



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. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

2. Qual è il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
}
```

- A. `public A(String b) {}`
- B. `void A() {}`
- C. `private A(int i, String s) {}`
- D. `public Object A(int i, String s) {}`
- E. Il costruttore di una classe non può essere sovraccaricato.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
    }
}
```

```
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

4. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
```

```

String s8 = "";
s2 = s1.concat("abcd");
s6 = "";
a4 = a3;
q(s4, s6, s8, a4, a3);
}
void q(String s3, String s5, String s7,
float[] a1, Object a2){
if(s3 == s2) {
System.out.print(1);
} else {
System.out.print(0);
}
if(s5 == s7) {
System.out.print(1);
} else {
System.out.print(0);
}
if(a2 == a1) {
System.out.print(1);
} else {
System.out.print(0);
}
}
}
}

```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

5. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

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

```

public class Coffee {
String brand ="Nespresso";
int ml = 20;
Coffee() {
System.out.println(brand);
}
Coffee(int ml){

```

```

this.ml = ml;
System.out.println(ml);
}
public static void main(String args[]){
Coffee ristretto = new Coffee();
Coffee kazar = new Coffee();
Coffee vanillio = new Coffee(40);
ristretto = null;
vanillio = ristretto;
/* QUI */
return;
}
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
void m();
}

```

```

class X implements I {
String s = "X";
}

```

```

class Y extends X {
String s = "Y";

```

```

public void m() {
System.out.print(super.s
+ ((X) this).s
+ ((Y) this).s
+ this.s);
}

```

```

public static void main(String[] args){
new Y().m();
}
}

```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Date le dichiarazioni:

```

String [] t;
Object [] [] w;
Object [] z;

```

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

- A. `t = w;`
- B. `t = z;`
- C. `t = (String []) w;`
- D. `w = (Object [] []) t;`
- E. `w = (Object [] []) z;`

9. Date le dichiarazioni:

```
class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();
```

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

- A. `a = (Sub2) t;`
- B. `h = (C1) a;`
- C. `h = (C1) t;`
- D. `a = (Sub2) h;`
- E. Nessuno dei precedenti

10. Qual'è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv){
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
```

```
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}
```

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

11. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

12. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
```

```

    }
    catch( MyExc1 t ) {
        throw( new MyExc1() );
    }
    catch( MyExc2 s ) {
    }
}
}

```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print( constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print( constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";
}

```

```

public void m() {
    System.out.print(super.s
        + ((X) this).s
        + ((Y) this).s
        + this.s);
}

public static void main(String[] args){
    new Y().m();
}
}

```

- A. XXYY
- B. XYYY
- 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 A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {

```

```

}
catch( MyExc1 j ) {
}
catch( Exception e ) {
    System.out.print(3);
}
finally {
    System.out.print(4);
}
}
static void n() throws Exception {
    try {
        throw( new MyExc2() );
    }
    catch( MyExc1 t ) {
    }
    catch( Exception i ) {
    }
    catch( MyExc3 s ) {
        throw( new MyExc3() );
    }
    finally {
        System.out.print(5);
    }
}
}

```

- A. Errore a tempo di compilazione
  - B. 152
  - C. 1524
  - D. 154
  - 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 il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}
```

- A. XYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

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

- 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 A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

3. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. `public void f(String x)`
- B. `public byte f(Object y)`
- C. `protected int f(String z)`
- D. `int f(String s) throws Exception`
- E. Nessuna delle precedenti.

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}
```



```

class Run {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

5. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class Al {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

6. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

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

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti

7. Quale output si ottiene invocando il metodo p?

```

class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
        float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {

```

```

        System.out.print(0);
    }
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

8. Qual'è l'output di questo codice?

```

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

```

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

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

```

public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml){
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]){
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {

```

```

Building a = new Building(3, "Jones");
Building b = new Skyscraper(21, "Smith");
Skyscraper c = new Skyscraper(50, "Black");
a.print();
b.print();
c.print();
}
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

11. Qual'è l'output di questo codice?

```

class MyExcl extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExcl e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExcl() );
        }
        catch( MyExcl t ) {
            throw( new MyExcl() );
        }
        catch( MyExc2 s ) {
        }
    }
}

```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception

- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}

```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {

```

```

        System.out.print(floors + " "
            + constructor + " ");
    }
}

```

```

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

14. Date le dichiarazioni:

```

String [] t;
Object [] [] w;
Object [] z;

```

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

- A. t = w;
- B. t = z;

- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

15. Qualè il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

```

- A. public A(String b) {}
- B. void A() {}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

16. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null



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

**Prova n. 3**

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}
```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}
```

- A. XYYX
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}
```

```

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

```

```

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

```

```

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

```

```

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
    }
}

```

```

Skyscraper c = new Skyscraper(50, "Black");
a.print();
b.print();
c.print();
}
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

5. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

6. Qual'è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
}
static void q() {

```

```

try {
    System.out.print(3);
    throw( new MyExc3() );
}
finally {
    System.out.print(4);
    throw( new MyExc3() );
}
}
}

```

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

7. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}

```

- A. XXYY
- B. YYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Qual è il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
}
```

- A. public A(String b) {}
- B. void A(){}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

10. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {

        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }

    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
    }
}

```



```

        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

12. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

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

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti

13. Quale output si ottiene invocando il metodo p?

```

class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
        float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 100
  - B. 011
  - C. 111
  - D. 000
  - E. 101
- 

14. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
  - B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
  - C. In Java tutti i tipi numerici sono con segno
  - D. Un array possiede dei membri
  - E. Un array vuoto può avere riferimento null
- 

15. Date le dichiarazioni:

```
String [] t;  
Object [] [] w;  
Object [] z;
```

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

- A. `t = w;`
- B. `t = z;`
- C. `t = (String []) w;`
- D. `w = (Object [] []) t;`

E. `w = (Object [] []) z;`

---

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

```
public class Coffee {  
    String brand = "Nespresso";  
    int ml = 20;  
    Coffee() {  
        System.out.println(brand);  
    }  
    Coffee(int ml) {  
        this.ml = ml;  
        System.out.println(ml);  
    }  
    public static void main(String args[]) {  
        Coffee ristretto = new Coffee();  
        Coffee kazar = new Coffee();  
        Coffee vanillio = new Coffee(40);  
        ristretto = null;  
        vanillio = ristretto;  
        /* QUI */  
        return;  
    }  
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
-



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. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}
```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

4. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

5. Qual'è l'output di questo codice?

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

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }
}
```

```
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}
```

```

}
class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

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

```

public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

9. Qualè il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

```

- A. public A(String b) {}
- B. void A() {}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

10. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

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

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti

11. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {

```

```

try {
    throw( new MyExc2() );
}
catch( MyExc1 t ) {
}
catch( Exception i ) {
}
catch( MyExc3 s ) {
    throw( new MyExc3() );
}
finally {
    System.out.print(5);
}
}
}

```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}

```

- A. XXYY
- B. XYYY
- 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 A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new 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 A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}

```

- A. 635Exception in thread "main" java.lang.Exception
  - B. 63Exception in thread "main" java.lang.Exception
  - C. 615
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

15. Date le dichiarazioni:

```
String [] t;  
Object [] [] w;  
Object [] z;
```

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

- A. t = w;
  - B. t = z;
  - C. t = (String []) w;
  - D. w = (Object [] []) t;
  - E. w = (Object [] []) z;
- 

16. Quale output si ottiene invocando il metodo p?

```
class B {  
    private String s1 = "";  
    private String s2;  
    private float [] a3 = new float [8];  
    private float [] a4;  
    void p() {  
        String s4 = s1.concat("abcd");
```

```
String s6;  
String s8 = "";  
s2 = s1.concat("abcd");  
s6 = "";  
a4 = a3;  
q(s4, s6, s8, a4, a3);  
}  
void q(String s3, String s5, String s7,  
    float[] a1, Object a2){  
    if(s3 == s2) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
    if(s5 == s7) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
    if(a2 == a1) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
    }  
}
```

- A. 100
  - B. 011
  - C. 111
  - D. 000
  - E. 101
-





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 il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}
```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

4. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

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

```
public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

6. Qualè il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
}
```

- A. public A(String b) {}
- B. void A() {}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

7. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
        float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

8. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I().

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

E. Nessuna delle precedenti.

---

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

A. Jones Smith 50 Black

B. Jones 21 Smith 50 Black

C. Black Black 50 Black

D. Errore a tempo di compilazione.

E. Errore a tempo di esecuzione.

---

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

class X implements I {
```

```
    String s = "X";
}
```

```
class Y extends X {
    String s = "Y";
```

```
    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }
```

```
    public static void main(String[] args){
        new Y().m();
    }
}
```

A. XYY

B. XYYY

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

---

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}
```

```
class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
    }
}
```

```

        c.print ();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

12. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1 ();
a = new Sub2 ();
h = new C1 ();

```

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

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti

13. Date le dichiarazioni:

```

String [] t;
Object [] [] w;
Object [] z;

```

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

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

14. Qual'è l'output di questo codice?

```

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

```

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- 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 A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
    }
}

```

```

    }
    finally {
        System.out.print (5);
    }
}
static void p() throws Exception {
    try {
        System.out.print (6);
        if (true) throw( new MyExc3());
        throw( new MyExc1() );
    }
    catch( MyExc1 t ) {
        throw( new MyExc1() );
    }
    catch( MyExc2 s ) {
    }
}
}

```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- 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 Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print (1);
            n();

```

```

        System.out.print (2);
    }
    catch( MyExc2 r ) {
    }
    catch( MyExc1 j ) {
    }
    catch( Exception e ) {
        System.out.print (3);
    }
    finally {
        System.out.print (4);
    }
}
static void n() throws Exception {
    try {
        throw( new MyExc2() );
    }
    catch( MyExc1 t ) {
    }
    catch( Exception i ) {
    }
    catch( MyExc3 s ) {
        throw( new MyExc3() );
    }
    finally {
        System.out.print (5);
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- 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. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}
```

```
class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}
```

```
class Run {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

3. Date le dichiarazioni:

```
String [] t;
Object [] [] w;
Object [] z;
```

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

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?



```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args) {
        new Y().m();
    }
}

```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

5. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

---

6. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

7. Qual'è l'output di questo codice?

```

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

```

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

---

8. Qual è il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
}
```

- A. public A(String b) {}
- B. void A() {}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

---

9. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

10. Quale output si ottiene invocando il metodo p?

```

class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
        float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 100
- B. 011

- C. 111
- D. 000
- E. 101

11. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- 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 A1 {
    public static void main(String[] argv)

```

```

        throws Exception {
    try {
        p();
        System.out.print(1);
    }
    catch( MyExc1 e ) {
        System.out.print(2);
    }
    catch( MyExc3 d ) {
        System.out.print(3);
        throw( new Exception() );
    }
    catch( Exception i ) {
        System.out.print(4);
    }
    finally {
        System.out.print(5);
    }
}
static void p() throws Exception {
    try {
        System.out.print(6);
        if (true) throw( new MyExc3());
        throw( new MyExc1() );
    }
    catch( MyExc1 t ) {
        throw( new MyExc1() );
    }
    catch( MyExc2 s ) {
    }
}
}

```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper() {}

```

```

    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

14. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

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

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.

E. Errore a tempo di esecuzione.

---

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}

```

- A. XXYY
  - B. XYYY
  - 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. 7**

1. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- 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 A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
```

```
        throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}
```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

3. Date le dichiarazioni:

```
String [] t;
Object [] [] w;
Object [] z;
```

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

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

4. Qual'è il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
}
```

- A. public A(String b) {}
- B. void A() {}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

6. Qual'è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv){
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
```

```

        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}
```

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}
```

- A. XYYX
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

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

```
public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

---

9. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
```

```
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

---

10. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
        float[] a1, Object a2) {
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 100



- B. 011
- C. 111
- D. 000
- E. 101

```
Sub2 a;  
C1 h;  
A t;  
t = new C1 ();  
a = new Sub2 ();  
h = new C1 ();
```

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {  
    static String constructor;  
    static int floors;  
  
    Building(int f, String c) {  
        floors = f;  
        constructor = c;  
    }  
  
    public static void print() {  
        System.out.print(constructor + " ");  
    }  
}  
  
class Skyscraper extends Building {  
    Skyscraper(int f, String c){super(f,c);}  
  
    public static void print() {  
        System.out.print(floors + " "  
            + constructor + " ");  
    }  
}  
  
class Run {  
    public static void main(String[] args) {  
        Building a = new Building(3,"Jones");  
        Building b = new Skyscraper(21,"Smith");  
        Skyscraper c = new Skyscraper(50,"Black");  
        a.print();  
        b.print();  
        c.print();  
    }  
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

12. Date le dichiarazioni:

```
class C1 extends A {...}  
class A extends Object {...}  
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

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

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti

13. Dato un file contenente il seguente codice:

```
package pk;  
  
public class A {  
    public class I {  
    }  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

14. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}

```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;
}

```

```

Building(int f, String c) {
    floors = f;
    constructor = c;
}

public void print() {
    System.out.print(constructor + " ");
}

}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
-



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

```
public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
```

```
+ ((X) this).s
+ ((Y) this).s
+ this.s);
```

```
}

public static void main(String[] args) {
    new Y().m();
}
}
```

- A. XXYY
- B. XYYY
- 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 Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}
```

```
}  
}
```

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {  
    void m();  
}  
  
class X implements I {  
    String s = "X";  
}  
  
class Y extends X {  
    String s = "Y";  
  
    public void m() {  
        System.out.print(super.s  
            + ((X) this).s  
            + ((Y) this).s  
            + this.s);  
    }  
  
    public static void main(String[] args){  
        new Y().m();  
    }  
}
```

- A. XYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Date le dichiarazioni:

```
String [] t;  
Object [] [] w;  
Object [] z;
```

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

- A. t = w;
- B. t = z;
- C. t = (String []) w;

```
D. w = (Object [] []) t;  
E. w = (Object [] []) z;
```

6. Qual è il sovraccaricamento corretto del costruttore della seguente classe (overloading)?

```
public class A {  
    A(String s) {}  
}  
  
A. public A(String b) {}  
B. void A(){}  
C. private A(int i, String s) {}  
D. public Object A(int i, String s) {}  
E. Il costruttore di una classe non può essere sovraccaricato.
```

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {  
    static String constructor;  
    static int floors;  
  
    Building(int f, String c) {  
        floors = f;  
        constructor = c;  
    }  
  
    public static void print() {  
        System.out.print(constructor + " ");  
    }  
}  
  
class Skyscraper extends Building {  
    Skyscraper() {}  
    Skyscraper(int f, String c) {super(f,c);}  
  
    public static void print() {  
        System.out.print(floors + " "  
            + constructor + " ");  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Building a = new Building(3, "Jones");  
        Building b = new Skyscraper(21, "Smith");  
        Skyscraper c = new Skyscraper(50, "Black");  
        a.print();  
        b.print();  
        c.print();  
    }  
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black

- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

9. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
```

```
String s4 = s1.concat("abcd");
String s6;
String s8 = "";
s2 = s1.concat("abcd");
s6 = "";
a4 = a3;
q(s4, s6, s8, a4, a3);
}

void q(String s3, String s5, String s7,
    float[] a1, Object a2){
    if(s3 == s2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s5 == s7) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}
```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}
```

```

}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

11. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

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

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti

12. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception

E. Nessuna delle precedenti.

13. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

14. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- 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 A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {

```

```

        System.out.print(5);
    }
}
static void p() throws Exception {
    try {
        System.out.print(6);
        if (true) throw( new MyExc3());
        throw( new MyExc1() );
    }
    catch( MyExc1 t ) {
        throw( new MyExc1() );
    }
    catch( MyExc2 s ) {
    }
}
}

```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- 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 Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);

```

```

        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
static void n() throws Exception {
    try {
        throw( new MyExc2() );
    }
    catch( MyExc1 t ) {
    }
    catch( Exception i ) {
    }
    catch( MyExc3 s ) {
        throw( new MyExc3() );
    }
    finally {
        System.out.print(5);
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- 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. Date le dichiarazioni:

```
class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

e le inizializzazioni di variabile:

```
Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();
```

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

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti

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

```
public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}
```

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}
```

- A. 635Exception in thread "main" java.lang.Exception

- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

---

4. Date le dichiarazioni:

```
String [] t;  
Object [] [] w;  
Object [] z;
```

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

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

---

5. Qual'è l'output di questo codice?

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

- A. 34
- B. 342Exception in thread "main" MyExc3

---

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {  
    String constructor;  
    int floors;  
  
    Building(int f, String c) {  
        floors = f;  
        constructor = c;  
    }  
  
    public void print() {  
        System.out.print( constructor + " ");  
    }  
}  
  
class Skyscraper extends Building {  
    Skyscraper(int f, String c) {super(f,c);}  
  
    public void print() {  
        System.out.print( floors + " "  
            + constructor + " ");  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Building a = new Building(3, "Jones");  
        Building b = new Skyscraper(21, "Smith");  
        Skyscraper c = new Skyscraper(50, "Black");  
        a.print();  
        b.print();  
        c.print();  
    }  
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

7. Qual'è il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {  
    A(String s) {}  
}
```

- A. public A(String b) {}
- B. void A() {}
- C. private A(int i, String s) {}

D. public Object A(int i, String s) {}

E. Il costruttore di una classe non può essere sovraccaricato.

---

8. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

---

9. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
```

```
private float [] a3 = new float [8];
private float [] a4;
void p() {
    String s4 = s1.concat("abcd");
    String s6;
    String s8 = "";
    s2 = s1.concat("abcd");
    s6 = "";
    a4 = a3;
    q(s4, s6, s8, a4, a3);
}
void q(String s3, String s5, String s7,
    float[] a1, Object a2){
    if(s3 == s2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s5 == s7) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}
```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

---

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }
}
```

```

public static void main(String[] args){
    new Y().m();
}
}

```

- A. XYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}

```

- A. XYY
- B. XYYY
- 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 A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

14. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
- 

15. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
  - B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
  - C. In Java tutti i tipi numerici sono con segno
  - D. Un array possiede dei membri
  - E. Un array vuoto può avere riferimento null
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }
}
```

```
public static void print() {
    System.out.print(constructor + " ");
}

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
-



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. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

2. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. `public void f(String x)`
- B. `public byte f(Object y)`
- C. `protected int f(String z)`
- D. `int f(String s) throws Exception`
- E. Nessuna delle precedenti.

3. Qual è il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
}
```

- A. `public A(String b) {}`
- B. `void A() {}`
- C. `private A(int i, String s) {}`
- D. `public Object A(int i, String s) {}`
- E. Il costruttore di una classe non può essere sovraccaricato.

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?



```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args) {
        new Y().m();
    }
}

```

- A. XYY
- B. XYYY
- 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 MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
    }
}

```

```

        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

7. Date le dichiarazioni:

```

String [] t;
Object [] [] w;
Object [] z;

```

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

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

8. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {

```

```

        System.out.print(5);
    }
}
static void p() throws Exception {
    try {
        System.out.print(6);
        if (true) throw( new MyExc3());
        throw( new MyExc1() );
    }
    catch( MyExc1 t ) {
        throw( new MyExc1() );
    }
    catch( MyExc2 s ) {
    }
}
}

```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```

public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee(){
        System.out.println(brand);
    }
    Coffee(int ml){
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]){
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}

```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
    }
}

```

```

    Skyscraper c = new Skyscraper(50, "Black");
    a.print();
    b.print();
    c.print();
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

12. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

13. Quale output si ottiene invocando il metodo p?

```

class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
        float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {

```

```

        System.out.print(0);
    }
    if(s5 == s7) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

```

```

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

```

```

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

15. Qual'è l'output di questo codice?

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

```
}
}
```

- A. 34
  - B. 342Exception in thread "main" MyExc3
  - C. Errore a tempo di compilazione
  - D. 342
  - E. Nessuna delle precedenti
- 

16. Date le dichiarazioni:

```
class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();
```

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

- A. a = (Sub2) t;
  - B. h = (C1) a;
  - C. h = (C1) t;
  - D. a = (Sub2) h;
  - 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. 11**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
        float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;
```

2. Quale output si ottiene invocando il metodo p?

```

Building(int f, String c) {
    floors = f;
    constructor = c;
}

public static void print() {
    System.out.print(constructor + " ");
}
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

4. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
    }
}

```

```

finally {
    System.out.print(5);
}
}
static void p() throws Exception {
    try {
        System.out.print(6);
        if (true) throw( new MyExc3());
        throw( new MyExc1() );
    }
    catch( MyExc1 t ) {
        throw( new MyExc1() );
    }
    catch( MyExc2 s ) {
    }
}
}

```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

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

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti

6. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)

- C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
- 

7. Date le dichiarazioni:

```
String [] t;  
Object [] [] w;  
Object [] z;
```

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

- A. t = w;
  - B. t = z;
  - C. t = (String []) w;
  - D. w = (Object [] []) t;
  - E. w = (Object [] []) z;
- 

8. Dato un file contenente il seguente codice:

```
package pk;  
  
public class A {  
    public class I {  
    }  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
  - 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 A.I().
  - D. Un'istanza della classe I può essere costruita con new A().new I().
  - E. Nessuna delle precedenti.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {  
    void m();  
}  
  
class X implements I {  
    String s = "X";  
}  
  
class Y extends X {
```

```
String s = "Y";  
  
public void m() {  
    System.out.print(super.s  
        + ((X) this).s  
        + ((Y) this).s  
        + this.s);  
}  
  
public static void main(String[] args){  
    new Y().m();  
}  
}
```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {  
    void m();  
}  
  
abstract class X implements I {  
    String s = "X";  
}  
  
class Y extends X {  
    String s = "Y";  
  
    public void m() {  
        System.out.print(super.s  
            + ((X) this).s  
            + ((Y) this).s  
            + this.s);  
    }  
  
    public static void main(String[] args){  
        new Y().m();  
    }  
}
```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?



```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print (constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print (floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

12. Qualè il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

```

- A. public A(String b) {}
- B. void A(){}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

13. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

14. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

15. Qual'è l'output di questo codice?

```

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

```

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342

E. Nessuna delle precedenti

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

```

public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
-



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 codice?

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

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;
```

```
Building(int f, String c) {
    floors = f;
    constructor = c;
}

public static void print() {
    System.out.print(constructor + " ");
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

3. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

4. Qual è il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
}
```

- A. `public A(String b) {}`
- B. `void A() {}`
- C. `private A(int i, String s) {}`
- D. `public Object A(int i, String s) {}`
- E. Il costruttore di una classe non può essere sovraccaricato.

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

```
public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}
```

```
class X implements I {
    String s = "X";
}
```

```
class Y extends X {
    String s = "Y";
```

```
    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }
```

```
    public static void main(String[] args) {
        new Y().m();
    }
}
```

- A. XYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
        float[] a1, Object a2) {
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
```

```

    } else {
        System.out.print(0);
    }
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

8. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3() );
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}

```

- A. 635Exception in thread "main" java.lang.Exception

- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- 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 A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

10. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione

- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

11. Date le dichiarazioni:

```
class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();
```

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

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}
```

```
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

13. Date le dichiarazioni:

```
String [] t;
Object [] [] w;
Object [] z;
```

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

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}
```

```

    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";
}

```

```

public void m() {
    System.out.print(super.s
        + ((X) this).s
        + ((Y) this).s
        + this.s);
}

public static void main(String[] args) {
    new Y().m();
}

```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

16. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws 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. 13**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}
```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}
```

```
}

}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

3. Qual è il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
}

A. public A(String b) {}
B. void A(){}
C. private A(int i, String s) {}
D. public Object A(int i, String s) {}
E. Il costruttore di una classe non può essere sovraccaricato.
```

4. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
        float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

7. Date le dichiarazioni:

```
String [] t;
Object [] [] w;
Object [] z;
```

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

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

6. Quale output si ottiene invocando il metodo p?

8. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- 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 A1 {
    public static void main(String[] argv)
        throws Exception {

        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

10. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

11. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

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

- A. `a = (Sub2) t;`
- B. `h = (C1) a;`
- C. `h = (C1) t;`
- D. `a = (Sub2) h;`
- E. Nessuno dei precedenti

12. Qual'è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv){
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {

```

```

    }
    catch( MyExc2 c ) {
        System.out.print(1);
    }
    finally {
        System.out.print(2);
    }
}
static void q() {
    try {
        System.out.print(3);
        throw( new MyExc3() );
    }
    finally {
        System.out.print(4);
        throw( new MyExc3() );
    }
}
}

```

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print( constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print( floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
    }
}

```

```

        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print( super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}

```

- A. XXYY
- B. XYYY
- 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 A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {

```

```

        System.out.print (2);
    }
    catch( MyExc3 d ) {
        System.out.print (3);
        throw( new Exception() );
    }
    catch( Exception i ) {
        System.out.print (4);
    }
    finally {
        System.out.print (5);
    }
}
static void p() throws Exception {
    try {
        System.out.print (6);
        if (true) throw( new MyExc3());
        throw( new MyExc1() );
    }
    catch( MyExc1 t ) {
        throw( new MyExc1() );
    }
    catch( MyExc2 s ) {
    }
}
}

```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

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

```

public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
-



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

**Prova n. 14**

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

2. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
        float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

3. Date le dichiarazioni:

```
class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}
```



e le inizializzazioni di variabile:

```
Sub2 a;  
C1 h;  
A t;  
t = new C1();  
a = new Sub2();  
h = new C1();
```

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

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti

---

4. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

---

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {  
    void m();  
}  
  
class X implements I {  
    String s = "X";  
}  
  
class Y extends X {  
    String s = "Y";  
  
    public void m() {  
        System.out.print(super.s  
            + ((X) this).s  
            + ((Y) this).s  
            + this.s);  
    }  
  
    public static void main(String[] args){  
        new Y().m();  
    }  
}
```

- A. XYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

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

```
public class Coffee {  
    String brand = "Nespresso";  
    int ml = 20;  
    Coffee() {  
        System.out.println(brand);  
    }  
    Coffee(int ml) {  
        this.ml = ml;  
        System.out.println(ml);  
    }  
    public static void main(String args[]) {  
        Coffee ristretto = new Coffee();  
        Coffee kazar = new Coffee();  
        Coffee vanillio = new Coffee(40);  
        ristretto = null;  
        vanillio = ristretto;  
        /* QUI */  
        return;  
    }  
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

---

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {  
    static String constructor;  
    static int floors;  
  
    Building(int f, String c) {  
        floors = f;  
        constructor = c;  
    }  
  
    public static void print() {  
        System.out.print(constructor + " ");  
    }  
}  
  
class Skyscraper extends Building {  
    Skyscraper() {}  
}
```

```

Skyscraper(int f, String c) {super(f,c);}

public static void print() {
    System.out.print(floors + " "
        + constructor + " ");
}
}

```

```

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

8. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

9. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno

- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

```

```

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

```

```

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

11. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
        }
    }
}

```

```

        System.out.print(1);
    }
    catch( MyExc1 e ) {
        System.out.print(2);
    }
    catch( MyExc3 d ) {
        System.out.print(3);
        throw( new Exception() );
    }
    catch( Exception i ) {
        System.out.print(4);
    }
    finally {
        System.out.print(5);
    }
}
static void p() throws Exception {
    try {
        System.out.print(6);
        if (true) throw( new MyExc3());
        throw( new MyExc1() );
    }
    catch( MyExc1 t ) {
        throw( new MyExc1() );
    }
    catch( MyExc2 s ) {
    }
}
}

```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. Date le dichiarazioni:

```

String [] t;
Object [] [] w;
Object [] z;

```

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

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

13. Qual'è l'output di questo codice?

```

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

```

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }
}

```

```

public static void main(String[] args){
    new Y().m();
}
}

```

- A. XYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

15. Qual è il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

```

- A. public A(String b) {}
- B. void A(){}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {

```

```

        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

```

```

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

```

```

class Run {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.



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. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
```

```
        }
    }
}
```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

3. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

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

```
public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
```

```

    Coffee kazar = new Coffee();
    Coffee vanillio = new Coffee(40);
    ristretto = null;
    vanillio = ristretto;
    /* QUI */
    return;
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}

```

- A. XXYY
- B. XYYY
- 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 Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv){

```

```

    try {
        q();
    }
    catch( MyExc3 x ) {
    }
    catch( MyExc1 c ) {
    }
    catch( MyExc2 c ) {
        System.out.print(1);
    }
    finally {
        System.out.print(2);
    }
}

static void q() {
    try {
        System.out.print(3);
        throw( new MyExc3() );
    }
    finally {
        System.out.print(4);
        throw( new MyExc3() );
    }
}
}

```

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

7. Qual'è il sovraccaricamento corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

```

- A. public A(String b) {}
- B. void A() {}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

8. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);

```

```

    }
    catch( MyExc2 r ) {
    }
    catch( MyExc1 j ) {
    }
    catch( Exception e ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
    }
}
static void n() throws Exception {
    try {
        throw( new MyExc2() );
    }
    catch( MyExc1 t ) {
    }
    catch( Exception i ) {
    }
    catch( MyExc3 s ) {
        throw( new MyExc3() );
    }
    finally {
        System.out.print(5);
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print( constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print( floors + " "
            + constructor + " ");
    }
}

```

```

    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

10. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

11. Quale output si ottiene invocando il metodo p?

```

class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
        float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {

```



```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black

- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}

```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

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

- A. a = (Sub2) t;
- B. h = (C1) a;

- C. h = (C1) t;
  - D. a = (Sub2) h;
  - E. Nessuno dei precedenti
- 

15. Date le dichiarazioni:

```
String [] t;  
Object [] [] w;  
Object [] z;
```

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

- A. t = w;
  - B. t = z;
  - C. t = (String []) w;
  - D. w = (Object [] []) t;
  - E. w = (Object [] []) z;
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {  
    static String constructor;  
    static int floors;  
  
    Building(int f, String c) {  
        floors = f;  
        constructor = c;  
    }  
}
```

```
public static void print () {  
    System.out.print (constructor + " ");  
}
```

```
class Skyscraper extends Building {  
    Skyscraper(int f, String c){super(f,c);}  
  
    public static void print () {  
        System.out.print (floors + " "  
            + constructor + " ");  
    }  
}
```

```
class Run {  
    public static void main(String[] args) {  
        Building a = new Building(3, "Jones");  
        Building b = new Skyscraper(21, "Smith");  
        Skyscraper c = new Skyscraper(50, "Black");  
        a.print();  
        b.print();  
        c.print();  
    }  
}
```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
-



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. Qualè il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
}
```

- A. `public A(String b) {}`
- B. `void A(){}`
- C. `private A(int i, String s) {}`
- D. `public Object A(int i, String s) {}`
- E. Il costruttore di una classe non può essere sovraccaricato.

2. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
    }
}
```

```
}
finally {
    System.out.print(4);
}
}
static void n() throws Exception {
    try {
        throw( new MyExc2() );
    }
    catch( MyExc1 t ) {
    }
    catch( Exception i ) {
    }
    catch( MyExc3 s ) {
        throw( new MyExc3() );
    }
    finally {
        System.out.print(5);
    }
}
}
```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

4. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I()`.

- D. Un'istanza della classe l può essere costruita con `new A().new l()`.
- E. Nessuna delle precedenti.

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

5. Date le dichiarazioni:

```
String [] t;  
Object [] [] w;  
Object [] z;
```

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

- A. `t = w;`
- B. `t = z;`
- C. `t = (String []) w;`
- D. `w = (Object [] []) t;`
- E. `w = (Object [] []) z;`

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {  
    String constructor;  
    int floors;  
  
    Building(int f, String c) {  
        floors = f;  
        constructor = c;  
    }  
  
    public void print () {  
        System.out.print (constructor + " ");  
    }  
}  
  
class Skyscraper extends Building {  
    Skyscraper(int f, String c) {super(f,c);}  
  
    public void print () {  
        System.out.print (floors + " "  
            + constructor + " ");  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Building a = new Building(3, "Jones");  
        Building b = new Skyscraper (21, "Smith");  
        Skyscraper c = new Skyscraper (50, "Black");  
        a.print ();  
        b.print ();  
        c.print ();  
    }  
}
```

7. Quale output si ottiene invocando il metodo p?

```
class B {  
    private String s1 = "";  
    private String s2;  
    private float [] a3 = new float [8];  
    private float [] a4;  
    void p() {  
        String s4 = s1.concat("abcd");  
        String s6;  
        String s8 = "";  
        s2 = s1.concat("abcd");  
        s6 = "";  
        a4 = a3;  
        q(s4, s6, s8, a4, a3);  
    }  
    void q(String s3, String s5, String s7,  
        float[] a1, Object a2){  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s5 == s7) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(a2 == a1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {  
    static String constructor;  
    static int floors;  
  
    Building(int f, String c) {  
        floors = f;
```

```

        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

9. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

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

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;

- D. a = (Sub2) h;
- E. Nessuno dei precedenti

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

```

public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args) {
        new Y().m();
    }
}

```

- A. XYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

12. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

14. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}
```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}

```

- A. XXYY
- B. XYYY
- 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 Error { }
class MyExc2 extends Error { }

```

```

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

```

- A. 34
  - B. 342Exception in thread "main" MyExc3
  - C. Errore a tempo di compilazione
  - D. 342
  - 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 Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv){
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}
```

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}
```

```
abstract class X implements I {
    String s = "X";
}
class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}
```

- A. XXYY
- B. XYYY
- 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 A1 {
    public static void main(String[] argv)
        throws Exception {

        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
    }
}
```

```

    }
    finally {
        System.out.print (5);
    }
}
static void p() throws Exception {
    try {
        System.out.print (6);
        if (true) throw( new MyExc3 ());
        throw( new MyExc1 ( ));
    }
    catch( MyExc1 t ) {
        throw( new MyExc1 ( ));
    }
    catch( MyExc2 s ) {
    }
}
}

```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. Qualè il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

```

- A. public A(String b) {}
- B. void A(){}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

5. Date le dichiarazioni:

```

String [] t;
Object [] [] w;
Object [] z;

```

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

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print (constructor + " ");
    }
}

```

```

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print (floors + " "
            + constructor + " ");
    }
}

```

```

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

7. Quale output si ottiene invocando il metodo p?

```

class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
    }
}

```

```

    a4 = a3;
    q(s4, s6, s8, a4, a3);
}
void q(String s3, String s5, String s7,
    float[] a1, Object a2){
    if(s3 == s2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s5 == s7) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}

```

- A. XXYY
- B. YYYY

- 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 A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

10. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

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

- A. `a = (Sub2) t;`
- B. `h = (C1) a;`
- C. `h = (C1) t;`
- D. `a = (Sub2) h;`
- E. Nessuno dei precedenti

11. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}
```

```
class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}
```

```
class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

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

```
public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
```

```
Coffee() {
    System.out.println(brand);
}
Coffee(int ml) {
    this.ml = ml;
    System.out.println(ml);
}
public static void main(String args[]) {
    Coffee ristretto = new Coffee();
    Coffee kazar = new Coffee();
    Coffee vanillio = new Coffee(40);
    ristretto = null;
    vanillio = ristretto;
    /* QUI */
    return;
}
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

14. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {

        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
    }
}
```

```

    }
    finally {
        System.out.print (5);
    }
}
}

```

- A. Errore a tempo di compilazione
  - B. 152
  - C. 1524
  - D. 154
  - E. Nessuna delle precedenti
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print () {
        System.out.print (constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper () {}
    Skyscraper(int f, String c) {super (f,c);}

    public static void print () {

```

```

        System.out.print (floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building (3, "Jones");
        Building b = new Skyscraper (21, "Smith");
        Skyscraper c = new Skyscraper (50, "Black");
        a.print ();
        b.print ();
        c.print ();
    }
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

16. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
  - B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
  - C. In Java tutti i tipi numerici sono con segno
  - D. Un array possiede dei membri
  - E. Un array vuoto può avere riferimento null
-



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

**Prova n. 18**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}
```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}
```

```

    }
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

3. Date le dichiarazioni:

```
String [] t;
Object [] [] w;
Object [] z;
```

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

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;



4. Date le dichiarazioni:

```
class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();
```

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

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args) {
        new Y().m();
    }
}

```

- A. XYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

```

s6 = "";
a4 = a3;
q(s4, s6, s8, a4, a3);
}

void q(String s3, String s5, String s7,
    float[] a1, Object a2) {
    if(s3 == s2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s5 == s7) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

8. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

9. Quale output si ottiene invocando il metodo p?

```

class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
    }
}

```

10. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {

```

```

        System.out.print(6);
        if (true) throw( new MyExc3());
        throw( new MyExc1() );
    }
    catch( MyExc1 t ) {
        throw( new MyExc1() );
    }
    catch( MyExc2 s ) {
    }
}
}

```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

11. Qual'è l'output di questo codice?

```

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

```

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

12. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

13. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

14. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {

```

```

try {
    throw( new MyExc2() );
}
catch( MyExc1 t ) {
}
catch( Exception i ) {
}
catch( MyExc3 s ) {
    throw( new MyExc3() );
}
finally {
    System.out.print(5);
}
}
}

```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

15. Qualè il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

```

- A. public A(String b) {}
- B. void A(){}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}

E. Il costruttore di una classe non può essere sovraccaricato.

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

```

public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6



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. Qualè il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
}
```

- A. public A(String b) {}
- B. void A(){}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

2. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

3. Date le dichiarazioni:

```
String [] t;
Object [] [] w;
Object [] z;
```

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

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

4. Qual'è l'output di questo codice?

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

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}
```

```
class X implements I {
```

```

String s = "X";
}
class Y extends X {
String s = "Y";

public void m() {
System.out.print(super.s
+ ((X) this).s
+ ((Y) this).s
+ this.s);
}

public static void main(String[] args){
new Y().m();
}
}

```

- A. XYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

6. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

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

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti

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

```

public class Coffee {
String brand = "Nespresso";
int ml = 20;
Coffee() {

```

```

System.out.println(brand);
}
Coffee(int ml) {
this.ml = ml;
System.out.println(ml);
}
public static void main(String args[]) {
Coffee ristretto = new Coffee();
Coffee kazar = new Coffee();
Coffee vanillio = new Coffee(40);
ristretto = null;
vanillio = ristretto;
/* QUI */
return;
}
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

8. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
public static void main(String[] argv)
throws Exception {

try {
System.out.print(1);
n();
System.out.print(2);
}
catch( MyExc2 r ) {
}
catch( MyExc1 j ) {
}
catch( Exception e ) {
System.out.print(3);
}
finally {
System.out.print(4);
}
}
static void n() throws Exception {
try {
throw( new MyExc2() );
}
catch( MyExc1 t ) {
}
catch( Exception i ) {
}
catch( MyExc3 s ) {
throw( new MyExc3() );
}
}
}

```

```

        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

9. Quale output si ottiene invocando il metodo p?

```

class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
        float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}

```

- A. XYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {

```



```

Building a = new Building(3, "Jones");
Building b = new Skyscraper(21, "Smith");
Skyscraper c = new Skyscraper(50, "Black");
a.print();
b.print();
c.print();
}
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.

E. Errore a tempo di esecuzione.

13. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

14. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. `public void f(String x)`
- B. `public byte f(Object y)`
- C. `protected int f(String z)`
- D. `int f(String s) throws Exception`
- E. Nessuna delle precedenti.

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}
}

```

```

public static void print () {
    System.out.print (floors + " "
        + constructor + " ");
}
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper (21, "Smith");
        Skyscraper c = new Skyscraper (50, "Black");
        a.print ();
        b.print ();
        c.print ();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

16. Qual'è l'output di questo codice?

```

class MyExcl extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p ();
            System.out.print (1);

```

```

}
catch( MyExcl e ) {
    System.out.print (2);
}
catch( MyExc3 d ) {
    System.out.print (3);
    throw( new Exception() );
}
catch( Exception i ) {
    System.out.print (4);
}
finally {
    System.out.print (5);
}
}
static void p() throws Exception {
    try {
        System.out.print (6);
        if (true) throw( new MyExc3());
        throw( new MyExcl() );
    }
    catch( MyExcl t ) {
        throw( new MyExcl() );
    }
    catch( MyExc2 s ) {
    }
}
}

```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- 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. 20**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

```
public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

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

3. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
        float[] a1, Object a2){
        if(s3 == s2) {
```

```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
if(s5 == s7) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(a2 == a1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
}

```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

4. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
    }
}

```

```

        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- 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 A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}

```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}
```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Date le dichiarazioni:

```
String [] t;
Object [] [] w;
Object [] z;
```

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

- A. `t = w;`
- B. `t = z;`
- C. `t = (String []) w;`
- D. `w = (Object [] []) t;`
- E. `w = (Object [] []) z;`

8. Qual'è l'output di questo codice?

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

```
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}
```

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

9. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. `public void f(String x)`
- B. `public byte f(Object y)`
- C. `protected int f(String z)`
- D. `int f(String s) throws Exception`
- E. Nessuna delle precedenti.

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }
}
```

```

public void print() {
    System.out.print (constructor + " ");
}
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print (floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper (21, "Smith");
        Skyscraper c = new Skyscraper (50, "Black");
        a.print ();
        b.print ();
        c.print ();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

11. Qualè il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

```

- A. public A(String b) {}
- B. void A() {}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

```

```

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print (super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}

```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print (constructor + " ");
    }
}

```

```

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print (floors + " "
            + constructor + " ");
    }
}

```

```

class Run {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper (21, "Smith");
        Skyscraper c = new Skyscraper (50, "Black");
        a.print ();
        b.print ();
        c.print ();
    }
}

```

- A. Jones Smith 50 Black

- B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

14. Date le dichiarazioni:

```
class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();
```

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

- A. a = (Sub2) t;
  - B. h = (C1) a;
  - C. h = (C1) t;
  - D. a = (Sub2) h;
  - E. Nessuno dei precedenti
- 

15. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
  - B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
  - C. In Java tutti i tipi numerici sono con segno
  - D. Un array possiede dei membri
  - E. Un array vuoto può avere riferimento null
- 

16. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
  - 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 A.I()`.
  - D. Un'istanza della classe I può essere costruita con `new A().new 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. 21**

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}
```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- 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 A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
    }
}
```

```

    }
    catch( MyExc1 t ) {
    }
    catch( Exception i ) {
    }
    catch( MyExc3 s ) {
        throw( new MyExc3() );
    }
    finally {
        System.out.print(5);
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print( constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black

- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

5. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

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

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti

6. Qual'è l'output di questo codice?

```

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

```

```

        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}

```

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print( constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print( constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

9. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

10. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

11. Qual è il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {  
    A(String s) {}  
}
```

- A. public A(String b) {}
- B. void A() {}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

12. Quale output si ottiene invocando il metodo p?

```
class B {  
    private String s1 = "";  
    private String s2;  
    private float [] a3 = new float [8];  
    private float [] a4;  
    void p() {  
        String s4 = s1.concat("abcd");  
        String s6;  
        String s8 = "";  
        s2 = s1.concat("abcd");  
        s6 = "";  
        a4 = a3;  
        q(s4, s6, s8, a4, a3);  
    }  
    void q(String s3, String s5, String s7,  
        float[] a1, Object a2){  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s5 == s7) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(a2 == a1) {  
            System.out.print(1);  
        }  
    }  
}
```

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

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

13. Date le dichiarazioni:

```
String [] t;  
Object [] [] w;  
Object [] z;
```

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

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {  
    void m();  
}  
  
abstract class X implements I {  
    String s = "X";  
}  
  
class Y extends X {  
    String s = "Y";  
  
    public void m() {  
        System.out.print(super.s  
            + ((X) this).s  
            + ((Y) this).s  
            + this.s);  
    }  
  
    public static void main(String[] args){  
        new Y().m();  
    }  
}
```

- A. XYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

---

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

```
public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args) {
        new Y().m();
    }
}
```

- A. XXYY
  - B. XYYY
  - 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. 22**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args) {
        new Y().m();
    }
}
```

- A. XYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Date le dichiarazioni:

```
class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();
```

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

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti

3. Qual è il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
}
```

- A. public A(String b) {}
- B. void A() {}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

4. Date le dichiarazioni:

```
String [] t;
Object [] [] w;
Object [] z;
```

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

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?



```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print (constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print (floors + " "
            + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper (21, "Smith");
        Skyscraper c = new Skyscraper (50, "Black");
        a.print ();
        b.print ();
        c.print ();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

6. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p ();
            System.out.print (1);
        }
        catch( MyExc1 e ) {
            System.out.print (2);
        }
        catch( MyExc3 d ) {
            System.out.print (3);
            throw( new Exception() );
        }
    }
}

```

```

}
catch( Exception i ) {
    System.out.print (4);
}
finally {
    System.out.print (5);
}
}
static void p() throws Exception {
    try {
        System.out.print (6);
        if (true) throw( new MyExc3());
        throw( new MyExc1() );
    }
    catch( MyExc1 t ) {
        throw( new MyExc1() );
    }
    catch( MyExc2 s ) {
    }
}
}

```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

8. Qual'è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv){
        try {
            q ();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print (1);
        }
    }
}

```

```

    }
    finally {
        System.out.print(2);
    }
}
static void q() {
    try {
        System.out.print(3);
        throw( new MyExc3() );
    }
    finally {
        System.out.print(4);
        throw( new MyExc3() );
    }
}
}

```

- A. 34
  - B. 342Exception in thread "main" MyExc3
  - C. Errore a tempo di compilazione
  - D. 342
  - E. Nessuna delle precedenti
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}

```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

10. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
  - 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 A.I()`.
  - D. Un'istanza della classe I può essere costruita con `new A().new I()`.
  - E. Nessuna delle precedenti.
- 

11. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
  - B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
  - C. In Java tutti i tipi numerici sono con segno
  - D. Un array possiede dei membri
  - E. Un array vuoto può avere riferimento null
- 

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

```

public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- A. 1
- B. 2

- C. 3
- D. 4
- E. 6

13. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

15. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
        float[] a1, Object a2){
```

```

if (s3 == s2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if (s5 == s7) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if (a2 == a1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
}

```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }
}

```

```

}

public static void print() {
    System.out.print(constructor + " ");
}

}

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.



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. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}
```

```
class Run {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

3. Date le dichiarazioni:

```
class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();
```

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

- A. `a = (Sub2) t;`
- B. `h = (C1) a;`
- C. `h = (C1) t;`
- D. `a = (Sub2) h;`
- E. Nessuno dei precedenti

4. Quale output si ottiene invocando il metodo p?

```

class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
        float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

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

```

public class Coffee {
    String brand ="Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml){
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]){
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
    }
}

```

```

        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}

```

- A. XYYX
- B. XYYY
- 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 Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv){
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
    }
}

```

```

    catch( MyExc1 c ) {
    }
    catch( MyExc2 c ) {
        System.out.print(1);
    }
    finally {
        System.out.print(2);
    }
}
static void q() {
    try {
        System.out.print(3);
        throw( new MyExc3() );
    }
    finally {
        System.out.print(4);
        throw( new MyExc3() );
    }
}
}

```

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print( constructor + " ");
    }
}

```

```

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print( floors + " "
            + constructor + " ");
    }
}

```

```

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
    }
}

```

```

        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

```

```

abstract class X implements I {
    String s = "X";
}

```

```

class Y extends X {
    String s = "Y";
}

```

```

    public void m() {
        System.out.print( super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }
}

```

```

    public static void main(String[] args){
        new Y().m();
    }
}

```

- A. XYYX
- B. XYYY
- 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 MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
    }
}

```



```

    catch( MyExc2 r ) {
    }
    catch( MyExc1 j ) {
    }
    catch( Exception e ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
    }
}
static void n() throws Exception {
    try {
        throw( new MyExc2() );
    }
    catch( MyExc1 t ) {
    }
    catch( Exception i ) {
    }
    catch( MyExc3 s ) {
        throw( new MyExc3() );
    }
    finally {
        System.out.print(5);
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

11. Qualè il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

```

- A. public A(String b) {}
- B. void A(){}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;
}

```

```

Building(int f, String c) {
    floors = f;
    constructor = c;
}

public static void print() {
    System.out.print(constructor + " ");
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

13. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

14. Date le dichiarazioni:

```

String [] t;
Object [] [] w;
Object [] z;

```

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

- A. t = w;

- B. `t = z;`
- C. `t = (String []) w;`
- D. `w = (Object [] []) t;`
- E. `w = (Object [] []) z;`

---

15. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. `public void f(String x)`
- B. `public byte f(Object y)`
- C. `protected int f(String z)`
- D. `int f(String s) throws Exception`
- 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 A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
```

```
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}
```

- A. 635Exception in thread "main" java.lang.Exception
  - B. 63Exception in thread "main" java.lang.Exception
  - C. 615
  - 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. 24**

1. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

```
}  
}
```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

2. Quale output si ottiene invocando il metodo p?

```
class B {  
    private String s1 = "";  
    private String s2;  
    private float [] a3 = new float [8];  
    private float [] a4;  
    void p() {  
        String s4 = s1.concat("abcd");  
        String s6;  
        String s8 = "";  
        s2 = s1.concat("abcd");  
        s6 = "";  
        a4 = a3;  
        q(s4, s6, s8, a4, a3);  
    }  
    void q(String s3, String s5, String s7,  
        float[] a1, Object a2){  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s5 == s7) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(a2 == a1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

3. Date le dichiarazioni:

```
String [] t;  
Object [] [] w;  
Object [] z;
```

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

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {  
    void m();  
}  
  
class X implements I {  
    String s = "X";  
}  
  
class Y extends X {  
    String s = "Y";  
  
    public void m() {  
        System.out.print(super.s  
            + ((X) this).s  
            + ((Y) this).s  
    }  
}
```

```

        + this.s);
    }

    public static void main(String[] args) {
        new Y().m();
    }
}

```

- A. XYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Qual è il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

```

- A. public A(String b) {}
- B. void A() {}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

```

```

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}

```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

8. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}
```

- A. XXYY
- B. XYYY

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

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

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

```
public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
```

```

        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]){
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

12. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class Al {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

```

    }
}

```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

13. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

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

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti

14. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }
}

```

```

    public static void print() {
        System.out.print( constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

16. Qual'è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }

```

```

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

```

- A. 34
  - B. 342Exception in thread "main" MyExc3
  - C. Errore a tempo di compilazione
  - D. 342
  - 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 codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

```

```

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}

```

- A. XYYX
- B. XYYY
- 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 A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}
```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

7. Qual'è il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
}
```

- A. public A(String b) {}
- B. void A() {}
- C. private A(int i, String s) {}

- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

---

8. Date le dichiarazioni:

```
class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();
```

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

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti

---

9. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

---

10. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I().

- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

- B. 011  
C. 111  
D. 000  
E. 101

---

11. Date le dichiarazioni:

```
String [] t;  
Object [] [] w;  
Object [] z;
```

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

- A. `t = w;`  
B. `t = z;`  
C. `t = (String []) w;`  
D. `w = (Object [] []) t;`  
E. `w = (Object [] []) z;`

---

12. Quale output si ottiene invocando il metodo p?

```
class B {  
    private String s1 = "";  
    private String s2;  
    private float [] a3 = new float [8];  
    private float [] a4;  
    void p() {  
        String s4 = s1.concat("abcd");  
        String s6;  
        String s8 = "";  
        s2 = s1.concat("abcd");  
        s6 = "";  
        a4 = a3;  
        q(s4, s6, s8, a4, a3);  
    }  
    void q(String s3, String s5, String s7,  
        float[] a1, Object a2){  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s5 == s7) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(a2 == a1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 100

---

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {  
    void m();  
}  
  
class X implements I {  
    String s = "X";  
}  
  
class Y extends X {  
    String s = "Y";  
  
    public void m() {  
        System.out.print(super.s  
            + ((X) this).s  
            + ((Y) this).s  
            + this.s);  
    }  
  
    public static void main(String[] args){  
        new Y().m();  
    }  
}
```

- A. XXYY  
B. XYYY  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.

---

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

```
public class Coffee {  
    String brand = "Nespresso";  
    int ml = 20;  
    Coffee() {  
        System.out.println(brand);  
    }  
    Coffee(int ml) {  
        this.ml = ml;  
        System.out.println(ml);  
    }  
    public static void main(String args[]) {  
        Coffee ristretto = new Coffee();  
        Coffee kazar = new Coffee();  
        Coffee vanillio = new Coffee(40);  
    }  
}
```

```

    ristretto = null;
    vanillio = ristretto;
    /* QUI */
    return;
}
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

---

15. Qual'è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv){
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
}

```

```

static void q() {
    try {
        System.out.print(3);
        throw( new MyExc3() );
    }
    finally {
        System.out.print(4);
        throw( new MyExc3() );
    }
}
}

```

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

---

16. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws 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. 26**

1. Date le dichiarazioni:

```
String [] t;
Object [] [] w;
Object [] z;
```

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

- A. `t = w;`
- B. `t = z;`
- C. `t = (String []) w;`
- D. `w = (Object [] []) t;`
- E. `w = (Object [] []) z;`

2. Qual'è l'output di questo codice?

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

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

3. Qual'è il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
}
```

- A. `public A(String b) {}`
- B. `void A(){}`
- C. `private A(int i, String s) {}`
- D. `public Object A(int i, String s) {}`
- E. Il costruttore di una classe non può essere sovraccaricato.

4. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
        float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
```



```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black

- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}

```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}
}

```

- A. Jones Smith 50 Black

```

public void print() {
    System.out.print(floors + " "
        + constructor + " ");
}
}

```

```

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

8. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
    }
}

```

```

}
catch( MyExc2 s ) {
}
}
}

```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

9. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- 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 A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

```

static void n() throws Exception {
    try {
        throw( new MyExc2() );
    }
    catch( MyExc1 t ) {
    }
    catch( Exception i ) {
    }
    catch( MyExc3 s ) {
        throw( new MyExc3() );
    }
    finally {
        System.out.print(5);
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

11. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

12. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

13. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

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

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti

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

```

public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}

```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;
}

```

```

Building(int f, String c) {
    floors = f;
    constructor = c;
}

public static void print() {
    System.out.print(constructor + " ");
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
-



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. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

2. Date le dichiarazioni:

```
String [] t;
Object [] [] w;
Object [] z;
```

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

- A. `t = w;`
- B. `t = z;`
- C. `t = (String []) w;`
- D. `w = (Object [] []) t;`
- E. `w = (Object [] []) z;`

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}
```

```
class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}
```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
        float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
```

```

        System.out.print(0);
    }
    if(s5 == s7) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

5. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

```

```

    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

- B. 2
- C. 3
- D. 4
- E. 6

8. Date le dichiarazioni:

```
class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();
```

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

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti

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

```
public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}
```

- A. 1

10. Qual è il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
}
```

- A. public A(String b) {}
- B. void A() {}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black



- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

12. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

13. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}
```

- A. XXYY
- B. XYYY
- 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 Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv){
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
    }
}
```

```

    }
    finally {
        System.out.print(2);
    }
}
static void q() {
    try {
        System.out.print(3);
        throw( new MyExc3() );
    }
    finally {
        System.out.print(4);
        throw( new MyExc3() );
    }
}
}

```

- A. 34
  - B. 342Exception in thread "main" MyExc3
  - C. Errore a tempo di compilazione
  - D. 342
  - 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 A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);

```

```

}
catch( MyExc1 e ) {
    System.out.print(2);
}
catch( MyExc3 d ) {
    System.out.print(3);
    throw( new Exception() );
}
catch( Exception i ) {
    System.out.print(4);
}
finally {
    System.out.print(5);
}
}
static void p() throws Exception {
    try {
        System.out.print(6);
        if (true) throw( new MyExc3());
        throw( new MyExc1() );
    }
    catch( MyExc1 t ) {
        throw( new MyExc1() );
    }
    catch( MyExc2 s ) {
    }
}
}

```

- A. 635Exception in thread "main" java.lang.Exception
  - B. 63Exception in thread "main" java.lang.Exception
  - C. 615
  - 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. 28**

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

2. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}
```

- A. XYY

- B. YYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

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

```
public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}
```

```
public static void print() {
    System.out.print(floors + " "
        + constructor + " ");
}
}
```

```
class Run {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

6. Qual'è l'output di questo codice?

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

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione

D. 342

E. Nessuna delle precedenti

---

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

A. Jones Smith 50 Black

B. Jones 21 Smith 50 Black

C. Black Black 50 Black

D. Errore a tempo di compilazione.

E. Errore a tempo di esecuzione.

---

8. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

A. public void f(String x)

B. public byte f(Object y)

C. protected int f(String z)

D. int f(String s) throws Exception

E. Nessuna delle precedenti.

---

9. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
        float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

A. 100

B. 011

C. 111

D. 000

E. 101

---

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }
}
```

```

}

public void print() {
    System.out.print( constructor + " ");
}
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

11. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
}

```

```

}
static void p() throws Exception {
    try {
        System.out.print(6);
        if (true) throw( new MyExc3());
        throw( new MyExc1() );
    }
    catch( MyExc1 t ) {
        throw( new MyExc1() );
    }
    catch( MyExc2 s ) {
    }
}
}

```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

13. Date le dichiarazioni:

```

String [] t;
Object [] [] w;
Object [] z;

```

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

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

14. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```
Sub2 a;  
C1 h;  
A t;  
t = new C1();  
a = new Sub2();  
h = new C1();
```

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

- A. a = (Sub2) t;
  - B. h = (C1) a;
  - C. h = (C1) t;
  - D. a = (Sub2) h;
  - E. Nessuno dei precedenti
- 

15. Qual è il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {  
    A(String s) {}  
}
```

- A. public A(String b) {}
  - B. void A() {}
  - C. private A(int i, String s) {}
  - D. public Object A(int i, String s) {}
  - E. Il costruttore di una classe non può essere sovraccaricato.
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {  
    void m();  
}  
  
class X implements I {  
    String s = "X";  
}  
  
class Y extends X {  
    String s = "Y";  
  
    public void m() {  
        System.out.print(super.s  
            + ((X) this).s  
            + ((Y) this).s  
            + this.s);  
    }  
  
    public static void main(String[] args) {  
        new Y().m();  
    }  
}
```

- A. XXYY
  - B. XYYY
  - 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. 29**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}
```

- A. XYYX
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
```

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

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print (floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

3. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

4. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";
}

```

```

public void m() {
    System.out.print(super.s
        + ((X) this).s
        + ((Y) this).s
        + this.s);
}

public static void main(String[] args){
    new Y().m();
}

```

- A. XYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

6. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

```

```

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

8. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

9. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

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

- A. `a = (Sub2) t;`
- B. `h = (C1) a;`
- C. `h = (C1) t;`
- D. `a = (Sub2) h;`
- 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 A1 {
    public static void main(String[] argv)
        throws Exception {

        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3() );
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}

```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

11. Qual'è l'output di questo codice?

```

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

```

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {

```

```

        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

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

```

public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee(){
        System.out.println(brand);
    }
    Coffee(int ml){
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]){
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

14. Quale output si ottiene invocando il metodo p?

```

class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
        float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 100
- B. 011
- C. 111
- D. 000

- 
15. Qualè il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

```

- A. public A(String b) {}
  - B. void A(){}
  - C. private A(int i, String s) {}
  - D. public Object A(int i, String s) {}
  - E. Il costruttore di una classe non può essere sovraccaricato.
- 

16. Date le dichiarazioni:

```

String [] t;
Object [] [] w;
Object [] z;

```

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

- A. t = w;
  - B. t = z;
  - C. t = (String []) w;
  - D. w = (Object [] []) t;
  - E. w = (Object [] []) z;
-



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 il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

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

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

2. Qual'è l'output di questo codice?

3. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
```



```

String s8 = "";
s2 = s1.concat("abcd");
s6 = "";
a4 = a3;
q(s4, s6, s8, a4, a3);
}
void q(String s3, String s5, String s7,
float[] a1, Object a2){
if(s3 == s2) {
System.out.print(1);
} else {
System.out.print(0);
}
if(s5 == s7) {
System.out.print(1);
} else {
System.out.print(0);
}
if(a2 == a1) {
System.out.print(1);
} else {
System.out.print(0);
}
}
}

```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
String constructor;
int floors;

Building(int f, String c) {
floors = f;
constructor = c;
}

public void print() {
System.out.print(constructor + " ");
}
}

class Skyscraper extends Building {
Skyscraper(int f, String c) {super(f,c);}

public void print() {
System.out.print(floors + " "
+ constructor + " ");
}
}

```

```

class Main {
public static void main(String[] args) {
Building a = new Building(3, "Jones");
Building b = new Skyscraper(21, "Smith");
Skyscraper c = new Skyscraper(50, "Black");
a.print();
b.print();
c.print();
}
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
void m();
}

abstract class X implements I {
String s = "X";
}

class Y extends X {
String s = "Y";

public void m() {
System.out.print(super.s
+ ((X) this).s
+ ((Y) this).s
+ this.s);
}

public static void main(String[] args){
new Y().m();
}
}

```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
void m();
}

```

```

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}

```

- A. XYY
- B. XYYY
- 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 A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

8. Date le dichiarazioni:

```

String [] t;
Object [] [] w;
Object [] z;

```

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

- A. `t = w;`

- B. `t = z;`
- C. `t = (String []) w;`
- D. `w = (Object [] []) t;`
- E. `w = (Object [] []) z;`

9. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

10. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

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

- A. `a = (Sub2) t;`
- B. `h = (C1) a;`
- C. `h = (C1) t;`
- D. `a = (Sub2) h;`
- E. Nessuno dei precedenti

11. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {

        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {

```

```

    }
    catch( Exception e ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
    }
}
static void n() throws Exception {
    try {
        throw( new MyExc2() );
    }
    catch( MyExc1 t ) {
    }
    catch( Exception i ) {
    }
    catch( MyExc3 s ) {
        throw( new MyExc3() );
    }
    finally {
        System.out.print(5);
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

12. Qualè il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

```

- A. public A(String b) {}
- B. void A(){}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

13. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();

```

```

        System.out.print(1);
    }
    catch( MyExc1 e ) {
        System.out.print(2);
    }
    catch( MyExc3 d ) {
        System.out.print(3);
        throw( new Exception() );
    }
    catch( Exception i ) {
        System.out.print(4);
    }
    finally {
        System.out.print(5);
    }
}
static void p() throws Exception {
    try {
        System.out.print(6);
        if (true) throw( new MyExc3());
        throw( new MyExc1() );
    }
    catch( MyExc1 t ) {
        throw( new MyExc1() );
    }
    catch( MyExc2 s ) {
    }
}
}

```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```

public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee(){
        System.out.println(brand);
    }
    Coffee(int ml){
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]){
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}
```

```
class Run {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

16. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws 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. 31**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}
```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
```

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

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
```

```

    }
    catch( Exception e ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
    }
}
static void n() throws Exception {
    try {
        throw( new MyExc2() );
    }
    catch( MyExc1 t ) {
    }
    catch( Exception i ) {
    }
    catch( MyExc3 s ) {
        throw( new MyExc3() );
    }
    finally {
        System.out.print(5);
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print( constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

```

```

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

5. Qual'è l'output di questo codice?

```

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

```

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

6. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

7. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}
```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```
public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

9. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?



```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args) {
        new Y().m();
    }
}

```

- A. XYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Qual è il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

```

- A. public A(String b) {}
- B. void A() {}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

12. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception

E. Nessuna delle precedenti.

13. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

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

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti

14. Date le dichiarazioni:

```

String [] t;
Object [] [] w;
Object [] z;

```

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

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

15. Quale output si ottiene invocando il metodo p?

```

class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
}

```

```

}
void q(String s3, String s5, String s7,
    float[] a1, Object a2){
    if(s3 == s2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s5 == s7) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {

```

```

        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

```

```

class Run {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.



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 il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
}
```

- A. public A(String b) {}
- B. void A(){}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}
```

- A. XYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.

- E. Nessuna delle precedenti.

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

```
public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}
```

```

}
class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

5. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());

```

```

        throw( new MyExc1() );
    }
    catch( MyExc1 t ) {
        throw( new MyExc1() );
    }
    catch( MyExc2 s ) {
    }
}

```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

7. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

---

8. Date le dichiarazioni:

```
String [] t;
Object [] [] w;
Object [] z;
```

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

- A. `t = w;`

- B. `t = z;`
- C. `t = (String []) w;`
- D. `w = (Object [] []) t;`
- E. `w = (Object [] []) z;`

---

9. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

---

10. Qual'è l'output di questo codice?

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

- A. 34
  - B. 342Exception in thread "main" MyExc3
  - C. Errore a tempo di compilazione
  - D. 342
  - E. Nessuna delle precedenti
-

11. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}
```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

14. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
```

```

float[] a1, Object a2){
if(s3 == s2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s5 == s7) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(a2 == a1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
}

```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

15. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

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

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti

16. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws 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. 33**

1. Dato un file contenente il seguente codice:

```
package pk;
```

```
public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

2. Qual'è l'output di questo codice?

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

```
}
finally {
    System.out.print(4);
    throw( new MyExc3() );
}
}
```

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

3. Qual'è il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
}
```

- A. `public A(String b) {}`
- B. `void A() {}`
- C. `private A(int i, String s) {}`
- D. `public Object A(int i, String s) {}`
- E. Il costruttore di una classe non può essere sovraccaricato.

4. Date le dichiarazioni:

```
class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();
```

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

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- 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 A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- 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 A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}
```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print (constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print (floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

9. Date le dichiarazioni:

```

String [] t;
Object [] [] w;
Object [] z;

```

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

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print (super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}

```

- A. XXYY
- B. YYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print (constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print (floors + " "
            + constructor + " ");
    }
}

```

```

}
class Run {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

12. Quale output si ottiene invocando il metodo p?

```

class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
        float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 100
- B. 011

- C. 111
- D. 000
- E. 101

---

Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

---

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black

- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

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

```
public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

---

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args) {
        new Y().m();
    }
}
```

- A. XXYY
  - B. XYYY
  - 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. 34**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

2. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}
```

- A. XXYY
- B. XYYY
- 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 A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black

- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

6. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {

        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }

    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}

```

- A. XYYX
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Quale output si ottiene invocando il metodo p?

```

class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
        float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```

        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

9. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {

        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}

```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

10. Date le dichiarazioni:

```
String [] t;  
Object [] [] w;  
Object [] z;
```

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

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

---

11. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

---

12. Qual è il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {  
    A(String s) {}  
}
```

- A. public A(String b) {}
- B. void A() {}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

---

13. Date le dichiarazioni:

```
class C1 extends A {...}  
class A extends Object {...}  
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;  
C1 h;  
A t;  
t = new C1();  
a = new Sub2();  
h = new C1();
```

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

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti

---

14. Qual è l'output di questo codice?

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

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

---

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black

- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

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

```

public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6



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

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

2. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

3. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
        float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

4. Date le dichiarazioni:

```
String [] t;
Object [] [] w;
Object [] z;
```

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

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

6. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

7. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

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

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti

8. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {

```

```

    }
}

```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

```

```

abstract class X implements I {
    String s = "X";
}

```

```

class Y extends X {
    String s = "Y";

```

```

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

```

```

    public static void main(String[] args){
        new Y().m();
    }
}

```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

```

```

class X implements I {
    String s = "X";
}

```

```

class Y extends X {
    String s = "Y";

```



```

public void m() {
    System.out.print(super.s
        + ((X) this).s
        + ((Y) this).s
        + this.s);
}

public static void main(String[] args) {
    new Y().m();
}
}

```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black

- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

12. Qual è il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

```

- A. public A(String b) {}
- B. void A() {}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

13. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

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

```

public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- A. 1

- B. 2
  - C. 3
  - D. 4
  - E. 6
- 

15. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
  - B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
  - C. In Java tutti i tipi numerici sono con segno
  - D. Un array possiede dei membri
  - E. Un array vuoto può avere riferimento null
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print () {
        System.out.print (constructor + " ");
    }
}
```

```
    }
}

class Skyscraper extends Building {
    Skyscraper () {}
    Skyscraper (int f, String c) {super (f,c);}

    public static void print () {
        System.out.print (floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main (String [] args) {
        Building a = new Building (3, "Jones");
        Building b = new Skyscraper (21, "Smith");
        Skyscraper c = new Skyscraper (50, "Black");
        a.print ();
        b.print ();
        c.print ();
    }
}
```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
-



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

**Prova n. 36**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

2. Quale output si ottiene invocando il metodo p?

```
class B {
```

```
private String s1 = "";
private String s2;
private float [] a3 = new float [8];
private float [] a4;
void p() {
    String s4 = s1.concat("abcd");
    String s6;
    String s8 = "";
    s2 = s1.concat("abcd");
    s6 = "";
    a4 = a3;
    q(s4, s6, s8, a4, a3);
}
void q(String s3, String s5, String s7,
    float[] a1, Object a2){
    if(s3 == s2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s5 == s7) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

3. Qual'è l'output di questo codice?

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

```

try {
    q();
}
catch( MyExc3 x ) {
}
catch( MyExc1 c ) {
}
catch( MyExc2 c ) {
    System.out.print(1);
}
finally {
    System.out.print(2);
}
}
static void q() {
    try {
        System.out.print(3);
        throw( new MyExc3() );
    }
    finally {
        System.out.print(4);
        throw( new MyExc3() );
    }
}
}

```

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}

```

- A. XXYY
- B. YYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

6. Qual è il sovraccaricamento corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

A. public A(String b) {}

```

- B. void A() {}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

7. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

8. Date le dichiarazioni:

```
class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();
```

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

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti

9. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
    }
}
```

```
catch( MyExc2 r ) {
}
catch( MyExc1 j ) {
}
catch( Exception e ) {
    System.out.print(3);
}
finally {
    System.out.print(4);
}
}
static void n() throws Exception {
    try {
        throw( new MyExc2() );
    }
    catch( MyExc1 t ) {
    }
    catch( Exception i ) {
    }
    catch( MyExc3 s ) {
        throw( new MyExc3() );
    }
    finally {
        System.out.print(5);
    }
}
}
```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

10. Date le dichiarazioni:

```
String [] t;
Object [] [] w;
Object [] z;
```

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

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args) {
        new Y().m();
    }
}

```

- A. XYY
- B. XYYY
- 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 A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {

```

```

            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}

```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

13. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

```

```

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

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

```

public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
    }
}

```

```

ristretto = null;
vanillio = ristretto;
/* QUI */
return;

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

---

16. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
  - 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 A.I()`.
  - D. Un'istanza della classe I può essere costruita con `new A().new 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. 37**

1. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
        float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

```
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
```

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print (constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print (floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print (constructor + " ");
    }
}

class Skyscraper extends Building {

```

```

    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print (floors + " "
            + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

5. Qual'è l'output di questo codice?

```

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

```

- A. 34

- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

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

```
public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

7. Date le dichiarazioni:

```
String [] t;
Object [] [] w;
Object [] z;
```

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

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

8. Qualè il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
}
```

- A. public A(String b) {}
- B. void A() {}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

9. Date le dichiarazioni:

```
class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();
```

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

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti

10. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f, c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

12. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
  - B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
  - C. In Java tutti i tipi numerici sono con segno
  - D. Un array possiede dei membri
  - E. Un array vuoto può avere riferimento null
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args) {
        new Y().m();
    }
}
```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args) {
        new Y().m();
    }
}
```

- A. XXYY

- B. XYYY
  - 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 A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
```

```
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}
```

- A. 635Exception in thread "main" java.lang.Exception
  - B. 63Exception in thread "main" java.lang.Exception
  - C. 615
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws 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. 38**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}
```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";
```

```
public void m() {
    System.out.print(super.s
        + ((X) this).s
        + ((Y) this).s
        + this.s);
}

public static void main(String[] args){
    new Y().m();
}
}
```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

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

```
public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4



## 4. Qual'è l'output di questo codice?

```

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

```

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

## 5. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

## 6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

## 7. Quale output si ottiene invocando il metodo p?

```

class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
}

```

```

void q(String s3, String s5, String s7,
    float[] a1, Object a2){
    if(s3 == s2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s5 == s7) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
    }
}

```

```

        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

9. Date le dichiarazioni:

```

String [] t;
Object [] [] w;
Object [] z;

```

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

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

10. Qual'è l'output di questo codice?

```

class MyExcl extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExcl e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExcl() );
        }
    }
}

```

```

    }
    catch( MyExc1 t ) {
        throw( new MyExc1() );
    }
    catch( MyExc2 s ) {
    }
}
}

```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

11. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

12. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. `public void f(String x)`
- B. `public byte f(Object y)`
- C. `protected int f(String z)`
- D. `int f(String s) throws Exception`
- E. Nessuna delle precedenti.

13. Qual è il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

```

- A. `public A(String b) {}`
- B. `void A() {}`
- C. `private A(int i, String s) {}`
- D. `public Object A(int i, String s) {}`
- E. Il costruttore di una classe non può essere sovraccaricato.

14. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

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

- A. `a = (Sub2) t;`
- B. `h = (C1) a;`
- C. `h = (C1) t;`
- D. `a = (Sub2) h;`
- E. Nessuno dei precedenti

15. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

```

}
static void n() throws Exception {
    try {
        throw( new MyExc2() );
    }
    catch( MyExc1 t ) {
    }
    catch( Exception i ) {
    }
    catch( MyExc3 s ) {
        throw( new MyExc3() );
    }
    finally {
        System.out.print(5);
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
    }
}

```

```

        constructor = c;
    }

    public static void print() {
        System.out.print( constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

```

```

class Run {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.



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

**Prova n. 39**

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

2. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}
```

```

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

4. Quale output si ottiene invocando il metodo p?

```

class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
        float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 100
- B. 011
- C. 111
- D. 000

E. 101

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

```

public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args) {
        new Y().m();
    }
}

```

- A. XYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

8. Date le dichiarazioni:

```
class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();
```

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

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti

9. Qual è il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
}
```

- A. public A(String b) {}
- B. void A() {}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

10. Date le dichiarazioni:

```
String [] t;
Object [] [] w;
Object [] z;
```

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

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

11. Qual è l'output di questo codice?



```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}

```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

13. Qual'è l'output di questo codice?

```

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

```

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

14. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}

```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;
}

```

```

Building(int f, String c) {
    floors = f;
    constructor = c;
}

public static void print() {
    System.out.print(constructor + " ");
}

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.



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. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
        float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

```
static String constructor;
static int floors;

Building(int f, String c) {
    floors = f;
    constructor = c;
}

public static void print() {
    System.out.print(constructor + " ");
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
```

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
```

```

}
abstract class X implements I {
    String s = "X";
}
class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}

```

- A. XYY
- B. XYYY
- 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 A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- 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 A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;
}

```

```

Building(int f, String c) {
    floors = f;
    constructor = c;
}

public void print() {
    System.out.print(constructor + " ");
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }
}

```

```

public static void main(String[] args){
    new Y().m();
}
}

```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Date le dichiarazioni:

```

String [] t;
Object [] [] w;
Object [] z;

```

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

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

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

```

public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee(){
        System.out.println(brand);
    }
    Coffee(int ml){
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]){
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4

E. 6

9. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}

```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

10. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

---

11. Qual è il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
}
```

- A. public A(String b) {}
- B. void A(){}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

---

12. Qual'è l'output di questo codice?

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

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}
```

```
class Run {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

14. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {

        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
    }
}
```

```

    catch( MyExc1 j ) {
    }
    catch( Exception e ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
    }
}
static void n() throws Exception {
    try {
        throw( new MyExc2() );
    }
    catch( MyExc1 t ) {
    }
    catch( Exception i ) {
    }
    catch( MyExc3 s ) {
        throw( new MyExc3() );
    }
    finally {
        System.out.print(5);
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

15. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

16. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

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

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti



Prova n. 1

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

---

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Prova n. 2

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

---

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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



Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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



Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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



Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

---

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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



Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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: