
class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Sub1 g;
B1 s;
Super y;
```

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

- A. `s = (B1) y;`
- B. `s = (B1) g;`
- C. `s = y;`
- D. `g = y;`
- E. `s = g;`

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
    }
}
```

```
catch( MyExc1 j ) {
    System.out.print(4);
    throw( new MyExc3() );
}
catch( Exception z ) {
    System.out.print(5);
}
}
```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

3. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.

- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}
```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

5. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
```

```
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

6. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2 ();
t = new B1 ();
s = new B1 ();
```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

7. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
```

```

        + family);
    }
    public static void main(String[] args) {
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}

```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

9. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

```

```

    }
}
class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

10. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    public static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
- D. Un'istanza della classe I può essere costruita con new C.I().
- E. Nessuna delle precedenti.

11. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    protected Plant(String f) {
        family = f;
    }
    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {
    public Tree(String f) {
        super(f);
    }
    public Tree() {}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args) {
        Plant a = new Tree(Olive);
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea() {}

    public void print() {

```

```

        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle", 3);
        ((Triangle)t).print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

13. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

14. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

15. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
 - B. XYZ
 - C. ZZZ
 - D. Errore a tempo di compilazione.
 - E. Nessuna delle precedenti.
-

16. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
 - B. 1342
 - C. 134
 - D. 134Exception in thread "main" java.lang.Exception
 - 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. 37

1. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

2. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
```

```
        + family);
    }
}

class Tree extends Plant {
    public Tree(String f){
        super(f);
    }
    public Tree(){}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree(Olive);
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```

    if(a3 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b1 == b2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

4. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }
    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

6. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
    }
}

```

```

    Y y = (Y)x;
    Z z = (Z)y;
    z.f(x); z.f(y); z.f(z);
}
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

7. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. s = (B1) y;
- B. s = (B1) g;
- C. s = y;
- D. g = y;
- E. s = g;

8. Qual'è l'output di questo codice?

```

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

```

```

}
finally {
    System.out.print(4);
}
}
}

```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}

```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

10. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
 - B. Viene lanciata una `NullPointerException`.
 - C. Viene lanciata una `ArrayIndexOutOfBoundsException`.
 - D. `null`
 - E. Nessuna delle precedenti.
-

11. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
 - B. Una classe non può essere dichiarata `final`
 - C. Il minimo numero di elementi che un array può contenere è 0
 - D. In Java tutti i tipi numerici sono con segno
 - E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente
-

12. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}
```

- A. 14... (ciclo infinito)
 - B. Errore a tempo di compilazione
 - C. 143
 - D. 1432
 - E. Nessuna delle precedenti
-

13. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe `I` può essere costruita solo all'interno della classe `C`.
 - B. Un'istanza della classe `I` può essere costruita solo all'interno del pacchetto `pk`.
 - C. All'interno del pacchetto `pk`, un'istanza della classe `I` può essere costruita con `new I()`.
 - D. Un'istanza della classe `I` può essere costruita con `new C.I()`.
 - E. Nessuna delle precedenti.
-

14. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();
```

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

- A. `s = (B1) z;`
 - B. `z = (B2) s;`
 - C. `z = (B2) t;`
 - D. `s = (B1) t;`
 - E. Nessuno dei precedenti
-

15. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```

public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}

```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

16. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {

```

```

        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea() {}

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}

```

- A. Geometric figure with 0 angles.
 - B. Geometric figure with 3 angles.
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - 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. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- 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 B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
```

```
        m();
        System.out.print(2);
    }
    catch( MyExc3 d ) {
        System.out.print(3);
    }
}
static void m() throws Exception {
    try {
        if(true) throw( new MyExc1() );
        else throw( new MyExc3() );
    }
    catch( MyExc3 c ) {
        throw( new MyExc3() );
    }
    catch( MyExc1 j ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( Exception z ) {
        System.out.print(5);
    }
}
}
```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

3. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
```

```

protected void f(Z z) {
    System.out.print("Z");
}
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
 - B. XYZ
 - C. ZZZ
 - D. Errore a tempo di compilazione.
 - E. Nessuna delle precedenti.
-

4. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    protected Plant(String f) {
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {
    public Tree(String f) {
        super(f);
    }
    public Tree() {}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args) {
        Plant a = new Tree(Olive);
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

5. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```

public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}

```

- A. 4
 - B. 5
 - C. 6
 - D. 7
 - E. Nessuna delle precedenti.
-

6. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    public static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
 - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
 - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
 - D. Un'istanza della classe I può essere costruita con new C.I().
 - E. Nessuna delle precedenti.
-

7. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea() {}

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. `s = (B1) y;`
- B. `s = (B1) g;`
- C. `s = y;`
- D. `g = y;`
- E. `s = g;`

9. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

10. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}

```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

12. Quale output si ottiene invocando il metodo q?

```

class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {

```

```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

13. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    Plant(String f){
        family = f;
    }
    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

```

```

class Tree extends Plant {
    public Tree(String f){
        super(f);
    }
    public void stampa() {
        System.out.println("Tree: "
            + family);
    }
    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){

```

```

try {
    System.out.print (1);
    m();
    System.out.print (2);
}
catch( Exception w ) {
}
finally {
    throw( new Exception() );
}
}
static void m() {
    try {
        System.out.print (3);
        throw( new Exception() );
    }
    finally {
        System.out.print (4);
    }
}
}

```

- A. Errore a tempo di compilazione
 - B. 1342
 - C. 134
 - D. 134Exception in thread "main" java.lang.Exception
 - E. Nessuna delle precedenti
-

15. Date le dichiarazioni:

```

class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B1 s;
C0 t;
B2 z;
z = new B2 ();
t = new B1 ();
s = new B1 ();

```

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

- A. s = (B1) z;
 - B. z = (B2) s;
 - C. z = (B2) t;
 - D. s = (B1) t;
 - E. Nessuno dei precedenti
-

16. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
 - B. Viene lanciata una NullPointerException.
 - C. Viene lanciata una ArrayIndexOutOfBoundsException.
 - D. null
 - 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. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con `new I()`.
- D. Un'istanza della classe I può essere costruita con `new C.I()`.
- E. Nessuna delle precedenti.

2. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}
```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

3. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }
}
```

```

}
abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea(){}

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- 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 B1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {

```

```

        System.out.print(5);
    }
}

```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

6. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. s = (B1) y;
- B. s = (B1) g;
- C. s = y;
- D. g = y;
- E. s = g;

7. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

8. Quale output si ottiene invocando il metodo q?

```

class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

9. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

10. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){

```

```

        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

11. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
 - B. XYZ
 - C. ZZZ
 - D. Errore a tempo di compilazione.
 - E. Nessuna delle precedenti.
-

12. Qual'è l'output di questo programma?

```
class Plant {  
  
    String family;  
    protected Plant(String f){  
        family = f;  
    }  
  
    protected void stampa() {  
        System.out.println("Plant: "  
            + family);  
    }  
}  
  
class Tree extends Plant {  
  
    public Tree(String f){  
        super(f);  
    }  
    public Tree(){}  
  
    public void stampa() {  
        System.out.println("Tree: "  
            + family);  
    }  
  
    public static void main(String[] args){  
        Plant a = new Tree(Olive);  
        a.stampa();  
    }  
}
```

- A. Plant: Olive
 - B. Tree: Olive
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

13. Date le dichiarazioni:

```
class B1 extends C0 {...}  
class C0 extends Object {...}  
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;  
C0 t;  
B2 z;  
z = new B2();  
t = new B1();  
s = new B1();
```

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

- A. s = (B1) z;
 - B. z = (B2) s;
 - C. z = (B2) t;
 - D. s = (B1) t;
 - E. Nessuno dei precedenti
-

14. Qual'è l'output di questo programma?

```
abstract class Figure {  
    String type;  
    int angles;  
  
    protected Figure(String t, int a) {  
        type = t;  
        angles = a;  
    }  
  
    abstract void getArea();  
}  
  
class Triangle extends Figure {  
    public Triangle(String t, int a) {  
        super(t, a);  
    }  
  
    public void print() {  
        System.out.print("Geometric figure with ");  
        System.out.print(angles + " angles.");  
    }  
  
    public static void main(String[] args) {  
        Triangle t = new Triangle("triangle" , 3);  
        t.print();  
    }  
}
```

- A. Geometric figure with 0 angles.
 - B. Geometric figure with 3 angles.
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - 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 D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}

```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}
class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

16. Qual'è l'output di questo programma?

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?

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

        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}
```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

2. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.

- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

3. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
- D. Un'istanza della classe I può essere costruita con new C.I().
- E. Nessuna delle precedenti.

4. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

5. Quale output si ottiene invocando il metodo q?

```

class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

6. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {

```

```

            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}

```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

7. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

8. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea() {}

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle", 3);
        ((Triangle)t).print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Date le dichiarazioni:

```

class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();

```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;

E. Nessuno dei precedenti

10. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {
    public Tree(String f){
        super(f);
    }
    public Tree() {}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree(Olive);
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

```

```

    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

12. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

13. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    Plant(String f) {
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {
    public Tree(String f) {
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args) {
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. s = (B1) y;
- B. s = (B1) g;
- C. s = y;
- D. g = y;
- E. s = g;

15. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.

- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

16. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```

public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}

```

- A. 4
- B. 5
- C. 6
- D. 7
- 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?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}
```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

2. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    Plant(String f){
        family = f;
    }
    protected void stampa() {
```

```
        System.out.println("Plant: "
            + family);
    }
}
class Tree extends Plant {
    public Tree(String f){
        super(f);
    }
    public void stampa() {
        System.out.println("Tree: "
            + family);
    }
    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();
```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;

- D. `s = (B1) t;`
- E. Nessuno dei precedenti

- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Date le dichiarazioni:

```
class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Sub1 g;
B1 s;
Super y;
```

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

- A. `s = (B1) y;`
- B. `s = (B1) g;`
- C. `s = y;`
- D. `g = y;`
- E. `s = g;`

5. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.

6. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {
    public Tree(String f){
        super(f);
    }
    public Tree(){ }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree(Olive);
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
```

```

public Triangle(String t, int a) {
    super(t, a);
}
void getArea() {}

public void print() {
    System.out.print("Geometric figure with ");
    System.out.print(angles + " angles.");
}

public static void main(String[] args) {
    Figure t = new Triangle("triangle" , 3);
    ((Triangle)t).print();
}
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Quale output si ottiene invocando il metodo q?

```

class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

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

9. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}

```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

10. Qual'è l'output di questo codice?

```

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

```

```

        throw( new Exception() );
    }
}
static void m() {
    try {
        System.out.print(3);
        throw( new Exception() );
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

11. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```

public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}

```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

12. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

13. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

14. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    public static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.

- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
 - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
 - D. Un'istanza della classe I può essere costruita con new C.I().
 - E. Nessuna delle precedenti.
-

15. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
 - B. Una classe non può essere dichiarata final
 - C. Il minimo numero di elementi che un array può contenere è 0
 - D. In Java tutti i tipi numerici sono con segno
 - E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente
-

16. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}
```

```
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
 - B. XYZ
 - C. ZZZ
 - D. Errore a tempo di compilazione.
 - 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. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();
```

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

- A. `s = (B1) z;`
- B. `z = (B2) s;`
- C. `z = (B2) t;`
- D. `s = (B1) t;`
- E. Nessuno dei precedenti

2. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe `I` può essere costruita solo all'interno della classe `C`.
- B. Un'istanza della classe `I` può essere costruita solo all'interno del pacchetto `pk`.
- C. All'interno del pacchetto `pk`, un'istanza della classe `I` può essere costruita con `new I()`.
- D. Un'istanza della classe `I` può essere costruita con `new C.I()`.
- E. Nessuna delle precedenti.

3. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

4. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}
```

```

}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    protected Plant(String f) {
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {
    public Tree(String f) {
        super(f);
    }
    public Tree() {}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args) {
        Plant a = new Tree(Olive);
        a.stampa();
    }
}

```

- A. Plant: Olive

- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- 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 B1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}

```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

7. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

```

```

    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

8. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```

public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}

```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

9. Quale output si ottiene invocando il metodo q?

```

class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
}

```

```

private Boolean b3 = b1;
void q() {
    float [] a2 = new float [4];
    m(f1, a2, b3);
}

void m(Float f2, float [] a1, Boolean b2){
    a3 = new float [4];
    if(f1 == f2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a3 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b1 == b2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

10. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}
```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

12. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.

- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

13. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    Plant(String f){
        family = f;
    }
    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}
class Tree extends Plant {
    public Tree(String f){
        super(f);
    }
    public void stampa() {
        System.out.println("Tree: "
            + family);
    }
    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

15. Date le dichiarazioni:

```
class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Sub1 g;
B1 s;
Super y;
```

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

- A. `s = (B1) y;`
 - B. `s = (B1) g;`
 - C. `s = y;`
 - D. `g = y;`
 - E. `s = g;`
-

16. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
```

```
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea(){}

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}
```

- A. Geometric figure with 0 angles.
 - B. Geometric figure with 3 angles.
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - 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. Date le dichiarazioni:

```
class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Sub1 g;
B1 s;
Super y;
```

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

- A. `s = (B1) y;`
- B. `s = (B1) g;`
- C. `s = y;`
- D. `g = y;`
- E. `s = g;`

2. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}
```

```
class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}
```

```
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
```

```
class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
```

```
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

3. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una `NullPointerException`.
- C. Viene lanciata una `ArrayIndexOutOfBoundsException`.
- D. null
- E. Nessuna delle precedenti.

4. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}
```

- A. 4
- B. 5
- C. 6
- D. 7
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}
```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

6. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}
```

```
class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea(){}

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

8. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();
```

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

- A. `s = (B1) z;`
- B. `z = (B2) s;`
- C. `z = (B2) t;`
- D. `s = (B1) t;`
- E. Nessuno dei precedenti

9. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    Plant(String f) {
        family = f;
    }
    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}
class Tree extends Plant {
    public Tree(String f) {
        super(f);
    }
    public void stampa() {
        System.out.println("Tree: "
            + family);
    }
    public static void main(String[] args) {
        Plant a = new Tree("Olive");
        a.stampa();
    }
}
```

- A. Plant: Olive

- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

10. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}
class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

11. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2) {
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```

    if(a3 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b1 == b2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

12. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

13. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    public static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
- D. Un'istanza della classe I può essere costruita con new C.I().
- 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 B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}

```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432

E. Nessuna delle precedenti

15. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
 - B. Una classe non può essere dichiarata final
 - C. Il minimo numero di elementi che un array può contenere è 0
 - D. In Java tutti i tipi numerici sono con segno
 - E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente
-

16. Qual'è l'output di questo programma?

```
class Plant {  
  
    String family;  
    protected Plant(String f){  
        family = f;  
    }  
  
    protected void stampa() {  
        System.out.println("Plant: "
```

```
        + family);  
    }  
}  
  
class Tree extends Plant {  
  
    public Tree(String f){  
        super(f);  
    }  
    public Tree(){}  
  
    public void stampa() {  
        System.out.println("Tree: "  
            + family);  
    }  
  
    public static void main(String[] args){  
        Plant a = new Tree(Olive);  
        a.stampa();  
    }  
}
```

- A. Plant: Olive
 - B. Tree: Olive
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - 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. 44

1. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

2. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }
    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}
```

```
class Tree extends Plant {
    public Tree(String f){
        super(f);
    }
    public Tree(){ }
    public void stampa() {
        System.out.println("Tree: "
            + family);
    }
    public static void main(String[] args){
```

```
Plant a = new Tree(Olive);
a.stampa();
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Dato un file contenente il seguente codice:

```
package pk;
public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
- D. Un'istanza della classe I può essere costruita con new C.I().
- E. Nessuna delle precedenti.

4. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

5. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
    }
}
```

```
finally {
    System.out.print(3);
    throw( new MyExc2() );
}
}
```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

7. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- 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 B1 {
    public static void main(String [] argv)
        throws Exception {
```

```

try {
    System.out.print(1);
    m();
    System.out.print(2);
}
catch( MyExc3 d ) {
    System.out.print(3);
}
}
static void m() throws Exception {
    try {
        if(true) throw( new MyExc1() );
        else throw( new MyExc3() );
    }
    catch( MyExc3 c ) {
        throw( new MyExc3() );
    }
    catch( MyExc1 j ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( Exception z ) {
        System.out.print(5);
    }
}
}

```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

9. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. s = (B1) y;
- B. s = (B1) g;
- C. s = y;
- D. g = y;
- E. s = g;

10. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```

public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}

```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

11. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    void getArea(){}

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

12. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();
```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;

E. Nessuno dei precedenti

13. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

A. Errore a tempo di compilazione

B. 1342

C. 134

D. 134Exception in thread "main" java.lang.Exception

E. Nessuna delle precedenti

14. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

A. 101

B. 110

C. 100

D. 011

E. 111

15. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    Plant(String f){
        family = f;
    }
    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}
```

```

class Tree extends Plant {

    public Tree(String f) {
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args) {
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

16. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

```

```

}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
 - B. XYZ
 - C. ZZZ
 - D. Errore a tempo di compilazione.
 - 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. 45

1. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

2. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con `new I()`.
- D. Un'istanza della classe I può essere costruita con `new C.I()`.
- E. Nessuna delle precedenti.

3. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();
```

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

- A. `s = (B1) z;`
- B. `z = (B2) s;`
- C. `z = (B2) t;`
- D. `s = (B1) t;`
- E. Nessuno dei precedenti

4. Date le dichiarazioni:

```
class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Sub1 g;
B1 s;
Super y;
```

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

- A. `s = (B1) y;`
- B. `s = (B1) g;`
- C. `s = y;`
- D. `g = y;`
- E. `s = g;`

5. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

6. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
```

```
    void f(Y y) {
        System.out.print("Y");
    }
}
```

```
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
```

```
class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

7. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

8. Qual'è l'output di questo codice?

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

```

        System.out.print(3);
        throw( new Exception() );
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

9. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```

public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}

```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

10. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

```

```

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }
    public Tree(){}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree(Olive);
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
```

```
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}
```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

14. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

15. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea(){}

    public void print() {
```

```

    System.out.print("Geometric figure with ");
    System.out.print(angles + " angles.");
}

public static void main(String[] args) {
    Figure t = new Triangle("triangle" , 3);
    ((Triangle)t).print();
}
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- 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 B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);

```

```

        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}

```

- A. 14... (ciclo infinito)
 - B. Errore a tempo di compilazione
 - C. 143
 - D. 1432
 - 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. 46

1. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

2. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno

E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

3. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    protected Plant(String f) {
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {
    public Tree(String f) {
        super(f);
    }
    public Tree() {}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args) {
        Plant a = new Tree(Olive);
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
- D. Un'istanza della classe I può essere costruita con new C.I().
- E. Nessuna delle precedenti.

5. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}
```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

7. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }
}
```

```

public static void main(String[] args) {
    Triangle t = new Triangle("triangle" , 3);
    t.print();
}
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```

public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();
}

```

```

static void makeStella(String pianeta) {
    Stella s = new Stella();
    s.pianetaVicino = pianeta;
}

```

```

public static void main(String [] arg) {
    Stella.makeStella("Marte");
    Stella.makeStella("Venere"); /* QUI */
    Stella.makeStella("Krypton");
}
}

```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

10. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. s = (B1) y;
- B. s = (B1) g;
- C. s = y;
- D. g = Y;
- E. s = g;

11. Date le dichiarazioni:

```

class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();

```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

12. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

13. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.

- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

14. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea(){}

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- 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 B1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            System.out.print(1);
            m();
        }
    }
}
```

```

        System.out.print(2);
    }
    catch( MyExc3 d ) {
        System.out.print(3);
    }
}
static void m() throws Exception {
    try {
        if(true) throw( new MyExc1() );
        else throw( new MyExc3() );
    }
    catch( MyExc3 c ) {
        throw( new MyExc3() );
    }
    catch( MyExc1 j ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( Exception z ) {
        System.out.print(5);
    }
}
}

```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

16. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
```

```

class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- 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. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

2. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
```

```
private void f(Y y) {
    System.out.print("Y");
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

3. Date le dichiarazioni:

```
class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Sub1 g;
B1 s;
Super y;
```

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

- A. s = (B1) y;
- B. s = (B1) g;
- C. s = y;
- D. g = y;
- E. s = g;

4. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

5. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

6. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

7. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();
```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

8. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}
```

class Tree extends Plant {

```
    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
```

```

        + family);
    }
    public static void main(String[] args) {
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }
    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}
class Tree extends Plant {
    public Tree(String f){
        super(f);
    }
    public Tree(){}
    public void stampa() {
        System.out.println("Tree: "
            + family);
    }
    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- 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 B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}

```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

11. Dato un file contenente il seguente codice:

```

package pk;
public class C {
    public static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con `new I()`.
- D. Un'istanza della classe I può essere costruita con `new C.I()`.
- E. Nessuna delle precedenti.

12. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea() {}

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

13. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}
```

15. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
```

- A. 4
- B. 5
- C. 6
- D. 7
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}
```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

```

protected void f(Z z) {
    System.out.print("Z");
}
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
 - B. XYZ
 - C. ZZZ
 - D. Errore a tempo di compilazione.
 - E. Nessuna delle precedenti.
-

16. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;

```

```

        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}

```

- A. Geometric figure with 0 angles.
 - B. Geometric figure with 3 angles.
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - 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 programma?

```
class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }
    public Tree(){}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree(Olive);
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();
```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

3. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

4. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}
```

```

    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print ("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z ();
        Y y = new Z ();
        Z z = new Z ();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

5. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print(){
        System.out.print ("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.

- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

6. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. s = (B1) y;
- B. s = (B1) g;
- C. s = y;
- D. g = y;
- E. s = g;

7. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print ("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print ("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print ("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z ();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.

E. Nessuna delle precedenti.

8. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

9. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}
```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

10. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea(){}

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {
    public Tree(String f){
        super(f);
    }

    public void stampa() {
```

```

        System.out.println("Tree: "
            + family);
    }
    public static void main(String[] args) {
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    public static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con `new I()`.
- D. Un'istanza della classe I può essere costruita con `new C.I()`.
- E. Nessuna delle precedenti.

13. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```

public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}

```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

14. Quale output si ottiene invocando il metodo q?

```

class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

15. Qual'è l'output di questo codice?

```

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

        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
    }
}

```

```

    }
    catch( MyExc2 f ) {
        System.out.print(1);
        throw( new MyExc3() );
    }
    catch( Exception d ) {
        System.out.print(2);
    }
    finally {
        System.out.print(3);
        throw( new MyExc2() );
    }
}
}

```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

16. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){

```

```

        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
 - B. 1342
 - C. 134
 - D. 134Exception in thread "main" java.lang.Exception
 - 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 MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

2. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
- D. Un'istanza della classe I può essere costruita con new C.I().
- E. Nessuna delle precedenti.

3. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

4. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {
```

```

public Tree(String f) {
    super(f);
}

public void stampa() {
    System.out.println("Tree: "
        + family);
}

public static void main(String[] args) {
    Plant a = new Tree("Olive");
    a.stampa();
}
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}

```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

6. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    void getArea() {}

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {

```

```

X x = new Z ();
Y y = new Z ();
Z z = new Z ();
z.f(x); z.f(y); z.f(z);
}
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

8. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final

- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

10. Qual'è l'output di questo programma?

```

class Plant {

    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }
    public Tree(){}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree(Olive);
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. `s = (B1) y;`
- B. `s = (B1) g;`
- C. `s = y;`
- D. `g = y;`
- E. `s = g;`

12. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- 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 B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
    }
}
```

```
        System.out.print(2);
    }
    catch( MyExc3 d ) {
        System.out.print(3);
    }
}
static void m() throws Exception {
    try {
        if(true) throw( new MyExc1() );
        else throw( new MyExc3() );
    }
    catch( MyExc3 c ) {
        throw( new MyExc3() );
    }
    catch( MyExc1 j ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( Exception z ) {
        System.out.print(5);
    }
}
}
```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

14. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```
}  
}
```

- A. 101
 - B. 110
 - C. 100
 - D. 011
 - E. 111
-

15. Date le dichiarazioni:

```
class B1 extends C0 {...}  
class C0 extends Object {...}  
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;  
C0 t;  
B2 z;  
z = new B2();  
t = new B1();  
s = new B1();
```

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

- A. `s = (B1) z;`
- B. `z = (B2) s;`
- C. `z = (B2) t;`

- D. `s = (B1) t;`
 - E. Nessuno dei precedenti
-

16. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {  
    static String pianetaVicino = new String();  
    String nome = new String();  
  
    static void makeStella(String pianeta) {  
        Stella s = new Stella();  
        s.pianetaVicino = pianeta;  
    }  
  
    public static void main(String [] arg) {  
        Stella.makeStella("Marte");  
        Stella.makeStella("Venere"); /* QUI */  
        Stella.makeStella("Krypton");  
    }  
}
```

- A. 4
 - B. 5
 - C. 6
 - D. 7
 - 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 programma?

```
class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {
    public Tree(String f){
        super(f);
    }
    public Tree(){}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree(Olive);
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- 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 B1 {
    public static void main(String [] argv)
```

```
throws Exception {
    try {
        System.out.print(1);
        m();
        System.out.print(2);
    }
    catch( MyExc3 d ) {
        System.out.print(3);
    }
}
static void m() throws Exception {
    try {
        if(true) throw( new MyExc1() );
        else throw( new MyExc3() );
    }
    catch( MyExc3 c ) {
        throw( new MyExc3() );
    }
    catch( MyExc1 j ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( Exception z ) {
        System.out.print(5);
    }
}
}
```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
```

```

    catch( Exception w ) {
    }
    finally {
        throw( new Exception() );
    }
}
static void m() {
    try {
        System.out.print(3);
        throw( new Exception() );
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

4. Date le dichiarazioni:

```

class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();

```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

5. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {

```

```

        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea() {}

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle", 3);
        ((Triangle)t).print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

6. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
 - B. XYZ
 - C. ZZZ
 - D. Errore a tempo di compilazione.
 - E. Nessuna delle precedenti.
-

7. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}
```

- A. Geometric figure with 0 angles.
 - B. Geometric figure with 3 angles.
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

8. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
```

```
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}
```

- A. 4
 - B. 5
 - C. 6
 - D. 7
 - E. Nessuna delle precedenti.
-

9. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {
    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}
```

- A. Plant: Olive
 - B. Tree: Olive
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

10. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0

- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

- C. All'interno del pacchetto `pk`, un'istanza della classe `I` può essere costruita con `new I()`.
- D. Un'istanza della classe `I` può essere costruita con `new C.I()`.
- E. Nessuna delle precedenti.

11. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

12. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe `I` può essere costruita solo all'interno della classe `C`.
- B. Un'istanza della classe `I` può essere costruita solo all'interno del pacchetto `pk`.

13. Date le dichiarazioni:

```
class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Sub1 g;
B1 s;
Super y;
```

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

- A. `s = (B1) y;`
- B. `s = (B1) g;`
- C. `s = y;`
- D. `g = y;`
- E. `s = g;`

14. Quale output si ottiene invocando il metodo `q`?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101

- B. 110
 - C. 100
 - D. 011
 - E. 111
-

15. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print (1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print (2);
        }
        finally {
            System.out.print (3);
            throw( new MyExc2() );
        }
    }
}
```

```
    }
}
}
```

- A. 12312
 - B. Errore a tempo di compilazione
 - C. 1Exception in thread "main" MyExc3
 - D. 13Exception in thread "main" MyExc2
 - E. Nessuna delle precedenti
-

16. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
 - B. Viene lanciata una NullPointerException.
 - C. Viene lanciata una ArrayIndexOutOfBoundsException.
 - D. null
 - 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. Date le dichiarazioni:

```
class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Sub1 g;
B1 s;
Super y;
```

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

- A. s = (B1) y;
- B. s = (B1) g;
- C. s = y;
- D. g = y;
- E. s = g;

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

```
}
}
}
```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}
```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2

E. Nessuna delle precedenti

4. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();
```

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

- A. `s = (B1) z;`
 - B. `z = (B2) s;`
 - C. `z = (B2) t;`
 - D. `s = (B1) t;`
 - E. Nessuno dei precedenti
-

5. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}
```

- A. 4
 - B. 5
 - C. 6
 - D. 7
 - E. Nessuna delle precedenti.
-

6. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe `I` può essere costruita solo all'interno della classe `C`.
 - B. Un'istanza della classe `I` può essere costruita solo all'interno del pacchetto `pk`.
 - C. All'interno del pacchetto `pk`, un'istanza della classe `I` può essere costruita con `new I()`.
 - D. Un'istanza della classe `I` può essere costruita con `new C.I()`.
 - E. Nessuna delle precedenti.
-

7. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
 - B. Una classe non può essere dichiarata `final`
 - C. Il minimo numero di elementi che un array può contenere è 0
 - D. In Java tutti i tipi numerici sono con segno
 - E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente
-

8. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
    }
}
```

```

    Y y = new Z ();
    Z z = new Z ();
    z.f(x); z.f(y); z.f(z);
}
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

9. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

10. Quale output si ottiene invocando il metodo q?

```

class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
}

```

```

void q() {
    float [] a2 = new float [4];
    m(f1, a2, b3);
}

void m(Float f2, float [] a1, Boolean b2){
    a3 = new float [4];
    if(f1 == f2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a3 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b1 == b2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

11. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }
    public Tree(){ }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
}

```

```

    Plant a = new Tree(Olive);
    a.stampa();
}
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

13. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

```

```

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

```

```

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

```

```

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

14. Qual'è l'output di questo programma?

```

class Plant {

    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

```

```

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

15. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

```

```

}
class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea() {}

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle", 3);
        ((Triangle)t).print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- 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 B1 {
    public static void main(String [] argv)
        throws Exception {

```

```

        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}

```

- A. 14... (ciclo infinito)
 - B. Errore a tempo di compilazione
 - C. 143
 - D. 1432
 - 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?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
}
```

```
    }
}
static void p() throws Exception {
    try {
        throw( new MyExc2() );
    }
    catch( MyExc2 f ) {
        System.out.print(1);
        throw( new MyExc3() );
    }
    catch( Exception d ) {
        System.out.print(2);
    }
    finally {
        System.out.print(3);
        throw( new MyExc2() );
    }
}
}
```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- 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 B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
        }
    }
}
```

```

        else throw( new MyExc3() );
    }
    catch( MyExc3 c ) {
        throw( new MyExc3() );
    }
    catch( MyExc1 j ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( Exception z ) {
        System.out.print(5);
    }
}
}

```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

4. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

5. Date le dichiarazioni:

```

class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();

```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

6. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

7. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea(){}

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
}
```

```
public void print(){
    System.out.print("Geometric figure with ");
    System.out.print(angles + " angles.");
}

public static void main(String[] args) {
    Triangle t = new Triangle("triangle" , 3);
    t.print();
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

10. Date le dichiarazioni:

```
class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Sub1 g;
B1 s;
Super y;
```

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

- A. `s = (B1) y;`
- B. `s = (B1) g;`
- C. `s = y;`
- D. `g = y;`
- E. `s = g;`

11. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con `new I()`.
- D. Un'istanza della classe I può essere costruita con `new C.I()`.

E. Nessuna delle precedenti.

12. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    protected Plant(String f) {
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f) {
        super(f);
    }
    public Tree() {}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args) {
        Plant a = new Tree(Olive);
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

13. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
    }
}
```

```
Stella.makeStella("Krypton");
}
}
```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

14. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

15. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2) {
        a3 = new float [4];
    }
}
```

```

if(f1 == f2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(a3 == a1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(b1 == b2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
}

```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

16. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

```

```

}
}
class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
 - B. XYZ
 - C. ZZZ
 - D. Errore a tempo di compilazione.
 - 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 programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea(){}

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual'è l'output di questo codice?

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

```
        m();
        System.out.print(2);
    }
    catch( Exception w ) {
    }
    finally {
        throw( new Exception() );
    }
}
static void m() {
    try {
        System.out.print(3);
        throw( new Exception() );
    }
    finally {
        System.out.print(4);
    }
}
}
```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

3. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
```

```

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

```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

4. Qual'è l'output di questo programma?

```

class Plant {

    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }
    public Tree(){
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree(Olive);
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

6. Date le dichiarazioni:

```

class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();

```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;

E. Nessuno dei precedenti

7. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con `new I()`.
- D. Un'istanza della classe I può essere costruita con `new C.I()`.
- E. Nessuna delle precedenti.

8. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.

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 Bl {
    public static void main(String [] argv)
        throws Exception {

        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}
```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

10. Date le dichiarazioni:

```
class Super extends Object {...}
class Sub1 extends Super {...}
class Bl extends Super {...}
```

e le dichiarazioni di variabile:

```
Sub1 g;
Bl s;
Super y;
```

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

- A. s = (B1) y;
- B. s = (B1) g;
- C. s = y;
- D. g = y;
- E. s = g;

11. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}
```

```
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

13. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

14. Qual'è l'output di questo codice?

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

        try {
            p();
        }
    }

    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
    }
}
```

```

        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}

```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

15. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```

public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}

```

```

    }
}

```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

16. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- 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. 54

1. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
```

```
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

3. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con `new I()`.
- D. Un'istanza della classe I può essere costruita con `new C.I()`.

E. Nessuna delle precedenti.

4. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}
```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

5. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea(){}

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

6. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

7. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
 - B. Una classe non può essere dichiarata final
 - C. Il minimo numero di elementi che un array può contenere è 0
 - D. In Java tutti i tipi numerici sono con segno
 - E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente
-

8. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}
```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

9. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

10. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Qual'è l'output di questo codice?

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

```

        System.out.print(4);
    }
}

```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

12. Date le dichiarazioni:

```

class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();

```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

13. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

```

```

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

14. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    protected Plant(String f) {
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {
    public Tree(String f) {
        super(f);
    }
    public Tree() {}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args) {
        Plant a = new Tree(Olive);
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

15. Date le dichiarazioni:

```
class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Sub1 g;
B1 s;
Super y;
```

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

- A. s = (B1) y;
- B. s = (B1) g;
- C. s = y;
- D. g = y;
- E. s = g;

16. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print (1);
            m();
        }
    }
}
```

```
        System.out.print (2);
    }
    catch( MyExc3 d ) {
        System.out.print (3);
    }
}
static void m() throws Exception {
    try {
        if(true) throw( new MyExc1() );
        else throw( new MyExc3() );
    }
    catch( MyExc3 c ) {
        throw( new MyExc3() );
    }
    catch( MyExc1 j ) {
        System.out.print (4);
        throw( new MyExc3() );
    }
    catch( Exception z ) {
        System.out.print (5);
    }
}
}
```

- A. 14... (ciclo infinito)
 - B. Errore a tempo di compilazione
 - C. 143
 - D. 1432
 - 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 programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
    }
}
```

```
}
catch( Exception w ) {
}
finally {
    throw( new Exception() );
}
}
static void m() {
    try {
        System.out.print(3);
        throw( new Exception() );
    }
    finally {
        System.out.print(4);
    }
}
}
```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

3. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

4. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }
    public Tree(){}

    public void stampa() {
        System.out.println("Tree: "

```

```

        + family);
    }

    public static void main(String[] args){
        Plant a = new Tree(Olive);
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

6. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

```

```

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

7. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    public static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con `new I()`.
- D. Un'istanza della classe I può essere costruita con `new C.I()`.
- E. Nessuna delle precedenti.

8. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea() {}

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Date le dichiarazioni:

```
class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Sub1 g;
B1 s;
Super y;
```

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

- A. `s = (B1) y;`
- B. `s = (B1) g;`
- C. `s = y;`
- D. `g = y;`
- E. `s = g;`

8. Qual'è l'output di questo programma?

10. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}
```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

11. Quale output si ottiene invocando il metodo q?

```

class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

12. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```

public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}

```

- A. 4
- B. 5
- C. 6

D. 7

E. Nessuna delle precedenti.

13. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

A. Plant: Olive

B. Tree: Olive

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

14. Dire quale delle seguenti affermazioni è falsa:

A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array

B. Una classe non può essere dichiarata final

C. Il minimo numero di elementi che un array può contenere è 0

D. In Java tutti i tipi numerici sono con segno

E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

15. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}

```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3

- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

16. Date le dichiarazioni:

```

class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();

```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- 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. 56

1. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

2. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno

E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

3. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]) {
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

4. Qual'è l'output di questo codice?

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

```

        System.out.print(4);
    }
}

```

- A. Errore a tempo di compilazione
 - B. 1342
 - C. 134
 - D. 134Exception in thread "main" java.lang.Exception
 - E. Nessuna delle precedenti
-

5. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea(){}

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}

```

- A. Geometric figure with 0 angles.
 - B. Geometric figure with 3 angles.
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

6. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    Plant(String f){

```

```

        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

- A. Plant: Olive
 - B. Tree: Olive
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

7. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}

```

```
}  
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Qual'è l'output di questo programma?

```
class Plant {  
  
    String family;  
    protected Plant(String f){  
        family = f;  
    }  
  
    protected void stampa() {  
        System.out.println("Plant: "  
            + family);  
    }  
}  
  
class Tree extends Plant {  
  
    public Tree(String f){  
        super(f);  
    }  
    public Tree(){}  
  
    public void stampa() {  
        System.out.println("Tree: "  
            + family);  
    }  
  
    public static void main(String[] args){  
        Plant a = new Tree(Olive);  
        a.stampa();  
    }  
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Date le dichiarazioni:

```
class Super extends Object {...}  
class Sub1 extends Super {...}  
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Sub1 g;  
B1 s;  
Super y;
```

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

- A. s = (B1) y;
- B. s = (B1) g;
- C. s = y;
- D. g = y;
- E. s = g;

10. Date le dichiarazioni:

```
class B1 extends C0 {...}  
class C0 extends Object {...}  
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;  
C0 t;  
B2 z;  
z = new B2();  
t = new B1();  
s = new B1();
```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

11. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {  
    static String pianetaVicino = new String();  
    String nome = new String();  
  
    static void makeStella(String pianeta) {  
        Stella s = new Stella();  
        s.pianetaVicino = pianeta;  
    }  
  
    public static void main(String [] arg) {  
        Stella.makeStella("Marte");  
        Stella.makeStella("Venere"); /* QUI */  
        Stella.makeStella("Krypton");  
    }  
}
```

A. 4

- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

12. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
- D. Un'istanza della classe I può essere costruita con new C.I().
- E. Nessuna delle precedenti.

13. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX

14. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

15. Qual'è l'output di questo codice?

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

```

    }
    catch( MyExc2 f ) {
        System.out.print(1);
        throw( new MyExc3() );
    }
    catch( Exception d ) {
        System.out.print(2);
    }
    finally {
        System.out.print(3);
        throw( new MyExc2() );
    }
}
}

```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- 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 B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);

```

```

        m();
        System.out.print(2);
    }
    catch( MyExc3 d ) {
        System.out.print(3);
    }
}
static void m() throws Exception {
    try {
        if(true) throw( new MyExc1() );
        else throw( new MyExc3() );
    }
    catch( MyExc3 c ) {
        throw( new MyExc3() );
    }
    catch( MyExc1 j ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( Exception z ) {
        System.out.print(5);
    }
}
}

```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- 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. Qual'è l'output di questo codice?

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

        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}
```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

2. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.

B. Viene lanciata una

NullPointerException.

C. Viene lanciata una

ArrayIndexOutOfBoundsException.

D. null

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 B1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}
```

A. 14... (ciclo infinito)

B. Errore a tempo di compilazione

C. 143

D. 1432

E. Nessuna delle precedenti

4. Date le dichiarazioni:

```
class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Sub1 g;
B1 s;
Super y;
```

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

- A. `s = (B1) y;`
- B. `s = (B1) g;`
- C. `s = y;`
- D. `g = y;`
- E. `s = g;`

5. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata `final`
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

6. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}
```

A. 4

B. 5

C. 6

D. 7

E. Nessuna delle precedenti.

7. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    Plant(String f) {
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {
    public Tree(String f) {
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args) {
        Plant a = new Tree("Olive");
        a.stampa();
    }
}
```

A. **Plant: Olive**

B. **Tree: Olive**

C. **Errore a tempo di compilazione.**

D. **Errore a tempo di esecuzione.**

E. **Nessuna delle precedenti.**

8. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();
```

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

- A. `s = (B1) z;`
 - B. `z = (B2) s;`
 - C. `z = (B2) t;`
 - D. `s = (B1) t;`
 - E. Nessuno dei precedenti
-

9. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }
    public Tree(){}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree(Olive);
        a.stampa();
    }
}
```

- A. Plant: Olive
 - B. Tree: Olive
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

10. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}
```

```
class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}
```

```
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
```

```
class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
 - B. XYZ
 - C. ZZZ
 - D. Errore a tempo di compilazione.
 - E. Nessuna delle precedenti.
-

11. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}
```

```
class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}
```

```
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
```

```
class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX

- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

12. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
- D. Un'istanza della classe I può essere costruita con new C.I().
- E. Nessuna delle precedenti.

13. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione

14. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

15. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}
```

```

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

16. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {

```

```

        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea(){}

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}

```

- A. Geometric figure with 0 angles.
 - B. Geometric figure with 3 angles.
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - 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?

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

        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}
```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

2. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno

E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

3. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
- D. Un'istanza della classe I può essere costruita con new C.I().
- E. Nessuna delle precedenti.

4. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea(){}

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }
}
```

```

}

public static void main(String[] args) {
    Figure t = new Triangle("triangle" , 3);
    ((Triangle)t).print();
}
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {
    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

6. Date le dichiarazioni:

```

class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();

```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

7. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print(){
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

9. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

10. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}
class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

11. Quale output si ottiene invocando il metodo q?

```

class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        }
    }
}

```

```

    } else {
        System.out.print(0);
    }
}

```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

12. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}

```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

13. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```

public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}

```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

14. Qual'è l'output di questo programma?

```

class Plant {

    String family;
    protected Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }
    public Tree(){}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args){
        Plant a = new Tree(Olive);
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.

- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

15. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

16. Date le dichiarazioni:

```
class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Sub1 g;
B1 s;
Super y;
```

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

- A. s = (B1) y;
 - B. s = (B1) g;
 - C. s = y;
 - D. g = y;
 - E. s = g;
-

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. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- B. Una classe non può essere dichiarata final
- C. Il minimo numero di elementi che un array può contenere è 0
- D. In Java tutti i tipi numerici sono con segno
- E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente

2. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}
```

- A. 4
- B. 5
- C. 6
- D. 7
- 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 B1 {
```

```
public static void main(String [] argv)
    throws Exception {
    try {
        System.out.print(1);
        m();
        System.out.print(2);
    }
    catch( MyExc3 d ) {
        System.out.print(3);
    }
}
static void m() throws Exception {
    try {
        if(true) throw( new MyExc1() );
        else throw( new MyExc3() );
    }
    catch( MyExc3 c ) {
        throw( new MyExc3() );
    }
    catch( MyExc1 j ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( Exception z ) {
        System.out.print(5);
    }
}
}
```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

4. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    protected Plant(String f){
        family = f;
    }
    protected void stampa() {
        System.out.println("Plant: "
```

```

        + family);
    }
}

class Tree extends Plant {

    public Tree(String f) {
        super(f);
    }
    public Tree() {}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args) {
        Plant a = new Tree(Olive);
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX

- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

6. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea() {}

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Figure t = new Triangle("triangle" , 3);
        ((Triangle)t).print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. s = (B1) y;

- B. `s = (B1) g;`
- C. `s = y;`
- D. `g = y;`
- E. `s = g;`

- C. All'interno del pacchetto `pk`, un'istanza della classe `I` può essere costruita con `new I()`.
- D. Un'istanza della classe `I` può essere costruita con `new C.I()`.
- E. Nessuna delle precedenti.

8. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}
```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

9. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe `I` può essere costruita solo all'interno della classe `C`.
- B. Un'istanza della classe `I` può essere costruita solo all'interno del pacchetto `pk`.

11. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
```

```

    super(t, a);
}

public void print() {
    System.out.print("Geometric figure with ");
    System.out.print(angles + " angles.");
}

public static void main(String[] args) {
    Triangle t = new Triangle("triangle", 3);
    t.print();
}
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Qual'è l'output di questo programma?

```

class Plant {

    String family;
    Plant(String f) {
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f) {
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args) {
        Plant a = new Tree("Olive");
        a.stampa();
    }
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

13. Date le dichiarazioni:

```

class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B1 s;
C0 t;
B2 z;
z = new B2();
t = new B1();
s = new B1();

```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

14. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1342
- C. 134
- D. 134Exception in thread "main" java.lang.Exception

E. Nessuna delle precedenti

15. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
 - B. Viene lanciata una NullPointerException.
 - C. Viene lanciata una ArrayIndexOutOfBoundsException.
 - D. null
 - E. Nessuna delle precedenti.
-

16. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
```

```
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
 - B. 110
 - C. 100
 - D. 011
 - E. 111
-

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. Date le dichiarazioni:

```
class B1 extends C0 {...}
class C0 extends Object {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B1 s;
C0 t;
B2 z;
z = new B2 ();
t = new B1 ();
s = new B1 ();
```

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

- A. s = (B1) z;
- B. z = (B2) s;
- C. z = (B2) t;
- D. s = (B1) t;
- E. Nessuno dei precedenti

2. Qual'è l'output di questo programma?

```
abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }
    void getArea() {}

    public void print() {
        System.out.print("Geometric figure with ");
    }
}
```

```
System.out.print(angles + " angles.");
}

public static void main(String[] args) {
    Figure t = new Triangle("triangle" , 3);
    ((Triangle)t).print();
}
}
```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Qual'è l'output di questo programma?

```
class Plant {
    String family;
    protected Plant(String f) {
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {
    public Tree(String f) {
        super(f);
    }
    public Tree() {}

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }

    public static void main(String[] args) {
        Plant a = new Tree(Olive);
        a.stampa();
    }
}
```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo indicato?

```
public class Stella {
    static String pianetaVicino = new String();
    String nome = new String();

    static void makeStella(String pianeta) {
        Stella s = new Stella();
        s.pianetaVicino = pianeta;
    }

    public static void main(String [] arg) {
        Stella.makeStella("Marte");
        Stella.makeStella("Venere"); /* QUI */
        Stella.makeStella("Krypton");
    }
}
```

- A. 4
- B. 5
- C. 6
- D. 7
- E. Nessuna delle precedenti.

5. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    public static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new I().
- D. Un'istanza della classe I può essere costruita con new C.I().
- 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 D1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            p();
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc2 f ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception d ) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}
```

- A. 12312
- B. Errore a tempo di compilazione
- C. 1Exception in thread "main" MyExc3
- D. 13Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

7. Quale output si ottiene invocando il metodo q?

```
class C {
    private Float f1 = new Float(1.0);
    private float [] a3;
    private Boolean b1 = new Boolean(false);
    private Boolean b3 = b1;
    void q() {
        float [] a2 = new float [4];
        m(f1, a2, b3);
    }
    void m(Float f2, float [] a1, Boolean b2){
        a3 = new float [4];
        if(f1 == f2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```

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

```

- A. 101
- B. 110
- C. 100
- D. 011
- E. 111

8. Qual'è l'output di questo programma?

```

class Plant {
    String family;
    Plant(String f){
        family = f;
    }

    protected void stampa() {
        System.out.println("Plant: "
            + family);
    }
}

class Tree extends Plant {

    public Tree(String f){
        super(f);
    }

    public void stampa() {
        System.out.println("Tree: "
            + family);
    }
}

public static void main(String[] args){
    Plant a = new Tree("Olive");
    a.stampa();
}

```

- A. Plant: Olive
- B. Tree: Olive
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Date le dichiarazioni:

```

class Super extends Object {...}
class Sub1 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Sub1 g;
B1 s;
Super y;

```

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

- A. s = (B1) y;
- B. s = (B1) g;
- C. s = y;
- D. g = y;
- E. s = g;

10. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

11. Qual'è l'output di questo programma?

```

abstract class Figure {
    String type;
    int angles;

    protected Figure(String t, int a) {
        type = t;
        angles = a;
    }

    abstract void getArea();
}

class Triangle extends Figure {
    public Triangle(String t, int a) {
        super(t, a);
    }

    public void print() {
        System.out.print("Geometric figure with ");
        System.out.print(angles + " angles.");
    }

    public static void main(String[] args) {
        Triangle t = new Triangle("triangle" , 3);
        t.print();
    }
}

```

- A. Geometric figure with 0 angles.
- B. Geometric figure with 3 angles.
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]){
        Object[] o = new Object[100];
        System.out.println(o[10]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- 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 B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            if(true) throw( new MyExc1() );
            else throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( Exception z ) {
            System.out.print(5);
        }
    }
}

```

- A. 14... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. 143
- D. 1432
- E. Nessuna delle precedenti

14. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

```

```

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
 - B. XYZ
 - C. ZZZ
 - D. Errore a tempo di compilazione.
 - E. Nessuna delle precedenti.
-

15. Dire quale delle seguenti affermazioni è falsa:

- A. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
 - B. Una classe non può essere dichiarata final
 - C. Il minimo numero di elementi che un array può contenere è 0
 - D. In Java tutti i tipi numerici sono con segno
 - E. La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente
-

16. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv){
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception w ) {
        }
        finally {
            throw( new Exception() );
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new Exception() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
 - B. 1342
 - C. 134
 - D. 134Exception in thread "main" java.lang.Exception
 - 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

18-06-2015

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: