

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

Prova n. 1

1. Qual è l'output di questo programma?

```
abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

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

2. Qual è l'output di questo programma?

```
class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

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

3. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}
```

```
class B {
    private A a1 = new A("o1");
    private A a2 = new A("o2");
    private A a3 = new A("o3");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
    private void m() {}
    public static void main (String[] args) {
        new B().f();
    }
}
```

- A. Quello con s=o1
- B. Quello con s=o2
- C. Quello con s=o3
- D. Quelli con s uguale a o1 e o2
- E. Nessuna delle precedenti.

4. Quale output si ottiene invocando il metodo q?

```

class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {
        String s7 = s4.concat("abc");
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 001
- C. 100
- D. 010
- E. 000

5. Qual'è l'output di questo codice?

```

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

```

```

try {
    System.out.print(4);
    throw( new MyExc1() );
}
catch( MyExc1 f ) {
}
finally {
    System.out.print(5);
}
}

```

- A. 45132222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

6. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception e ) {
            System.out.print(2);
        }
        catch( MyExc3 j ) {
            System.out.print(3);
        }
        catch( MyExc2 w ) {
            System.out.print(4);
        }
    }
    static void m() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(6);
        }
    }
}

```

- A. 156Exception in thread main MyExc1
- B. 156
- C. 1562
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}
```

- A. 125
 - B. 135
 - C. 1235
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

8. Data il seguente codice:

```
1. public interface I {
2.     int x = 0;
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
 - B. abstract int x = 0;
 - C. protected int x = 0;
 - D. public int x;
 - E. Nessuna delle precedenti
-

9. Qual è l'output di questo programma?

```
abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
    }
}
```

```
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

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

10. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```
class X {
    static char a;
    static byte b;
    static short c;
    static int d;
    static long e;
    static String s;

    public static void main(String[] args) {
        System.out.println(a+b+c+d+e+s);
    }
}
```

- A. 00000null
 - B. 0null
 - C. 0
 - D. null
 - E. Errore di compilazione.
-

11. Date le dichiarazioni:

```
Boolean p;
Object w;
String z;
```

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

- A. p = (Boolean) z;
 - B. p = w;
 - C. w = z;
 - D. p = z;
 - E. z = p;
-

12. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- D. È possibile dichiarare un attributo senza iniziarlo
- E. Una classe non può essere dichiarata final

13. Qual'è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 u ) {
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc1 g ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

14. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
```

```
}
class Rectangle extends Polygon{
    Rectangle() {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

15. Date le dichiarazioni:

```
Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];
```

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

- A. a = (Boolean []) z;
- B. t = (Error []) z;
- C. t = (Error []) a;
- D. a = (Boolean []) t;
- E. Nessuno dei precedenti

16. Data la seguente classe:

```
1. public class C {
2.     public void m(int i, String s){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. public int m(int i, String s)
- B. public void m(int i, String n)
- C. public void M(int i, String s)
- D. private void m(int j, String q)
- E. Nessuna delle precedenti

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

Prova n. 2

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 f ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc1 f ) {
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Qual è l'output di questo programma?

```
abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}
```

```
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

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

3. Data la seguente classe:

```
1. public class C {
2.     public void m(int i, String s){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. public int m(int i, String s)
- B. public void m(int i, String n)
- C. public void M(int i, String s)
- D. private void m(int j, String q)
- E. Nessuna delle precedenti

4. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
```

```

}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle() {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- 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 D1 {

```

```

public static void main(String [] argv) {
    try {
        p();
        System.out.print(1);
    }
    catch( MyExc3 u ) {
    }
}
static void p() {
    try {
        System.out.print(2);
        throw( new MyExc1() );
    }
    catch( MyExc1 g ) {
        System.out.print(3);
        throw( new MyExc1() );
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Date le dichiarazioni:

```

Boolean p;
Object w;
String z;

```

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

- A. p = (Boolean) z;
- B. p = w;
- C. w = z;
- D. p = z;
- E. z = p;

8. Qual è l'output di questo programma?

```

class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {

```



```

        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

9. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}

class B {
    private A a1 = new A(o1");
    private A a2 = new A(o2");
    private A a3 = new A(o3");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
    private void m() {}
    public static void main (String[] args) {
        new B().f();
    }
}

```

- A. Quello con s=o1
- B. Quello con s=o2
- C. Quello con s=o3
- D. Quelli con s uguale a o1 e o2
- E. Nessuna delle precedenti.

10. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```

class X {
    static char a;
    static byte b;
    static short c;
}

```

```

static int d;
static long e;
static String s;

public static void main(String[] args) {
    System.out.println(a+b+c+d+e+s);
}
}

```

- A. 00000null
- B. 0null
- C. 0
- D. null
- E. Errore di compilazione.

11. Qual'è l'output di questo codice?

```

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

        try {
            System.out.print(1);
            m();
        }
        catch( Exception e ) {
            System.out.print(2);
        }
        catch( MyExc3 j ) {
            System.out.print(3);
        }
        catch( MyExc2 w ) {
            System.out.print(4);
        }
    }
    static void m() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(6);
        }
    }
}

```

- A. 156Exception in thread main MyExc1
- B. 156
- C. 1562
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. Data il seguente codice:

```

1. public interface I {
2.     int x = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
- B. abstract int x = 0;
- C. protected int x = 0;
- D. public int x;
- E. Nessuna delle precedenti

13. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- D. È possibile dichiarare un attributo senza inizializzarlo
- E. Una classe non può essere dichiarata final

14. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

15. Quale output si ottiene invocando il metodo q?

```

class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {
        String s7 = s4.concat("abc");
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 001
- C. 100
- D. 010
- E. 000

16. Date le dichiarazioni:

```

Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];

```

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

- A. a = (Boolean []) z;
- B. t = (Error []) z;
- C. t = (Error []) a;
- D. a = (Boolean []) t;
- E. Nessuno dei precedenti

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

Prova n. 3

1. Date le dichiarazioni:

```
Boolean p;
Object w;
String z;
```

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

- A. `p = (Boolean) z;`
- B. `p = w;`
- C. `w = z;`
- D. `p = z;`
- E. `z = p;`

2. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- 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 D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 u ) {
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc1 g ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. Data il seguente codice:

```
1. public interface I {
2.     int x = 0;
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. `final int x = 0;`
- B. `abstract int x = 0;`
- C. `protected int x = 0;`

- D. public int x;
- E. Nessuna delle precedenti

5. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```
class X {
    static char a;
    static byte b;
    static short c;
    static int d;
    static long e;
    static String s;

    public static void main(String[] args) {
        System.out.println(a+b+c+d+e+s);
    }
}
```

- A. 00000null
- B. 0null
- C. 0
- D. null
- E. Errore di compilazione.

6. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle() {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Data la seguente classe:

```
1. public class C {
2.     public void m(int i, String s){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. public int m(int i, String s)
- B. public void m(int i, String n)
- C. public void M(int i, String s)
- D. private void m(int j, String q)
- E. Nessuna delle precedenti

8. Qual è l'output di questo programma?

```
class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

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

9. Quale output si ottiene invocando il metodo q?

```
class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {
```

```

String s7 = s4.concat("abc");
if(s3 == s2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(b2 == b1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s5 == s7) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
}

```

- A. 110
- B. 001
- C. 100
- D. 010
- E. 000

10. Date le dichiarazioni:

```

Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];

```

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

- A. a = (Boolean []) z;
- B. t = (Error []) z;
- C. t = (Error []) a;
- D. a = (Boolean []) t;
- E. Nessuno dei precedenti

11. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
}

```

```

}
public void f() {
    System.out.println("F");
}
public void m() {
    f();
}
}

```

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

12. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}

```

```

class B {
    private A a1 = new A("o1");
    private A a2 = new A("o2");
    private A a3 = new A("o3");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
    private void m() {}
    public static void main (String[] args) {
        new B().f();
    }
}

```

- A. Quello con s=o1
- B. Quello con s=o2
- C. Quello con s=o3
- D. Quelli con s uguale a o1 e o2
- E. Nessuna delle precedenti.

13. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi

- C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- D. È possibile dichiarare un attributo senza inizializzarlo
- E. Una classe non può essere dichiarata final

14. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 f ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc1 f ) {
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

15. Qual'è l'output di questo codice?

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

```
        m();
        }
        catch( Exception e ) {
            System.out.print(2);
        }
        catch( MyExc3 j ) {
            System.out.print(3);
        }
        catch( MyExc2 w ) {
            System.out.print(4);
        }
    }
    static void m() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(6);
        }
    }
}
```

- A. 156Exception in thread main MyExc1
- B. 156
- C. 1562
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

16. Qual è l'output di questo programma?

```
abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

- A. M
- B. F
- 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. 4

1. Date le dichiarazioni:

```
Boolean p;
Object w;
String z;
```

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

- A. `p = (Boolean) z;`
- B. `p = w;`
- C. `w = z;`
- D. `p = z;`
- E. `z = p;`

2. Qual è l'output di questo programma?

```
class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

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

3. Quale output si ottiene invocando il metodo q?

```
class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {
        String s7 = s4.concat("abc");
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 001
- C. 100
- D. 010
- E. 000

4. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```
class X {
    static char a;
    static byte b;
    static short c;
    static int d;
    static long e;
```

```

static String s;

public static void main(String[] args) {
    System.out.println(a+b+c+d+e+s);
}
}

```

- A. 0000null
- B. 0null
- C. 0
- D. null
- E. Errore di compilazione.

5. Data la seguente classe:

```

1. public class C {
2.     public void m(int i, String s){}
3.
4. }

```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. public int m(int i, String s)
- B. public void m(int i, String n)
- C. public void M(int i, String s)
- D. private void m(int j, String q)
- E. Nessuna delle precedenti

6. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print (2);
    }
    Polygon(int i) {
        System.out.print (3);
    }
}
class Rectangle extends Polygon{
    Rectangle() {
        super (6);
        System.out.print (5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print (1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235

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

7. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}

```

```

class B {
    private A a1 = new A(o1");
    private A a2 = new A(o2");
    private A a3 = new A(o3");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
    private void m() {}
    public static void main (String[] args) {
        new B().f();
    }
}

```

- A. Quello con s=o1
- B. Quello con s=o2
- C. Quello con s=o3
- D. Quelli con s uguale a o1 e o2
- E. Nessuna delle precedenti.

8. Date le dichiarazioni:

```

Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];

```

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

- A. a = (Boolean []) z;
- B. t = (Error []) z;
- C. t = (Error []) a;
- D. a = (Boolean []) t;
- E. Nessuno dei precedenti

9. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception e ) {
            System.out.print(2);
        }
        catch( MyExc3 j ) {
            System.out.print(3);
        }
        catch( MyExc2 w ) {
            System.out.print(4);
        }
    }
    static void m() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(6);
        }
    }
}
```

- A. 156Exception in thread main MyExc1
- B. 156
- C. 1562
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

10. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Un attributo **static** non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- D. È possibile dichiarare un attributo senza inicializzarlo
- E. Una classe non può essere dichiarata **final**

11. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 f ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc1 f ) {
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. Qual'è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 u ) {
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc1 g ) {
```

```

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

```

- A. 23Exception in thread main MyExcl
- B. 234Exception in thread main MyExcl
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

13. Data il seguente codice:

```

1. public interface I {
2.     int x = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
- B. abstract int x = 0;
- C. protected int x = 0;
- D. public int x;
- E. Nessuna delle precedenti

14. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione

E. Nessuna delle precedenti

15. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

16. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

- A. M
- B. F
- 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. 5

1. Quale output si ottiene invocando il metodo q?

```
class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {
        String s7 = s4.concat("abc");
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 001
- C. 100
- D. 010
- E. 000

2. Qual'è l'output di questo codice?

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

```
try {
    q();
    System.out.print(1);
}
catch( MyExc2 f ) {
    System.out.print(2);
    throw( new MyExc3() );
}
finally {
    System.out.print(3);
    throw( new MyExc3() );
}
}
static void q() throws Exception {
    try {
        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc1 f ) {
    }
    finally {
        System.out.print(5);
    }
}
}
```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

3. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- D. È possibile dichiarare un attributo senza iniziarlo
- E. Una classe non può essere dichiarata final

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception e ) {
            System.out.print(2);
        }
        catch( MyExc3 j ) {
            System.out.print(3);
        }
        catch( MyExc2 w ) {
            System.out.print(4);
        }
    }
    static void m() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(6);
        }
    }
}
```

- A. 156Exception in thread main MyExc1
- B. 156
- C. 1562
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
```

```
        Square s = new Square();
    }
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

6. Date le dichiarazioni:

```
Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];
```

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

- A. a = (Boolean []) z;
- B. t = (Error []) z;
- C. t = (Error []) a;
- D. a = (Boolean []) t;
- E. Nessuno dei precedenti

7. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```
class X {
    static char a;
    static byte b;
    static short c;
    static int d;
    static long e;
    static String s;

    public static void main(String[] args) {
        System.out.println(a+b+c+d+e+s);
    }
}
```

- A. 00000null
- B. 0null
- C. 0
- D. null
- E. Errore di compilazione.

8. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle() {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
 - B. 135
 - C. 1235
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

9. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

10. Date le dichiarazioni:

```

Boolean p;
Object w;
String z;

```

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

- A. p = (Boolean) z;
 - B. p = w;
 - C. w = z;
 - D. p = z;
 - E. z = p;
-

11. Data il seguente codice:

```

1. public interface I {
2.     int x = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
 - B. abstract int x = 0;
 - C. protected int x = 0;
 - D. public int x;
 - E. Nessuna delle precedenti
-

12. Data la seguente classe:

```

1. public class C {
2.     public void m(int i, String s){}
3.
4. }

```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. public int m(int i, String s)
 - B. public void m(int i, String n)
 - C. public void M(int i, String s)
 - D. private void m(int j, String q)
 - E. Nessuna delle precedenti
-

13. Qual è l'output di questo programma?


```

class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

14. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

- A. M
- B. F
- 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 MyExcl extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 u ) {
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExcl() );
        }
        catch( MyExcl g ) {
            System.out.print(3);
            throw( new MyExcl() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 23Exception in thread main MyExcl
- B. 234Exception in thread main MyExcl
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

16. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}
class B {
    private A a1 = new A(o1");
    private A a2 = new A(o2");
    private A a3 = new A(o3");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
    private void m() {}
    public static void main (String[] args) {
        new B().f();
    }
}

```

}
}

- A. Quello con $s=01$
- B. Quello con $s=02$

- C. Quello con $s=03$
 - D. Quelli con s uguale a $o1$ e $o2$
 - 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. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}

class B {
    private A a1 = new A("01");
    private A a2 = new A("02");
    private A a3 = new A("03");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
    private void m() {}
    public static void main (String[] args) {
        new B().f();
    }
}
```

- A. Quello con s="01"
- B. Quello con s="02"
- C. Quello con s="03"
- D. Quelli con s uguale a "01" e "02"
- E. Nessuna delle precedenti.

2. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```
class X {
    static char a;
    static byte b;
    static short c;
    static int d;
    static long e;
    static String s;
}
```

```
public static void main(String[] args) {
    System.out.println(a+b+c+d+e+s);
}
}
```

- A. 00000null
- B. 0null
- C. 0
- D. null
- E. Errore di compilazione.

3. Qual è l'output di questo programma?

```
abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}

class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

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

4. Qual'è l'output di questo codice?

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

```

        p();
        System.out.print(1);
    }
    catch( MyExc3 u ) {
    }
}
static void p() {
    try {
        System.out.print(2);
        throw( new MyExc1() );
    }
    catch( MyExc1 g ) {
        System.out.print(3);
        throw( new MyExc1() );
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

6. Data la seguente classe:

```

1. public class C {
2.     public void m(int i, String s){}
3.
4. }

```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. public int m(int i, String s)
- B. public void m(int i, String n)
- C. public void M(int i, String s)
- D. private void m(int j, String q)
- E. Nessuna delle precedenti

7. Date le dichiarazioni:

```

Boolean p;
Object w;
String z;

```

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

- A. p = (Boolean) z;
- B. p = w;
- C. w = z;
- D. p = z;
- E. z = p;

8. Quale output si ottiene invocando il metodo q?

```

class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {
        String s7 = s4.concat("abc");
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        }
    }
}

```

```

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

```

- A. 110
- B. 001
- C. 100
- D. 010
- E. 000

9. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

10. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- D. È possibile dichiarare un attributo senza iniziarlo
- E. Una classe non può essere dichiarata final

11. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle() {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. Date le dichiarazioni:

```

Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];

```

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

- A. a = (Boolean []) z;
- B. t = (Error []) z;
- C. t = (Error []) a;
- D. a = (Boolean []) t;
- E. Nessuno dei precedenti

13. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 f ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc1 f ) {
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

14. Qual'è l'output di questo codice?

```

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

```

```

            System.out.print(5);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(6);
        }
    }
}

```

- A. 156Exception in thread main MyExc1
- B. 156
- C. 1562
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

15. Data il seguente codice:

```

1. public interface I {
2.     int x = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
- B. abstract int x = 0;
- C. protected int x = 0;
- D. public int x;
- E. Nessuna delle precedenti

16. Qual è l'output di questo programma?

```

class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

- A. M
- B. F
- 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. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 f ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc1 f ) {
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
```

```
public static void main(String [] argv)
    throws Exception {
    try {
        System.out.print(1);
        m();
    }
    catch( Exception e ) {
        System.out.print(2);
    }
    catch( MyExc3 j ) {
        System.out.print(3);
    }
    catch( MyExc2 w ) {
        System.out.print(4);
    }
}
static void m() throws Exception {
    try {
        System.out.print(5);
        throw( new MyExc1() );
    }
    finally {
        System.out.print(6);
    }
}
}
```

- A. 156Exception in thread main MyExc1
- B. 156
- C. 1562
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

3. Qual è l'output di questo programma?

```
abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
}
```

```

public void f() {
    System.out.println("F");
}
public void m() {
    f();
}
}

```

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

4. Quale output si ottiene invocando il metodo q?

```

class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {
        String s7 = s4.concat("abc");
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 001
- C. 100
- D. 010
- E. 000

5. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle() {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

6. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- D. È possibile dichiarare un attributo senza inizializzarlo
- E. Una classe non può essere dichiarata final

7. Data il seguente codice:

```

1. public interface I {
2.     int x = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
- B. abstract int x = 0;
- C. protected int x = 0;
- D. public int x;
- E. Nessuna delle precedenti

8. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}

class B {
    private A a1 = new A("o1");
    private A a2 = new A("o2");
    private A a3 = new A("o3");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
    private void m() {}
    public static void main (String[] args) {
        new B().f();
    }
}
```

- A. Quello con s=o1
- B. Quello con s=o2
- C. Quello con s=o3
- D. Quelli con s uguale a o1 e o2
- E. Nessuna delle precedenti.

9. Qual è l'output di questo programma?

```
class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}

class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

- A. M
- B. F

- 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 Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 u ) {
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc1 g ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

11. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```
class X {
    static char a;
    static byte b;
    static short c;
    static int d;
    static long e;
    static String s;

    public static void main(String[] args) {
        System.out.println(a+b+c+d+e+s);
    }
}
```

- A. 00000null
- B. 0null

- C. 0
 - D. null
 - E. Errore di compilazione.
-

12. Data la seguente classe:

```
1. public class C {  
2.     public void m(int i, String s){}  
3.  
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. `public int m(int i, String s)`
 - B. `public void m(int i, String n)`
 - C. `public void M(int i, String s)`
 - D. `private void m(int j, String q)`
 - E. Nessuna delle precedenti
-

13. Date le dichiarazioni:

```
Boolean p;  
Object w;  
String z;
```

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

- A. `p = (Boolean) z;`
 - B. `p = w;`
 - C. `w = z;`
 - D. `p = z;`
 - E. `z = p;`
-

14. Date le dichiarazioni:

```
Boolean [] a;  
Error [] t;  
Object [] z;  
z = new Boolean [3];  
t = new Error [2];  
a = new Boolean [8];
```

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

- A. `a = (Boolean []) z;`
 - B. `t = (Error []) z;`
 - C. `t = (Error []) a;`
 - D. `a = (Boolean []) t;`
 - E. Nessuno dei precedenti
-

15. Qual è l'output di questo programma?

```
class Polygon {  
    Polygon() {  
        System.out.print (2);  
    }  
    Polygon(int i) {  
        System.out.print (3);  
    }  
}  
class Rectangle extends Polygon{  
    Rectangle(int i) {  
        super (6);  
        System.out.print (5);  
    }  
}  
class Square extends Rectangle {  
    public static void main(String argv[]) {  
        System.out.print (1);  
        Square s = new Square ();  
    }  
}
```

- A. 125
 - B. 135
 - C. 1235
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

16. Qual è l'output di questo programma?

```
abstract class A {  
    public void f() {}  
    public static void m() {  
        System.out.println ("M");  
    }  
}  
class B extends A {  
    public static void main(String argv[]) {  
        B b = new B ();  
        b.m ();  
    }  
    public void f() {  
        System.out.println ("F");  
    }  
    public void m() {  
        f ();  
    }  
}
```

- A. M
 - B. F
 - 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. 8

1. Qual'è l'output di questo codice?

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

        try {
            System.out.print(1);
            m();
        }
        catch( Exception e ) {
            System.out.print(2);
        }
        catch( MyExc3 j ) {
            System.out.print(3);
        }
        catch( MyExc2 w ) {
            System.out.print(4);
        }
    }
    static void m() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(6);
        }
    }
}
```

- A. 156Exception in thread main MyExc1
- B. 156
- C. 1562
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi

- C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- D. È possibile dichiarare un attributo senza inicializzarlo
- E. Una classe non può essere dichiarata final

3. Data la seguente classe:

```
1. public class C {
2.     public void m(int i, String s){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. public int m(int i, String s)
- B. public void m(int i, String n)
- C. public void M(int i, String s)
- D. private void m(int j, String q)
- E. Nessuna delle precedenti

4. Qual'è l'output di questo codice?

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

```

        finally {
            System.out.print(4);
        }
    }
}

```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. Date le dichiarazioni:

```

Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];

```

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

- A. a = (Boolean []) z;
- B. t = (Error []) z;
- C. t = (Error []) a;
- D. a = (Boolean []) t;
- E. Nessuno dei precedenti

6. Qual'è l'output di questo codice?

```

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

```

```

    }
    catch( MyExc1 f ) {
    }
    finally {
        System.out.print(5);
    }
}

```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Data il seguente codice:

```

1. public interface I {
2.     int x = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
- B. abstract int x = 0;
- C. protected int x = 0;
- D. public int x;
- E. Nessuna delle precedenti

8. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

9. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```
class X {
    static char a;
    static byte b;
    static short c;
    static int d;
    static long e;
    static String s;

    public static void main(String[] args) {
        System.out.println(a+b+c+d+e+s);
    }
}
```

- A. 00000null
- B. 0null
- C. 0
- D. null
- E. Errore di compilazione.

10. Quale output si ottiene invocando il metodo q?

```
class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {
        String s7 = s4.concat("abc");
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110

- B. 001
- C. 100
- D. 010
- E. 000

11. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}
```

```
class B {
    private A a1 = new A("o1");
    private A a2 = new A("o2");
    private A a3 = new A("o3");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
    private void m() {}
    public static void main (String[] args) {
        new B().f();
    }
}
```

- A. Quello con s=o1
- B. Quello con s=o2
- C. Quello con s=o3
- D. Quelli con s uguale a o1 e o2
- E. Nessuna delle precedenti.

12. Qual è l'output di questo programma?

```
class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
}
```



```

    public void m() {
        f();
    }
}

```

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

13. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

14. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {

```

```

        f();
    }
}

```

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

15. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle() {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

16. Date le dichiarazioni:

```

Boolean p;
Object w;
String z;

```

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

- A. p = (Boolean) z;
- B. p = w;
- C. w = z;
- D. p = z;
- E. z = p;

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

Prova n. 9

1. Qual è l'output di questo programma?

```
abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

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

2. Date le dichiarazioni:

```
Boolean p;
Object w;
String z;
```

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

- A. `p = (Boolean) z;`
- B. `p = w;`
- C. `w = z;`
- D. `p = z;`
- E. `z = p;`

3. Data la seguente classe:

```
1. public class C {
2.     public void m(int i, String s){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. `public int m(int i, String s)`
- B. `public void m(int i, String n)`
- C. `public void M(int i, String s)`
- D. `private void m(int j, String q)`
- E. Nessuna delle precedenti

4. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}

class B {
    private A a1 = new A(o1");
    private A a2 = new A(o2");
    private A a3 = new A(o3");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
    private void m() {}
    public static void main (String[] args) {
        new B().f();
    }
}

```

- A. Quello con s=o1
- B. Quello con s=o2
- C. Quello con s=o3
- D. Quelli con s uguale a o1 e o2
- E. Nessuna delle precedenti.

6. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print (2);
    }
    Polygon(int i) {
        System.out.print (3);
    }
}

class Rectangle extends Polygon{
    Rectangle() {
        super (6);
        System.out.print (5);
    }
}

class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print (1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione

E. Nessuna delle precedenti

7. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}

class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

8. Qual'è l'output di questo codice?

```

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

        try {
            System.out.print(1);
            m();
        }
        catch( Exception e ) {
            System.out.print (2);
        }
        catch( MyExc3 j ) {
            System.out.print (3);
        }
        catch( MyExc2 w ) {
            System.out.print (4);
        }
    }
    static void m() throws Exception {
        try {
            System.out.print (5);
            throw( new MyExc1() );
        }
    }
}

```

```

        finally {
            System.out.print(6);
        }
    }
}

```

- A. 156Exception in thread main MyExc1
- B. 156
- C. 1562
- 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 MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 f ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc1 f ) {
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

10. Date le dichiarazioni:

```

Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];

```

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

- A. a = (Boolean []) z;
- B. t = (Error []) z;
- C. t = (Error []) a;
- D. a = (Boolean []) t;
- E. Nessuno dei precedenti

11. Qual è l'output di questo programma?

```

class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

12. Quale output si ottiene invocando il metodo q?

```

class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
}

```

```

void m(String s3, Boolean b1, String s5) {
    String s7 = s4.concat("abc");
    if(s3 == s2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s5 == s7) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 110
- B. 001
- C. 100
- D. 010
- E. 000

13. Data il seguente codice:

```

1. public interface I {
2.     int x = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
- B. abstract int x = 0;
- C. protected int x = 0;
- D. public int x;
- E. Nessuna delle precedenti

14. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- D. È possibile dichiarare un attributo senza inizializzarlo
- E. Una classe non può essere dichiarata final

15. Qual'è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 u ) {
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc1 g ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

16. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```

class X {
    static char a;
    static byte b;
    static short c;
    static int d;
    static long e;
    static String s;

    public static void main(String[] args) {
        System.out.println(a+b+c+d+e+s);
    }
}

```

- A. 00000null
- B. 0null
- C. 0
- D. null
- E. Errore di compilazione.

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

Prova n. 10

1. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```
class X {
    static char a;
    static byte b;
    static short c;
    static int d;
    static long e;
    static String s;

    public static void main(String[] args) {
        System.out.println(a+b+c+d+e+s);
    }
}
```

- A. 00000null
- B. 0null
- C. 0
- D. null
- E. Errore di compilazione.

2. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}
```

- A. 125

- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

3. Data il seguente codice:

```
1. public interface I {
2.     int x = 0;
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
- B. abstract int x = 0;
- C. protected int x = 0;
- D. public int x;
- E. Nessuna delle precedenti

4. Qual'è l'output di questo codice?

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



```

    }
    catch( MyExc1 f ) {
    }
    finally {
        System.out.print(5);
    }
}
}

```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- 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 D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 u ) {
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc1 g ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

6. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public void m() {

```

```

        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

7. Date le dichiarazioni:

```

Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];

```

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

- A. a = (Boolean []) z;
- B. t = (Error []) z;
- C. t = (Error []) a;
- D. a = (Boolean []) t;
- 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 B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception e ) {
            System.out.print(2);
        }
        catch( MyExc3 j ) {
            System.out.print(3);
        }
    }
}

```

```

        catch( MyExc2 w ) {
            System.out.print (4);
        }
    }
    static void m() throws Exception {
        try {
            System.out.print (5);
            throw( new MyExc1() );
        }
        finally {
            System.out.print (6);
        }
    }
}

```

- A. 156Exception in thread main MyExc1
- B. 156
- C. 1562
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

9. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print (2);
    }
    Polygon(int i) {
        System.out.print (3);
    }
}
class Rectangle extends Polygon{
    Rectangle() {
        super (6);
        System.out.print (5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print (1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

10. Qual è l'output di questo programma?

```

class A {
    public abstract void f();
    public void m() {

```

```

        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

11. Quale output si ottiene invocando il metodo q?

```

class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {
        String s7 = s4.concat("abc");
        if(s3 == s2) {
            System.out.print (1);
        } else {
            System.out.print (0);
        }
        if(b2 == b1) {
            System.out.print (1);
        } else {
            System.out.print (0);
        }
        if(s5 == s7) {
            System.out.print (1);
        } else {
            System.out.print (0);
        }
    }
}

```

- A. 110
- B. 001
- C. 100
- D. 010
- E. 000

12. Data la seguente classe:

```
1. public class C {
2.     public void m(int i, String s){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. `public int m(int i, String s)`
- B. `public void m(int i, String n)`
- C. `public void M(int i, String s)`
- D. `private void m(int j, String q)`
- E. Nessuna delle precedenti

13. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Un attributo `static` non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- D. È possibile dichiarare un attributo senza inizializzarlo
- E. Una classe non può essere dichiarata `final`

14. Qual è l'output di questo programma?

```
abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

- A. M
- B. F

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

15. Date le dichiarazioni:

```
Boolean p;
Object w;
String z;
```

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

- A. `p = (Boolean) z;`
- B. `p = w;`
- C. `w = z;`
- D. `p = z;`
- E. `z = p;`

16. Quali dei tre oggetti `a1`, `a2` o `a3` sono eleggibili per la garbage collection quando il metodo `m` comincia ad essere eseguito?

```
class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}
class B {
    private A a1 = new A("o1");
    private A a2 = new A("o2");
    private A a3 = new A("o3");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
    private void m() {}
    public static void main (String[] args) {
        new B().f();
    }
}
```

- A. Quello con `s=o1`
 - B. Quello con `s=o2`
 - C. Quello con `s=o3`
 - D. Quelli con `s` uguale a `o1` e `o2`
 - E. Nessuna delle precedenti.
-

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

Prova n. 11

1. Qual è l'output di questo programma?

```
class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

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

2. Qual è l'output di questo programma?

```
abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

- A. M
- B. F
- 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 D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 u ) {
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc1 g ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 f ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc1 f ) {
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. Data il seguente codice:

```

1. public interface I {
2.     int x = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
- B. abstract int x = 0;
- C. protected int x = 0;
- D. public int x;
- E. Nessuna delle precedenti

6. Date le dichiarazioni:

```

Boolean p;
Object w;
String z;

```

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

- A. p = (Boolean) z;
- B. p = w;
- C. w = z;
- D. p = z;
- E. z = p;

7. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle() {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

8. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```

class X {
    static char a;
    static byte b;
    static short c;
    static int d;
    static long e;
    static String s;

    public static void main(String[] args) {
        System.out.println(a+b+c+d+e+s);
    }
}

```

- A. 0000null
- B. 0null
- C. 0
- D. null
- E. Errore di compilazione.

9. Qual'è l'output di questo codice?

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

        try {
            System.out.print(1);
            m();
        }
        catch( Exception e ) {
            System.out.print(2);
        }
        catch( MyExc3 j ) {
            System.out.print(3);
        }
        catch( MyExc2 w ) {
            System.out.print(4);
        }
    }
    static void m() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(6);
        }
    }
}
```

- A. 156Exception in thread main MyExc1
- B. 156
- C. 1562
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

10. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza

- D. È possibile dichiarare un attributo senza iniziarlo
- E. Una classe non può essere dichiarata final

11. Date le dichiarazioni:

```
Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];
```

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

- A. a = (Boolean []) z;
- B. t = (Error []) z;
- C. t = (Error []) a;
- D. a = (Boolean []) t;
- E. Nessuno dei precedenti

12. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}
```

```
class B {
    private A a1 = new A(o1");
    private A a2 = new A(o2");
    private A a3 = new A(o3");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
    private void m() {}
    public static void main (String[] args) {
        new B().f();
    }
}
```

- A. Quello con s=o1
- B. Quello con s=o2
- C. Quello con s=o3
- D. Quelli con s uguale a o1 e o2
- E. Nessuna delle precedenti.

13. Data la seguente classe:

```
1. public class C {
2.     public void m(int i, String s){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. public int m(int i, String s)
 - B. public void m(int i, String n)
 - C. public void M(int i, String s)
 - D. private void m(int j, String q)
 - E. Nessuna delle precedenti
-

14. Quale output si ottiene invocando il metodo q?

```
class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {
        String s7 = s4.concat("abc");
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
 - B. 001
 - C. 100
 - D. 010
 - E. 000
-

15. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}
```

- A. 125
 - B. 135
 - C. 1235
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

16. Qual è l'output di questo programma?

```
abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

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

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

Prova n. 12

1. Qual è l'output di questo programma?

```
abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

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

2. Data la seguente classe:

```
1. public class C {
2.     public void m(int i, String s){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. public int m(int i, String s)
- B. public void m(int i, String n)
- C. public void M(int i, String s)
- D. private void m(int j, String q)
- E. Nessuna delle precedenti

3. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle() {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
```

```

        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. Qual'è l'output di questo codice?

```

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

        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 f ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc1 f ) {
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

6. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}

class B {
    private A a1 = new A(o1");
    private A a2 = new A(o2");
    private A a3 = new A(o3");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
    private void m() {}
    public static void main (String[] args) {
        new B().f();
    }
}

```

- A. Quello con s=o1
- B. Quello con s=o2
- C. Quello con s=o3
- D. Quelli con s uguale a o1 e o2
- 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 D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 u ) {
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc1 g ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

```

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

```

8. Qual è l'output di questo programma?

```

class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

- A. 110
- B. 001
- C. 100
- D. 010
- E. 000

9. Quale output si ottiene invocando il metodo q?

```

class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {
        String s7 = s4.concat("abc");
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {

```

10. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

11. Data il seguente codice:

```

1. public interface I {
2.     int x = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
- B. abstract int x = 0;
- C. protected int x = 0;
- D. public int x;
- E. Nessuna delle precedenti

12. Date le dichiarazioni:

```
Boolean [] a;  
Error [] t;  
Object [] z;  
z = new Boolean [3];  
t = new Error [2];  
a = new Boolean [8];
```

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

- A. a = (Boolean []) z;
- B. t = (Error []) z;
- C. t = (Error []) a;
- D. a = (Boolean []) t;
- E. Nessuno dei precedenti

13. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```
class X {  
    static char a;  
    static byte b;  
    static short c;  
    static int d;  
    static long e;  
    static String s;  
  
    public static void main(String[] args) {  
        System.out.println(a+b+c+d+e+s);  
    }  
}
```

- A. 00000null
- B. 0null
- C. 0
- D. null
- E. Errore di compilazione.

14. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- D. È possibile dichiarare un attributo senza iniziarlo
- E. Una classe non può essere dichiarata final

15. Date le dichiarazioni:

```
Boolean p;  
Object w;  
String z;
```

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

- A. p = (Boolean) z;
- B. p = w;
- C. w = z;
- D. p = z;
- E. z = p;

16. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends MyExc2 { }  
public class Bl {  
    public static void main(String [] argv)  
        throws Exception {  
  
        try {  
            System.out.print(1);  
            m();  
        }  
        catch( Exception e ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 j ) {  
            System.out.print(3);  
        }  
        catch( MyExc2 w ) {  
            System.out.print(4);  
        }  
    }  
    static void m() throws Exception {  
        try {  
            System.out.print(5);  
            throw( new MyExc1() );  
        }  
        finally {  
            System.out.print(6);  
        }  
    }  
}
```

- A. 156Exception in thread main MyExc1
 - B. 156
 - C. 1562
 - 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. 13

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception e ) {
            System.out.print(2);
        }
        catch( MyExc3 j ) {
            System.out.print(3);
        }
        catch( MyExc2 w ) {
            System.out.print(4);
        }
    }
    static void m() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(6);
        }
    }
}
```

- A. 156Exception in thread main MyExc1
- B. 156
- C. 1562
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    private A ref;
    private String s;
```

```
public A(String s) {this.s = s;}
public void setRef(A a) {ref = a;}
}

class B {
    private A a1 = new A(o1");
    private A a2 = new A(o2");
    private A a3 = new A(o3");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
    private void m() {}
    public static void main (String[] args) {
        new B().f();
    }
}
```

- A. Quello con s=o1
- B. Quello con s=o2
- C. Quello con s=o3
- D. Quelli con s uguale a o1 e o2
- E. Nessuna delle precedenti.

3. Data la seguente classe:

```
1. public class C {
2.     public void m(int i, String s){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. public int m(int i, String s)
- B. public void m(int i, String n)
- C. public void M(int i, String s)
- D. private void m(int j, String q)
- E. Nessuna delle precedenti

4. Qual è l'output di questo programma?

```

class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

5. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

6. Date le dichiarazioni:

```

Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];

```

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

- A. a = (Boolean []) z;
- B. t = (Error []) z;
- C. t = (Error []) a;
- D. a = (Boolean []) t;
- E. Nessuno dei precedenti

7. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print (2);
    }
    Polygon(int i) {
        System.out.print (3);
    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super(6);
        System.out.print (5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print (1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

8. Quale output si ottiene invocando il metodo q?

```

class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
    }
}

```



```

        m(s2, new Boolean(true), s6);
    }
}
void m(String s3, Boolean b1, String s5) {
    String s7 = s4.concat("abc");
    if(s3 == s2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s5 == s7) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 110
- B. 001
- C. 100
- D. 010
- E. 000

9. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 f ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc1 f ) {
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

10. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle() {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- 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 D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 u ) {
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc1() );
        }
    }
}

```

```

    }
    catch( MyExcl g ) {
        System.out.print(3);
        throw( new MyExcl() );
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. 23Exception in thread main MyExcl
- B. 234Exception in thread main MyExcl
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. Data il seguente codice:

```

1. public interface I {
2.     int x = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
- B. abstract int x = 0;
- C. protected int x = 0;
- D. public int x;
- E. Nessuna delle precedenti

13. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- D. È possibile dichiarare un attributo senza iniziarlo
- E. Una classe non può essere dichiarata final

14. Date le dichiarazioni:

```

Boolean p;
Object w;
String z;

```

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

- A. p = (Boolean) z;

- B. p = w;
- C. w = z;
- D. p = z;
- E. z = p;

15. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```

class X {
    static char a;
    static byte b;
    static short c;
    static int d;
    static long e;
    static String s;

    public static void main(String[] args) {
        System.out.println(a+b+c+d+e+s);
    }
}

```

- A. 00000null
- B. 0null
- C. 0
- D. null
- E. Errore di compilazione.

16. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

- A. M
- B. F
- 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. 14

1. Qual è l'output di questo programma?

```
class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

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

2. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}
class B {
    private A a1 = new A("o1");
    private A a2 = new A("o2");
    private A a3 = new A("o3");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
    }
}
```

```
a3 = a1; a2 = a1;
m();
}
private void m() {}
public static void main (String[] args) {
    new B().f();
}
}
```

- A. Quello con s=o1
- B. Quello con s=o2
- C. Quello con s=o3
- D. Quelli con s uguale a o1 e o2
- E. Nessuna delle precedenti.

3. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle() {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. Date le dichiarazioni:

```
Boolean p;  
Object w;  
String z;
```

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

- A. `p = (Boolean) z;`
- B. `p = w;`
- C. `w = z;`
- D. `p = z;`
- E. `z = p;`

5. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            q();  
            System.out.print(1);  
        }  
        catch( MyExc2 f ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(3);  
            throw( new MyExc3() );  
        }  
    }  
    static void q() throws Exception {  
        try {  
            System.out.print(4);  
            throw( new MyExc1() );  
        }  
        catch( MyExc1 f ) {  
        }  
        finally {  
            System.out.print(5);  
        }  
    }  
}
```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

```
class MyExc1 extends Error { }  
class MyExc2 extends Error { }  
class MyExc3 extends MyExc2 { }  
public class D1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
            System.out.print(1);  
        }  
        catch( MyExc3 u ) {  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(2);  
            throw( new MyExc1() );  
        }  
        catch( MyExc1 g ) {  
            System.out.print(3);  
            throw( new MyExc1() );  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```
class X {  
    static char a;  
    static byte b;  
    static short c;  
    static int d;  
    static long e;  
    static String s;  
  
    public static void main(String[] args) {  
        System.out.println(a+b+c+d+e+s);  
    }  
}
```

- A. 00000null
- B. 0null
- C. 0
- D. null
- E. Errore di compilazione.

6. Qual'è l'output di questo codice?

8. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- D. È possibile dichiarare un attributo senza inicializzarlo
- E. Una classe non può essere dichiarata final

```

        finally {
            System.out.print(6);
        }
    }
}

```

- A. 156Exception in thread main MyExc1
- B. 156
- C. 1562
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

9. Data la seguente classe:

```

1. public class C {
2.     public void m(int i, String s){}
3.
4. }

```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. public int m(int i, String s)
- B. public void m(int i, String n)
- C. public void M(int i, String s)
- D. private void m(int j, String q)
- E. Nessuna delle precedenti

10. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class Bl {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception e ) {
            System.out.print(2);
        }
        catch( MyExc3 j ) {
            System.out.print(3);
        }
        catch( MyExc2 w ) {
            System.out.print(4);
        }
    }
    static void m() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc1() );
        }
    }
}

```

11. Date le dichiarazioni:

```

Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];

```

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

- A. a = (Boolean []) z;
- B. t = (Error []) z;
- C. t = (Error []) a;
- D. a = (Boolean []) t;
- E. Nessuno dei precedenti

12. Data il seguente codice:

```

1. public interface I {
2.     int x = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
- B. abstract int x = 0;
- C. protected int x = 0;
- D. public int x;
- E. Nessuna delle precedenti

13. Quale output si ottiene invocando il metodo q?

```

class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {
        String s7 = s4.concat("abc");
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 001
- C. 100
- D. 010
- E. 000

14. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

15. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

16. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

- A. M
- B. F

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. 15

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 f ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc1 f ) {
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```
class X {
    static char a;
```

```
static byte b;
static short c;
static int d;
static long e;
static String s;

public static void main(String[] args) {
    System.out.println(a+b+c+d+e+s);
}
```

- A. 00000null
- B. 0null
- C. 0
- D. null
- E. Errore di compilazione.

3. Quale output si ottiene invocando il metodo q?

```
class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {
        String s7 = s4.concat("abc");
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```
}  
}
```

- A. 110
- B. 001
- C. 100
- D. 010
- E. 000

4. Qual è l'output di questo programma?

```
class Polygon {  
    Polygon() {  
        System.out.print(2);  
    }  
    Polygon(int i) {  
        System.out.print(3);  
    }  
}  
class Rectangle extends Polygon{  
    Rectangle(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Square extends Rectangle {  
    public static void main(String argv[]) {  
        System.out.print(1);  
        Square s = new Square();  
    }  
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. Qual è l'output di questo programma?

```
abstract class A {  
    public void f() {}  
    public static void m() {  
        System.out.println("M");  
    }  
}  
class B extends A {  
    public static void main(String argv[]) {  
        B b = new B();  
        b.m();  
    }  
    public void f() {  
        System.out.println("F");  
    }  
    public void m() {
```

```
        f();  
    }  
}
```

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

6. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            m();  
        }  
        catch( Exception e ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 j ) {  
            System.out.print(3);  
        }  
        catch( MyExc2 w ) {  
            System.out.print(4);  
        }  
    }  
    static void m() throws Exception {  
        try {  
            System.out.print(5);  
            throw( new MyExc1() );  
        }  
        finally {  
            System.out.print(6);  
        }  
    }  
}
```

- A. 156Exception in thread main MyExc1
- B. 156
- C. 1562
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Data il seguente codice:

```
1. public interface I {  
2.     int x = 0;  
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
- B. abstract int x = 0;
- C. protected int x = 0;
- D. public int x;
- E. Nessuna delle precedenti

8. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}

class B {
    private A a1 = new A(o1");
    private A a2 = new A(o2");
    private A a3 = new A(o3");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
    private void m() {}
    public static void main (String[] args) {
        new B().f();
    }
}
```

- A. Quello con s=o1
- B. Quello con s=o2
- C. Quello con s=o3
- D. Quelli con s uguale a o1 e o2
- E. Nessuna delle precedenti.

9. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- D. È possibile dichiarare un attributo senza inicializzarlo
- E. Una classe non può essere dichiarata final

10. Qual è l'output di questo programma?

```
abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}

class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

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

11. Date le dichiarazioni:

```
Boolean p;
Object w;
String z;
```

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

- A. p = (Boolean) z;
- B. p = w;
- C. w = z;
- D. p = z;
- E. z = p;

12. Data la seguente classe:

```
1. public class C {
2.     public void m(int i, String s){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. public int m(int i, String s)
- B. public void m(int i, String n)
- C. public void M(int i, String s)

- D. private void m(int j, String q)
- E. Nessuna delle precedenti

13. Qual'è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 u ) {
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc1 g ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

14. Date le dichiarazioni:

```
Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];
```

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

- A. a = (Boolean []) z;
- B. t = (Error []) z;
- C. t = (Error []) a;
- D. a = (Boolean []) t;
- E. Nessuno dei precedenti

15. Qual è l'output di questo programma?

```
class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

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

16. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle() {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}
```

- A. 125
- B. 135
- C. 1235
- 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. 16

1. Quale output si ottiene invocando il metodo q?

```
class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {
        String s7 = s4.concat("abc");
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 001
- C. 100
- D. 010
- E. 000

2. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
}
```

```
public void setRef(A a) {ref = a;}
}

class B {
    private A a1 = new A(o1");
    private A a2 = new A(o2");
    private A a3 = new A(o3");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
    private void m() {}
    public static void main (String[] args) {
        new B().f();
    }
}
```

- A. Quello con s=o1
- B. Quello con s=o2
- C. Quello con s=o3
- D. Quelli con s uguale a o1 e o2
- E. Nessuna delle precedenti.

3. Qual'è l'output di questo codice?

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

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

```

}
static void q() throws Exception {
    try {
        System.out.print(4);
        throw( new MyExcl() );
    }
    catch( MyExcl f ) {
    }
    finally {
        System.out.print(5);
    }
}
}

```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```

class X {
    static char a;
    static byte b;
    static short c;
    static int d;
    static long e;
    static String s;

    public static void main(String[] args) {
        System.out.println(a+b+c+d+e+s);
    }
}

```

- A. 00000null
- B. 0null
- C. 0
- D. null
- E. Errore di compilazione.

5. Data la seguente classe:

```

1. public class C {
2.     public void m(int i, String s){}
3.
4. }

```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. public int m(int i, String s)
- B. public void m(int i, String n)

- C. public void M(int i, String s)
- D. private void m(int j, String q)
- E. Nessuna delle precedenti

6. Qual è l'output di questo programma?

```

class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

7. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

8. Qual è l'output di questo programma?

```
abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

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

9. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

10. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- D. È possibile dichiarare un attributo senza iniziarlo
- E. Una classe non può essere dichiarata final

11. Qual'è l'output di questo codice?

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

        try {
            System.out.print(1);
            m();
        }
        catch( Exception e ) {
            System.out.print(2);
        }
        catch( MyExc3 j ) {
            System.out.print(3);
        }
        catch( MyExc2 w ) {
            System.out.print(4);
        }
    }
    static void m() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(6);
        }
    }
}
```

- A. 156Exception in thread main MyExc1
- B. 156
- C. 1562
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. Data il seguente codice:

```

1. public interface I {
2.     int x = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
- B. abstract int x = 0;
- C. protected int x = 0;
- D. public int x;
- E. Nessuna delle precedenti

13. Date le dichiarazioni:

```

Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];

```

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

- A. a = (Boolean []) z;
- B. t = (Error []) z;
- C. t = (Error []) a;
- D. a = (Boolean []) t;
- E. Nessuno dei precedenti

14. Date le dichiarazioni:

```

Boolean p;
Object w;
String z;

```

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

- A. p = (Boolean) z;
- B. p = w;
- C. w = z;
- D. p = z;
- E. z = p;

15. Qual'è l'output di questo codice?

```

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

```

```

        p();
        System.out.print(1);
    }
    catch( MyExc3 u ) {
    }
}
static void p() {
    try {
        System.out.print(2);
        throw( new MyExc1() );
    }
    catch( MyExc1 g ) {
        System.out.print(3);
        throw( new MyExc1() );
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

16. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle() {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

Prova n. 17

1. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}

class B {
    private A a1 = new A("o1");
    private A a2 = new A("o2");
    private A a3 = new A("o3");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
    private void m() {}
    public static void main (String[] args) {
        new B().f();
    }
}
```

- A. Quello con s=o1
- B. Quello con s=o2
- C. Quello con s=o3
- D. Quelli con s uguale a o1 e o2
- E. Nessuna delle precedenti.

2. Qual'è l'output di questo codice?

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

```
catch( MyExc2 f ) {
    System.out.print(2);
    throw( new MyExc3() );
}
finally {
    System.out.print(3);
    throw( new MyExc3() );
}
}
static void q() throws Exception {
    try {
        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc1 f ) {
    }
    finally {
        System.out.print(5);
    }
}
}
```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

3. Qual è l'output di questo programma?

```
class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

```
}  
}
```

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

4. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- D. È possibile dichiarare un attributo senza iniziarlo
- E. Una classe non può essere dichiarata final

5. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```
class X {  
    static char a;  
    static byte b;  
    static short c;  
    static int d;  
    static long e;  
    static String s;  
  
    public static void main(String[] args) {  
        System.out.println(a+b+c+d+e+s);  
    }  
}
```

- A. 00000null
- B. 0null
- C. 0
- D. null
- E. Errore di compilazione.

6. Data il seguente codice:

```
1. public interface I {  
2.     int x = 0;  
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
- B. abstract int x = 0;
- C. protected int x = 0;
- D. public int x;
- E. Nessuna delle precedenti

7. Qual è l'output di questo programma?

```
abstract class A {  
    public void f() {}  
    public void m() {  
        System.out.println("M");  
    }  
}  
class B extends A {  
    public static void main(String argv[]) {  
        B b = new B();  
        b.m();  
    }  
    public void f() {  
        System.out.println("F");  
    }  
    public void m() {  
        f();  
    }  
}
```

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

8. Data la seguente classe:

```
1. public class C {  
2.     public void m(int i, String s){}  
3.  
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. public int m(int i, String s)
- B. public void m(int i, String n)
- C. public void M(int i, String s)
- D. private void m(int j, String q)
- E. Nessuna delle precedenti

9. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception e ) {
            System.out.print(2);
        }
        catch( MyExc3 j ) {
            System.out.print(3);
        }
        catch( MyExc2 w ) {
            System.out.print(4);
        }
    }
    static void m() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(6);
        }
    }
}

```

- A. 156Exception in thread main MyExc1
- B. 156
- C. 1562
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

10. Qual'è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 u ) {
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc1 g ) {
            System.out.print(3);

```

```

        throw( new MyExc1() );
    }
    finally {
        System.out.print(4);
    }
}

```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

11. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle() {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {

```

```

        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

13. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

14. Quale output si ottiene invocando il metodo q?

```

class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {
        String s7 = s4.concat("abc");
        if(s3 == s2) {
            System.out.print(1);

```

```

        } else {
            System.out.print(0);
        }
    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s5 == s7) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 110
- B. 001
- C. 100
- D. 010
- E. 000

15. Date le dichiarazioni:

```

Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];

```

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

- A. a = (Boolean []) z;
- B. t = (Error []) z;
- C. t = (Error []) a;
- D. a = (Boolean []) t;
- E. Nessuno dei precedenti

16. Date le dichiarazioni:

```

Boolean p;
Object w;
String z;

```

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

- A. p = (Boolean) z;
- B. p = w;
- C. w = z;
- D. p = z;
- E. z = p;

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. Data la seguente classe:

```
1. public class C {
2.     public void m(int i, String s){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. public int m(int i, String s)
- B. public void m(int i, String n)
- C. public void M(int i, String s)
- D. private void m(int j, String q)
- 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 Bl {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception e ) {
            System.out.print(2);
        }
        catch( MyExc3 j ) {
            System.out.print(3);
        }
        catch( MyExc2 w ) {
            System.out.print(4);
        }
    }
    static void m() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(6);
        }
    }
}
```

- A. 156Exception in thread main MyExc1
- B. 156
- C. 1562
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

3. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
```

```

    }
}
class Rectangle extends Polygon{
    Rectangle() {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- D. È possibile dichiarare un attributo senza inizializzarlo
- E. Una classe non può essere dichiarata final

6. Qual'è l'output di questo codice?

```

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

```

```

}
static void q() throws Exception {
    try {
        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc1 f ) {
    }
    finally {
        System.out.print(5);
    }
}
}

```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Data il seguente codice:

```

1. public interface I {
2.     int x = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
- B. abstract int x = 0;
- C. protected int x = 0;
- D. public int x;
- E. Nessuna delle precedenti

8. Date le dichiarazioni:

```

Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];

```

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

- A. a = (Boolean []) z;
- B. t = (Error []) z;
- C. t = (Error []) a;
- D. a = (Boolean []) t;
- E. Nessuno dei precedenti

9. Qual'è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 u ) {
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc1 g ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

10. Date le dichiarazioni:

```

Boolean p;
Object w;
String z;

```

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

- A. p = (Boolean) z;
- B. p = w;
- C. w = z;
- D. p = z;
- E. z = p;

11. Quale output si ottiene invocando il metodo q?

```

class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";

```

```

void q() {
    String s6;
    s6 = s4.concat("abc");
    m(s2, new Boolean(true), s6);
}
void m(String s3, Boolean b1, String s5) {
    String s7 = s4.concat("abc");
    if(s3 == s2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s5 == s7) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 110
- B. 001
- C. 100
- D. 010
- E. 000

12. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```

class X {
    static char a;
    static byte b;
    static short c;
    static int d;
    static long e;
    static String s;

    public static void main(String[] args) {
        System.out.println(a+b+c+d+e+s);
    }
}

```

- A. 00000null
- B. 0null
- C. 0
- D. null
- E. Errore di compilazione.

13. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}

class B {
    private A a1 = new A(o1");
    private A a2 = new A(o2");
    private A a3 = new A(o3");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
    private void m() {}
    public static void main (String[] args) {
        new B().f();
    }
}

```

- A. Quello con s=o1
- B. Quello con s=o2
- C. Quello con s=o3
- D. Quelli con s uguale a o1 e o2
- E. Nessuna delle precedenti.

14. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}

class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

15. Qual è l'output di questo programma?

```

class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}

class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

16. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}

class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

- A. M
 - B. F
 - 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. 19

1. Qual è l'output di questo programma?

```
abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

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

2. Date le dichiarazioni:

```
Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];
```

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

- A. a = (Boolean []) z;
- B. t = (Error []) z;
- C. t = (Error []) a;
- D. a = (Boolean []) t;
- E. Nessuno dei precedenti

3. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. Qual è l'output di questo programma?

```
class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
```

```

        f();
    }
}

```

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

5. Data la seguente classe:

```

1. public class C {
2.     public void m(int i, String s){}
3.
4. }

```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. public int m(int i, String s)
- B. public void m(int i, String n)
- C. public void M(int i, String s)
- D. private void m(int j, String q)
- E. Nessuna delle precedenti

6. Data il seguente codice:

```

1. public interface I {
2.     int x = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
- B. abstract int x = 0;
- C. protected int x = 0;
- D. public int x;
- E. Nessuna delle precedenti

7. Qual'è l'output di questo codice?

```

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

```

```

catch( MyExc2 f ) {
    System.out.print(2);
    throw( new MyExc3() );
}
finally {
    System.out.print(3);
    throw( new MyExc3() );
}
}
static void q() throws Exception {
    try {
        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc1 f ) {
    }
    finally {
        System.out.print(5);
    }
}
}
}

```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

8. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```

class X {
    static char a;
    static byte b;
    static short c;
    static int d;
    static long e;
    static String s;

    public static void main(String[] args) {
        System.out.println(a+b+c+d+e+s);
    }
}

```

- A. 00000null
- B. 0null
- C. 0
- D. null
- E. Errore di compilazione.

9. Qual'è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 u ) {
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc1 g ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

10. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

- A. M
- B. F
- C. Errore a tempo di compilazione

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

11. Date le dichiarazioni:

```

Boolean p;
Object w;
String z;

```

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

- A. p = (Boolean) z;
- B. p = w;
- C. w = z;
- D. p = z;
- E. z = p;

12. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle() {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

13. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi

- C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- D. È possibile dichiarare un attributo senza inizializzarlo
- E. Una classe non può essere dichiarata final

14. Quale output si ottiene invocando il metodo q?

```
class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {
        String s7 = s4.concat("abc");
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 001
- C. 100
- D. 010
- E. 000

15. Qual'è l'output di questo codice?

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

```
}
catch( MyExc3 j ) {
    System.out.print(3);
}
catch( MyExc2 w ) {
    System.out.print(4);
}
}
static void m() throws Exception {
    try {
        System.out.print(5);
        throw( new MyExc1() );
    }
    finally {
        System.out.print(6);
    }
}
}
```

- A. 156Exception in thread main MyExc1
- B. 156
- C. 1562
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

16. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}

class B {
    private A a1 = new A(o1");
    private A a2 = new A(o2");
    private A a3 = new A(o3");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
    private void m() {}
    public static void main (String[] args) {
        new B().f();
    }
}
```

- A. Quello con s=o1
- B. Quello con s=o2
- C. Quello con s=o3
- D. Quelli con s uguale a o1 e o2
- 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. Data il seguente codice:

```
1. public interface I {
2.     int x = 0;
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
- B. abstract int x = 0;
- C. protected int x = 0;
- D. public int x;
- E. Nessuna delle precedenti

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 f ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc1 f ) {
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

3. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```
class X {
    static char a;
    static byte b;
    static short c;
```

```

static int d;
static long e;
static String s;

public static void main(String[] args) {
    System.out.println(a+b+c+d+e+s);
}
}

```

- A. 00000null
- B. 0null
- C. 0
- D. null
- E. Errore di compilazione.

5. Date le dichiarazioni:

```

Boolean p;
Object w;
String z;

```

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

- A. p = (Boolean) z;
- B. p = w;
- C. w = z;
- D. p = z;
- E. z = p;

6. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- D. È possibile dichiarare un attributo senza iniziarlo
- E. Una classe non può essere dichiarata final

7. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}
class B extends A {

```

```

public static void main(String argv[]) {
    B b = new B();
    b.m();
}
public void f() {
    System.out.println("F");
}
public void m() {
    f();
}
}

```

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

8. Qual è l'output di questo programma?

```

class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

9. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle() {

```

```

        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

10. Date le dichiarazioni:

```

Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];

```

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

- A. a = (Boolean []) z;
- B. t = (Error []) z;
- C. t = (Error []) a;
- D. a = (Boolean []) t;
- E. Nessuno dei precedenti

11. Qual'è l'output di questo codice?

```

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

```

```

    }
}
static void m() throws Exception {
    try {
        System.out.print(5);
        throw( new MyExc1() );
    }
    finally {
        System.out.print(6);
    }
}
}

```

- A. 156Exception in thread main MyExc1
- B. 156
- C. 1562
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}

```

```

class B {
    private A a1 = new A(o1");
    private A a2 = new A(o2");
    private A a3 = new A(o3");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
    private void m() {}
    public static void main (String[] args) {
        new B().f();
    }
}

```

- A. Quello con s=o1
- B. Quello con s=o2
- C. Quello con s=o3
- D. Quelli con s uguale a o1 e o2
- E. Nessuna delle precedenti.

13. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

14. Qual'è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 u ) {
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc1 g ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione

E. Nessuna delle precedenti

15. Quale output si ottiene invocando il metodo q?

```

class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {
        String s7 = s4.concat("abc");
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 001
- C. 100
- D. 010
- E. 000

16. Data la seguente classe:

```

1. public class C {
2.     public void m(int i, String s){}
3.
4. }

```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. public int m(int i, String s)
- B. public void m(int i, String n)
- C. public void M(int i, String s)
- D. private void m(int j, String q)
- 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. Quale output si ottiene invocando il metodo q?

```
class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {
        String s7 = s4.concat("abc");
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 001
- C. 100
- D. 010
- E. 000

2. Data il seguente codice:

```
1. public interface I {
2.     int x = 0;
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
- B. abstract int x = 0;
- C. protected int x = 0;
- D. public int x;
- E. Nessuna delle precedenti

3. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- D. È possibile dichiarare un attributo senza inicializzarlo
- E. Una classe non può essere dichiarata final

4. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}
```

```
class B {
    private A a1 = new A("01");
    private A a2 = new A("02");
    private A a3 = new A("03");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
    private void m() {}
}
```



```

    public static void main (String[] args) {
        new B().f();
    }
}

```

- A. Quello con s=o1
- B. Quello con s=o2
- C. Quello con s=o3
- D. Quelli con s uguale a o1 e o2
- E. Nessuna delle precedenti.

5. Date le dichiarazioni:

```

Boolean p;
Object w;
String z;

```

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

- A. p = (Boolean) z;
- B. p = w;
- C. w = z;
- D. p = z;
- E. z = p;

6. Qual'è l'output di questo codice?

```

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

```

```

        System.out.print(5);
    }
}

```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

8. Qual è l'output di questo programma?

```

class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

9. Qual'è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 u ) {
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc1 g ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

10. Qual'è l'output di questo codice?

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

```
}
catch( MyExc3 j ) {
    System.out.print(3);
}
catch( MyExc2 w ) {
    System.out.print(4);
}
}
static void m() throws Exception {
    try {
        System.out.print(5);
        throw( new MyExc1() );
    }
    finally {
        System.out.print(6);
    }
}
}
```

- A. 156Exception in thread main MyExc1
- B. 156
- C. 1562
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

11. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```
class X {
    static char a;
    static byte b;
    static short c;
    static int d;
    static long e;
    static String s;

    public static void main(String[] args) {
        System.out.println(a+b+c+d+e+s);
    }
}
```

- A. 00000null
- B. 0null
- C. 0
- D. null
- E. Errore di compilazione.

12. Data la seguente classe:

```
1. public class C {
2.     public void m(int i, String s){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. public int m(int i, String s)
- B. public void m(int i, String n)
- C. public void M(int i, String s)
- D. private void m(int j, String q)
- E. Nessuna delle precedenti

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

13. Qual è l'output di questo programma?

```
abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

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

14. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}
```

15. Date le dichiarazioni:

```
Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];
```

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

- A. a = (Boolean []) z;
- B. t = (Error []) z;
- C. t = (Error []) a;
- D. a = (Boolean []) t;
- E. Nessuno dei precedenti

16. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle() {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}
```

- A. 125
 - B. 135
 - C. 1235
 - 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. 22

1. Data il seguente codice:

```
1. public interface I {
2.     int x = 0;
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
- B. abstract int x = 0;
- C. protected int x = 0;
- D. public int x;
- E. Nessuna delle precedenti

2. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```
class X {
    static char a;
    static byte b;
    static short c;
    static int d;
    static long e;
    static String s;

    public static void main(String[] args) {
        System.out.println(a+b+c+d+e+s);
    }
}
```

- A. 00000null
- B. 0null
- C. 0
- D. null
- E. Errore di compilazione.

3. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
}
```

```
Polygon(int i) {
    System.out.print(3);
}

class Rectangle extends Polygon{
    Rectangle() {
        super(6);
        System.out.print(5);
    }
}

class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}
```

```
class B {
    private A a1 = new A("01");
    private A a2 = new A("02");
    private A a3 = new A("03");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
    private void m() {}
}
```

```

    public static void main (String[] args) {
        new B().f();
    }
}

```

- A. Quello con s=o1
- B. Quello con s=o2
- C. Quello con s=o3
- D. Quelli con s uguale a o1 e o2
- E. Nessuna delle precedenti.

5. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

6. Qual è l'output di questo programma?

```

class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

- A. M
- B. F
- 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 D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 u ) {
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc1 g ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

8. Quale output si ottiene invocando il metodo q?

```

class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {

```

```

String s7 = s4.concat("abc");
if(s3 == s2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(b2 == b1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s5 == s7) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
}

```

- A. 110
- B. 001
- C. 100
- D. 010
- E. 000

9. Qual'è l'output di questo codice?

```

class MyExcl extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception e ) {
            System.out.print(2);
        }
        catch( MyExc3 j ) {
            System.out.print(3);
        }
        catch( MyExc2 w ) {
            System.out.print(4);
        }
    }
    static void m() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExcl() );
        }
        finally {
            System.out.print(6);
        }
    }
}

```

- A. 156Exception in thread main MyExcl
- B. 156
- C. 1562
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

10. Date le dichiarazioni:

```

Boolean p;
Object w;
String z;

```

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

- A. p = (Boolean) z;
- B. p = w;
- C. w = z;
- D. p = z;
- E. z = p;

11. Data la seguente classe:

```

1. public class C {
2.     public void m(int i, String s){}
3.
4. }

```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. public int m(int i, String s)
- B. public void m(int i, String n)
- C. public void M(int i, String s)
- D. private void m(int j, String q)
- E. Nessuna delle precedenti

12. Date le dichiarazioni:

```

Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];

```

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

- A. a = (Boolean []) z;
- B. t = (Error []) z;
- C. t = (Error []) a;
- D. a = (Boolean []) t;
- E. Nessuno dei precedenti

13. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 f ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc1 f ) {
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 4513222222... (ciclo infinito)
 - B. 45132Exception in thread main MyExc3
 - C. 41
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

14. Qual è l'output di questo programma?

```
abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
```

```
        f();
    }
}
```

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

15. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
 - B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
 - C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
 - D. È possibile dichiarare un attributo senza iniziarlo
 - E. Una classe non può essere dichiarata final
-

16. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}
```

- A. 125
 - B. 135
 - C. 1235
 - 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. 23

1. Qual è l'output di questo programma?

```
abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

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

2. Date le dichiarazioni:

```
Boolean p;
Object w;
String z;
```

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

- A. `p = (Boolean) z;`
- B. `p = w;`
- C. `w = z;`
- D. `p = z;`
- E. `z = p;`

3. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle() {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```
class X {
    static char a;
    static byte b;
    static short c;
    static int d;
    static long e;
    static String s;

    public static void main(String[] args) {
        System.out.println(a+b+c+d+e+s);
    }
}
```

- A. 00000null
- B. 0null
- C. 0

- D. null
- E. Errore di compilazione.

5. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

6. Qual'è l'output di questo codice?

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

```
        throw( new MyExc1() );
        }
        catch( MyExc1 f ) {
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Qual è l'output di questo programma?

```
abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

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

8. Qual'è l'output di questo codice?

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

```

        catch( MyExc3 j ) {
            System.out.print(3);
        }
        catch( MyExc2 w ) {
            System.out.print(4);
        }
    }
    static void m() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(6);
        }
    }
}

```

- A. 156Exception in thread main MyExc1
- B. 156
- C. 1562
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

9. Quale output si ottiene invocando il metodo q?

```

class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {
        String s7 = s4.concat("abc");
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 001

- C. 100
- D. 010
- E. 000

10. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- D. È possibile dichiarare un attributo senza inicializzarlo
- E. Una classe non può essere dichiarata final

11. Qual è l'output di questo programma?

```

class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

12. Data la seguente classe:

```

1. public class C {
2.     public void m(int i, String s){}
3.
4. }

```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. public int m(int i, String s)
- B. public void m(int i, String n)
- C. public void M(int i, String s)
- D. private void m(int j, String q)
- E. Nessuna delle precedenti

13. Date le dichiarazioni:

```
Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];
```

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

- A. a = (Boolean []) z;
- B. t = (Error []) z;
- C. t = (Error []) a;
- D. a = (Boolean []) t;
- 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 D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 u ) {
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc1 g ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1

- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

15. Data il seguente codice:

```
1. public interface I {
2.     int x = 0;
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
- B. abstract int x = 0;
- C. protected int x = 0;
- D. public int x;
- E. Nessuna delle precedenti

16. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}

class B {
    private A a1 = new A(o1");
    private A a2 = new A(o2");
    private A a3 = new A(o3");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
    private void m() {}
    public static void main (String[] args) {
        new B().f();
    }
}
```

- A. Quello con s=o1
- B. Quello con s=o2
- C. Quello con s=o3
- D. Quelli con s uguale a o1 e o2
- E. Nessuna delle precedenti.

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

Prova n. 24

1. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle() {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Quale output si ottiene invocando il metodo q?

```
class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {
        String s7 = s4.concat("abc");
        if(s3 == s2) {
            System.out.print(1);
        } else {
```

```
        System.out.print(0);
    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s5 == s7) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 110
- B. 001
- C. 100
- D. 010
- E. 000

3. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Un attributo `static` non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- D. È possibile dichiarare un attributo senza inicializzarlo
- E. Una classe non può essere dichiarata `final`

4. Qual è l'output di questo programma?

```
abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
```

```

        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

5. Data il seguente codice:

```

1. public interface I {
2.     int x = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
- B. abstract int x = 0;
- C. protected int x = 0;
- D. public int x;
- E. Nessuna delle precedenti

6. Qual'è l'output di questo codice?

```

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

```

```

            System.out.print(5);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(6);
        }
    }
}

```

- A. 156Exception in thread main MyExc1
- B. 156
- C. 1562
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```

class X {
    static char a;
    static byte b;
    static short c;
    static int d;
    static long e;
    static String s;

    public static void main(String[] args) {
        System.out.println(a+b+c+d+e+s);
    }
}

```

- A. 00000null
- B. 0null
- C. 0
- D. null
- E. Errore di compilazione.

8. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}

```

```

class B {
    private A a1 = new A("01");
    private A a2 = new A("02");
    private A a3 = new A("03");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
    }
}

```



```

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

```

- A. Quello con s=o1
- B. Quello con s=o2
- C. Quello con s=o3
- D. Quelli con s uguale a o1 e o2
- E. Nessuna delle precedenti.

9. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print (2);
    }
    Polygon(int i) {
        System.out.print (3);
    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super (6);
        System.out.print (5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print (1);
        Square s = new Square ();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

10. Data la seguente classe:

```

1. public class C {
2.     public void m(int i, String s){}
3.
4. }

```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. public int m(int i, String s)
- B. public void m(int i, String n)

- C. public void M(int i, String s)
- D. private void m(int j, String q)
- E. Nessuna delle precedenti

11. Qual è l'output di questo programma?

```

class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

12. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

13. Qual'è l'output di questo codice?

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

        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 f ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc1 f ) {
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

14. Qual'è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 u ) {
        }
    }
    static void p() {
        try {
```

```
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc1 g ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

15. Date le dichiarazioni:

```
Boolean p;
Object w;
String z;
```

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

- A. p = (Boolean) z;
- B. p = w;
- C. w = z;
- D. p = z;
- E. z = p;

16. Date le dichiarazioni:

```
Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];
```

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

- A. a = (Boolean []) z;
 - B. t = (Error []) z;
 - C. t = (Error []) a;
 - D. a = (Boolean []) t;
 - E. Nessuno dei precedenti
-

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

Prova n. 25

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 f ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc1 f ) {
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Qual è l'output di questo programma?

```
class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}
```

```
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

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

3. Quale output si ottiene invocando il metodo q?

```
class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {
        String s7 = s4.concat("abc");
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
```

```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 110
- B. 001
- C. 100
- D. 010
- E. 000

4. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle() {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. Qual'è l'output di questo codice?

```

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

```

```

        System.out.print(2);
    }
    catch( MyExc3 j ) {
        System.out.print(3);
    }
    catch( MyExc2 w ) {
        System.out.print(4);
    }
}
static void m() throws Exception {
    try {
        System.out.print(5);
        throw( new MyExc1() );
    }
    finally {
        System.out.print(6);
    }
}
}

```

- A. 156Exception in thread main MyExc1
- B. 156
- C. 1562
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

6. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

- A. M
- B. F
- 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 D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 u ) {
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc1 g ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

8. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Un attributo **static** non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- D. È possibile dichiarare un attributo senza inizializzarlo
- E. Una classe non può essere dichiarata **final**

9. Date le dichiarazioni:

```

Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];

```

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

- A. a = (Boolean []) z;
- B. t = (Error []) z;
- C. t = (Error []) a;
- D. a = (Boolean []) t;
- E. Nessuno dei precedenti

10. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

11. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

12. Date le dichiarazioni:

```
Boolean p;  
Object w;  
String z;
```

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

- A. `p = (Boolean) z;`
- B. `p = w;`
- C. `w = z;`
- D. `p = z;`
- E. `z = p;`

13. Data il seguente codice:

```
1. public interface I {  
2.     int x = 0;  
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. `final int x = 0;`
- B. `abstract int x = 0;`
- C. `protected int x = 0;`
- D. `public int x;`
- E. Nessuna delle precedenti

14. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```
class X {  
    static char a;  
    static byte b;  
    static short c;  
    static int d;  
    static long e;  
    static String s;  
  
    public static void main(String[] args) {  
        System.out.println(a+b+c+d+e+s);  
    }  
}
```

- A. 00000null
- B. 0null

- C. 0
- D. null
- E. Errore di compilazione.

15. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {  
    private A ref;  
    private String s;  
    public A(String s) {this.s = s;}  
    public void setRef(A a) {ref = a;}  
}
```

```
class B {  
    private A a1 = new A("o1");  
    private A a2 = new A("o2");  
    private A a3 = new A("o3");  
    private void f() {  
        a1.setRef(a2);  
        a2.setRef(a1);  
        a3.setRef(a3);  
        a3 = a1; a2 = a1;  
        m();  
    }  
    private void m() {}  
    public static void main (String[] args) {  
        new B().f();  
    }  
}
```

- A. Quello con s=o1
- B. Quello con s=o2
- C. Quello con s=o3
- D. Quelli con s uguale a o1 e o2
- E. Nessuna delle precedenti.

16. Data la seguente classe:

```
1. public class C {  
2.     public void m(int i, String s){}  
3.  
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. `public int m(int i, String s)`
- B. `public void m(int i, String n)`
- C. `public void M(int i, String s)`
- D. `private void m(int j, String q)`
- E. Nessuna delle precedenti

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

Prova n. 26

1. Qual è l'output di questo programma?

```
abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

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

2. Qual'è l'output di questo codice?

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

```
        System.out.print(4);
    }
}
static void m() throws Exception {
    try {
        System.out.print(5);
        throw( new MyExc1() );
    }
    finally {
        System.out.print(6);
    }
}
```

- A. 156Exception in thread main MyExc1
- B. 156
- C. 1562
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

3. Date le dichiarazioni:

```
Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];
```

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

- A. a = (Boolean []) z;
- B. t = (Error []) z;
- C. t = (Error []) a;
- D. a = (Boolean []) t;
- E. Nessuno dei precedenti

4. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
```

```

    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- 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 D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 u ) {
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc1 g ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

6. Qual è l'output di questo programma?

```

class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

7. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle() {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

8. Quale output si ottiene invocando il metodo q?

```

class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {
        String s7 = s4.concat("abc");
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 001
- C. 100
- D. 010
- E. 000

9. Data il seguente codice:

```

1. public interface I {
2.     int x = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
- B. abstract int x = 0;
- C. protected int x = 0;
- D. public int x;
- E. Nessuna delle precedenti

10. Date le dichiarazioni:

```

Boolean p;
Object w;
String z;

```

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

- A. p = (Boolean) z;
- B. p = w;
- C. w = z;
- D. p = z;
- E. z = p;

11. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 f ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc1 f ) {
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

-
13. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}
class B {
    private A a1 = new A("o1");
    private A a2 = new A("o2");
    private A a3 = new A("o3");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
    private void m() {}
    public static void main (String[] args) {
        new B().f();
    }
}

```

- A. Quello con s=o1
- B. Quello con s=o2
- C. Quello con s=o3
- D. Quelli con s uguale a o1 e o2
- E. Nessuna delle precedenti.

-
14. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```

class X {
    static char a;
    static byte b;
    static short c;
    static int d;
    static long e;
    static String s;

    public static void main(String[] args) {
        System.out.println(a+b+c+d+e+s);
    }
}

```

- A. 00000null
- B. 0null
- C. 0
- D. null
- E. Errore di compilazione.

-
15. Data la seguente classe:

```

1. public class C {
2.     public void m(int i, String s){}
3.
4. }

```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. public int m(int i, String s)
- B. public void m(int i, String n)
- C. public void M(int i, String s)
- D. private void m(int j, String g)
- E. Nessuna delle precedenti

-
16. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
 - B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
 - C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
 - D. È possibile dichiarare un attributo senza inicializzarlo
 - E. Una classe non può essere dichiarata final
-

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

Prova n. 27

1. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print (2);
    }
    Polygon(int i) {
        System.out.print (3);
    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super (6);
        System.out.print (5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print (1);
        Square s = new Square();
    }
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Date le dichiarazioni:

```
Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];
```

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

- A. a = (Boolean []) z;
- B. t = (Error []) z;
- C. t = (Error []) a;
- D. a = (Boolean []) t;

E. Nessuno dei precedenti

3. Data il seguente codice:

```
1. public interface I {
2.     int x = 0;
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
- B. abstract int x = 0;
- C. protected int x = 0;
- D. public int x;
- E. Nessuna delle precedenti

4. Qual è l'output di questo programma?

```
abstract class A {
    public void f() {}
    public void m() {
        System.out.println ("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println ("F");
    }
    public void m() {
        f();
    }
}
```

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

5. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 f ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc1 f ) {
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

6. Qual'è l'output di questo codice?

```

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

```

```

}
}
static void m() throws Exception {
    try {
        System.out.print(5);
        throw( new MyExc1() );
    }
    finally {
        System.out.print(6);
    }
}
}

```

- A. 156Exception in thread main MyExc1
- B. 156
- C. 1562
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- D. È possibile dichiarare un attributo senza iniziarlo
- E. Una classe non può essere dichiarata final

8. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}

class B {
    private A a1 = new A("01");
    private A a2 = new A("02");
    private A a3 = new A("03");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
}

```

```

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

```

- A. Quello con s=o1
- B. Quello con s=o2
- C. Quello con s=o3
- D. Quelli con s uguale a o1 e o2
- E. Nessuna delle precedenti.

9. Qual è l'output di questo programma?

```

class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

10. Data la seguente classe:

```

1. public class C {
2.     public void m(int i, String s){}
3.
4. }

```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. public int m(int i, String s)
- B. public void m(int i, String n)
- C. public void M(int i, String s)
- D. private void m(int j, String q)
- 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 D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 u ) {
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc1 g ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. Quale output si ottiene invocando il metodo q?

```

class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {
        String s7 = s4.concat("abc");
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```



```

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

```

- A. 110
- B. 001
- C. 100
- D. 010
- E. 000

13. Date le dichiarazioni:

```

Boolean p;
Object w;
String z;

```

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

- A. `p = (Boolean) z;`
- B. `p = w;`
- C. `w = z;`
- D. `p = z;`
- E. `z = p;`

14. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```

class X {
    static char a;
    static byte b;
    static short c;
    static int d;
    static long e;
    static String s;

    public static void main(String[] args) {
        System.out.println(a+b+c+d+e+s);
    }
}

```

- A. 00000null
- B. 0null
- C. 0
- D. null
- E. Errore di compilazione.

15. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle() {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

16. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

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

Prova n. 28

1. Date le dichiarazioni:

```
Boolean p;
Object w;
String z;
```

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

- A. `p = (Boolean) z;`
- B. `p = w;`
- C. `w = z;`
- D. `p = z;`
- E. `z = p;`

2. Data la seguente classe:

```
1. public class C {
2.     public void m(int i, String s){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. `public int m(int i, String s)`
- B. `public void m(int i, String n)`
- C. `public void M(int i, String s)`
- D. `private void m(int j, String s)`
- E. Nessuna delle precedenti

3. Qual è l'output di questo programma?

```
abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
}
```

```
public void f() {
    System.out.println("F");
}
public void m() {
    f();
}
}
```

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

4. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```
class X {
    static char a;
    static byte b;
    static short c;
    static int d;
    static long e;
    static String s;

    public static void main(String[] args) {
        System.out.println(a+b+c+d+e+s);
    }
}
```

- A. 00000null
- B. 0null
- C. 0
- D. null
- E. Errore di compilazione.

5. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

6. Qual è l'output di questo programma?

```

class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

7. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception e ) {
            System.out.print(2);
        }
        catch( MyExc3 j ) {
            System.out.print(3);
        }
        catch( MyExc2 w ) {
            System.out.print(4);
        }
    }
    static void m() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(6);
        }
    }
}

```

- A. 156Exception in thread main MyExc1
- B. 156
- C. 1562
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

8. Data il seguente codice:

```

1. public interface I {
2.     int x = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
- B. abstract int x = 0;
- C. protected int x = 0;
- D. public int x;
- E. Nessuna delle precedenti

9. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle() {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
 - B. 135
 - C. 1235
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

10. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
 - B. 135
 - C. 1235
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

11. Date le dichiarazioni:

```

Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];

```

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

- A. a = (Boolean []) z;
 - B. t = (Error []) z;
 - C. t = (Error []) a;
 - D. a = (Boolean []) t;
 - E. Nessuno dei precedenti
-

12. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
 - B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
 - C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
 - D. È possibile dichiarare un attributo senza inizializzarlo
 - E. Una classe non può essere dichiarata final
-

13. Qual'è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 u ) {
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc1 g ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
}

```

```

        finally {
            System.out.print(4);
        }
    }
}

```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

14. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}

class B {
    private A a1 = new A(o1");
    private A a2 = new A(o2");
    private A a3 = new A(o3");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
    private void m() {}
    public static void main (String[] args) {
        new B().f();
    }
}

```

- A. Quello con s=o1
- B. Quello con s=o2
- C. Quello con s=o3
- D. Quelli con s uguale a o1 e o2
- E. Nessuna delle precedenti.

15. Quale output si ottiene invocando il metodo q?

```

class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;

```

```

        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
}

void m(String s3, Boolean b1, String s5) {
    String s7 = s4.concat("abc");
    if(s3 == s2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s5 == s7) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 110
- B. 001
- C. 100
- D. 010
- E. 000

16. Qual'è l'output di questo codice?

```

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

static void q() throws Exception {
    try {
        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc1 f ) {
    }
    finally {

```

```
        System.out.print(5);  
    }  
}  
}
```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3

- C. 41
 - 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. 29

1. Qual è l'output di questo programma?

```
abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

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

2. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```
class X {
    static char a;
    static byte b;
    static short c;
    static int d;
    static long e;
    static String s;

    public static void main(String[] args) {
        System.out.println(a+b+c+d+e+s);
    }
}
```

- A. 00000null
- B. 0null

- C. 0
- D. null
- E. Errore di compilazione.

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 u ) {
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc1 g ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. Qual è l'output di questo programma?

```
abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}
```

```

}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

5. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
 - B. 135
 - C. 1235
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

6. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}
class B {
    private A a1 = new A("o1");
    private A a2 = new A("o2");
    private A a3 = new A("o3");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
    private void m() {}
    public static void main (String[] args) {
        new B().f();
    }
}

```

- A. Quello con s=o1
 - B. Quello con s=o2
 - C. Quello con s=o3
 - D. Quelli con s uguale a o1 e o2
 - E. Nessuna delle precedenti.
-

7. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle() {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione

E. Nessuna delle precedenti

8. Date le dichiarazioni:

```
Boolean [] a;  
Error [] t;  
Object [] z;  
z = new Boolean [3];  
t = new Error [2];  
a = new Boolean [8];
```

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

- A. a = (Boolean []) z;
 - B. t = (Error []) z;
 - C. t = (Error []) a;
 - D. a = (Boolean []) t;
 - E. Nessuno dei precedenti
-

9. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
 - B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
 - C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
 - D. È possibile dichiarare un attributo senza inizializzarlo
 - E. Una classe non può essere dichiarata final
-

10. Qual'è l'output di questo codice?

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

```
    }  
}  
static void m() throws Exception {  
    try {  
        System.out.print(5);  
        throw( new MyExc1() );  
    }  
    finally {  
        System.out.print(6);  
    }  
}  
}
```

- A. 156Exception in thread main MyExc1
 - B. 156
 - C. 1562
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

11. Data il seguente codice:

```
1. public interface I {  
2.     int x = 0;  
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
 - B. abstract int x = 0;
 - C. protected int x = 0;
 - D. public int x;
 - E. Nessuna delle precedenti
-

12. Data la seguente classe:

```
1. public class C {  
2.     public void m(int i, String s){}  
3.  
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. public int m(int i, String s)
 - B. public void m(int i, String n)
 - C. public void M(int i, String s)
 - D. private void m(int j, String q)
 - E. Nessuna delle precedenti
-

13. Date le dichiarazioni:

```
Boolean p;  
Object w;  
String z;
```

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

- A. p = (Boolean) z;
- B. p = w;
- C. w = z;
- D. p = z;
- E. z = p;

14. Quale output si ottiene invocando il metodo q?

```
class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {
        String s7 = s4.concat("abc");
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 001
- C. 100
- D. 010
- E. 000

15. Qual è l'output di questo programma?

```
class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
```

```
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

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

16. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 f ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc1 f ) {
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- 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. 30

1. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}

class B {
    private A a1 = new A("o1");
    private A a2 = new A("o2");
    private A a3 = new A("o3");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
    private void m() {}
    public static void main (String[] args) {
        new B().f();
    }
}
```

- A. Quello con s=o1
 B. Quello con s=o2
 C. Quello con s=o3
 D. Quelli con s uguale a o1 e o2
 E. Nessuna delle precedenti.

2. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
 B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
 C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
 D. È possibile dichiarare un attributo senza inizializzarlo

- E. Una classe non può essere dichiarata final

3. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```
class X {
    static char a;
    static byte b;
    static short c;
    static int d;
    static long e;
    static String s;

    public static void main(String[] args) {
        System.out.println(a+b+c+d+e+s);
    }
}
```

- A. 00000null
 B. 0null
 C. 0
 D. null
 E. Errore di compilazione.

4. Qual è l'output di questo programma?

```
class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}

class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

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

```
        System.out.print(2);
        throw( new MyExcl() );
    }
    catch( MyExcl g ) {
        System.out.print(3);
        throw( new MyExcl() );
    }
    finally {
        System.out.print(4);
    }
}
```

5. Date le dichiarazioni:

```
Boolean p;
Object w;
String z;
```

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

- A. p = (Boolean) z;
- B. p = w;
- C. w = z;
- D. p = z;
- E. z = p;

- A. 23Exception in thread main MyExcl
- B. 234Exception in thread main MyExcl
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

6. Data la seguente classe:

```
1. public class C {
2.     public void m(int i, String s){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. public int m(int i, String s)
- B. public void m(int i, String n)
- C. public void M(int i, String s)
- D. private void m(int j, String q)
- E. Nessuna delle precedenti

8. Date le dichiarazioni:

```
Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];
```

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

- A. a = (Boolean []) z;
- B. t = (Error []) z;
- C. t = (Error []) a;
- D. a = (Boolean []) t;
- E. Nessuno dei precedenti

7. Qual'è l'output di questo codice?

```
class MyExcl extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 u ) {
        }
    }
    static void p() {
        try {
```

9. Data il seguente codice:

```
1. public interface I {
2.     int x = 0;
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
- B. abstract int x = 0;
- C. protected int x = 0;
- D. public int x;
- E. Nessuna delle precedenti

10. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

11. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 f ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc1 f ) {
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```

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

```



```

}
static void m() throws Exception {
    try {
        System.out.print(5);
        throw( new MyExcl() );
    }
    finally {
        System.out.print(6);
    }
}
}

```

- A. 156Exception in thread main MyExcl
- B. 156
- C. 1562
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

14. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle() {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

15. Quale output si ottiene invocando il metodo q?

```

class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;

```

```

        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {
        String s7 = s4.concat("abc");
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 001
- C. 100
- D. 010
- E. 000

16. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

- A. M
- B. F
- 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. 31

1. Quale output si ottiene invocando il metodo q?

```
class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {
        String s7 = s4.concat("abc");
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 001
- C. 100
- D. 010
- E. 000

2. Date le dichiarazioni:

```
Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];
```

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

- A. a = (Boolean []) z;
- B. t = (Error []) z;
- C. t = (Error []) a;
- D. a = (Boolean []) t;
- E. Nessuno dei precedenti

3. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```
class X {
    static char a;
    static byte b;
    static short c;
    static int d;
    static long e;
    static String s;

    public static void main(String[] args) {
        System.out.println(a+b+c+d+e+s);
    }
}
```

- A. 00000null
- B. 0null
- C. 0
- D. null
- E. Errore di compilazione.

4. Data il seguente codice:

```
1. public interface I {
2.     int x = 0;
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
- B. abstract int x = 0;
- C. protected int x = 0;

- D. public int x;
 - E. Nessuna delle precedenti
-

5. Data la seguente classe:

```
1. public class C {
2.     public void m(int i, String s){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. public int m(int i, String s)
 - B. public void m(int i, String n)
 - C. public void M(int i, String s)
 - D. private void m(int j, String q)
 - E. Nessuna delle precedenti
-

6. Qual è l'output di questo programma?

```
class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

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

7. Qual'è l'output di questo codice?

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

```
try {
    System.out.print(1);
    m();
}
catch( Exception e ) {
    System.out.print(2);
}
catch( MyExc3 j ) {
    System.out.print(3);
}
catch( MyExc2 w ) {
    System.out.print(4);
}
}
static void m() throws Exception {
    try {
        System.out.print(5);
        throw( new MyExc1() );
    }
    finally {
        System.out.print(6);
    }
}
}
```

- A. 156Exception in thread main MyExc1
 - B. 156
 - C. 1562
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

8. Date le dichiarazioni:

```
Boolean p;
Object w;
String z;
```

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

- A. p = (Boolean) z;
 - B. p = w;
 - C. w = z;
 - D. p = z;
 - E. z = p;
-

9. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
```

```

    Rectangle(int i) {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

10. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- D. È possibile dichiarare un attributo senza inizializzarlo
- E. Una classe non può essere dichiarata final

11. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

- A. M
- B. F

- 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 Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 u ) {
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc1 g ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

13. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle() {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {

```

```

public static void main(String argv[]) {
    System.out.print(1);
    Square s = new Square();
}
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

14. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}

class B {
    private A a1 = new A(o1");
    private A a2 = new A(o2");
    private A a3 = new A(o3");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
    private void m() {}
    public static void main (String[] args) {
        new B().f();
    }
}

```

- A. Quello con s=o1
- B. Quello con s=o2
- C. Quello con s=o3
- D. Quelli con s uguale a o1 e o2
- E. Nessuna delle precedenti.

15. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}

class B extends A {
    public static void main(String argv[]) {

```

```

        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

16. Qual'è l'output di questo codice?

```

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

        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 f ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc1 f ) {
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

Prova n. 32

1. Date le dichiarazioni:

```
Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];
```

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

- A. a = (Boolean []) z;
- B. t = (Error []) z;
- C. t = (Error []) a;
- D. a = (Boolean []) t;
- E. Nessuno dei precedenti

2. Qual è l'output di questo programma?

```
abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

- A. M
- B. F
- 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 B1 {
    public static void main(String [] argv)
        throws Exception {

        try {
            System.out.print(1);
            m();
        }
        catch( Exception e ) {
            System.out.print(2);
        }
        catch( MyExc3 j ) {
            System.out.print(3);
        }
        catch( MyExc2 w ) {
            System.out.print(4);
        }
    }
    static void m() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(6);
        }
    }
}
```

- A. 156Exception in thread main MyExc1
- B. 156
- C. 1562
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. Qual è l'output di questo programma?

```
abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}
```



```

class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

5. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}

class B {
    private A a1 = new A("o1");
    private A a2 = new A("o2");
    private A a3 = new A("o3");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
    private void m() {}
    public static void main (String[] args) {
        new B().f();
    }
}

```

- A. Quello con s=o1
- B. Quello con s=o2
- C. Quello con s=o3
- D. Quelli con s uguale a o1 e o2
- E. Nessuna delle precedenti.

6. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}

class Rectangle extends Polygon{
    Rectangle() {
        super(6);
        System.out.print(5);
    }
}

class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- 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 D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 u ) {
        }
    }

    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc1 g ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1
- C. 23333333... (ciclo infinito)

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

E. 000

8. Data la seguente classe:

```
1. public class C {
2.     public void m(int i, String s){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. `public int m(int i, String s)`
- B. `public void m(int i, String n)`
- C. `public void M(int i, String s)`
- D. `private void m(int j, String q)`
- E. Nessuna delle precedenti

9. Quale output si ottiene invocando il metodo q?

```
class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {
        String s7 = s4.concat("abc");
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 001
- C. 100
- D. 010

10. Qual è l'output di questo programma?

```
class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

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

11. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```
class X {
    static char a;
    static byte b;
    static short c;
    static int d;
    static long e;
    static String s;

    public static void main(String[] args) {
        System.out.println(a+b+c+d+e+s);
    }
}
```

- A. 00000null
- B. 0null
- C. 0
- D. null
- E. Errore di compilazione.

12. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

13. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- D. È possibile dichiarare un attributo senza iniziarlo
- E. Una classe non può essere dichiarata final

14. Qual'è l'output di questo codice?

```

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

```

```

        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc1 f ) {
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

15. Data il seguente codice:

```

1. public interface I {
2.     int x = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
- B. abstract int x = 0;
- C. protected int x = 0;
- D. public int x;
- E. Nessuna delle precedenti

16. Date le dichiarazioni:

```

Boolean p;
Object w;
String z;

```

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

- A. p = (Boolean) z;
- B. p = w;
- C. w = z;
- D. p = z;
- E. z = p;

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. Quale output si ottiene invocando il metodo q?

```
class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {
        String s7 = s4.concat("abc");
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 001
- C. 100
- D. 010
- E. 000

2. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
```

```

    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

3. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}
```

```
class B {
    private A a1 = new A("o1");
    private A a2 = new A("o2");
    private A a3 = new A("o3");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
    private void m() {}
    public static void main (String[] args) {
        new B().f();
    }
}
```

```
}  
}
```

- A. Quello con s=o1
 - B. Quello con s=o2
 - C. Quello con s=o3
 - D. Quelli con s uguale a o1 e o2
 - E. Nessuna delle precedenti.
-

4. Qual è l'output di questo programma?

```
abstract class A {  
    public void f() {}  
    public void m() {  
        System.out.println("M");  
    }  
}  
class B extends A {  
    public static void main(String argv[]) {  
        B b = new B();  
        b.m();  
    }  
    public void f() {  
        System.out.println("F");  
    }  
    public void m() {  
        f();  
    }  
}
```

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

5. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
 - B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
 - C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
 - D. È possibile dichiarare un attributo senza inicializzarlo
 - E. Una classe non può essere dichiarata final
-

6. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```
class X {  
    static char a;  
    static byte b;  
    static short c;  
    static int d;  
    static long e;  
    static String s;  
  
    public static void main(String[] args) {  
        System.out.println(a+b+c+d+e+s);  
    }  
}
```

- A. 00000null
 - B. 0null
 - C. 0
 - D. null
 - E. Errore di compilazione.
-

7. Date le dichiarazioni:

```
Boolean p;  
Object w;  
String z;
```

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

- A. p = (Boolean) z;
 - B. p = w;
 - C. w = z;
 - D. p = z;
 - E. z = p;
-

8. Qual è l'output di questo programma?

```
class A {  
    public abstract void f();  
    public void m() {  
        System.out.println("M");  
    }  
}  
class B extends A {  
    public static void main(String argv[]) {  
        B b = new B();  
        b.m();  
    }  
    public void f() {  
        System.out.println("F");  
    }  
    public void m() {  
        f();  
    }  
}
```

- A. M

- B. F
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

```
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

9. Data la seguente classe:

```
1. public class C {
2.     public void m(int i, String s){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. `public int m(int i, String s)`
- B. `public void m(int i, String n)`
- C. `public void M(int i, String s)`
- D. `private void m(int j, String q)`
- E. Nessuna delle precedenti

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

10. Date le dichiarazioni:

```
Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];
```

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

- A. `a = (Boolean []) z;`
- B. `t = (Error []) z;`
- C. `t = (Error []) a;`
- D. `a = (Boolean []) t;`
- E. Nessuno dei precedenti

11. Qual è l'output di questo programma?

```
abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
```

12. Qual'è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 u ) {
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc1 g ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

13. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception e ) {
            System.out.print(2);
        }
        catch( MyExc3 j ) {
            System.out.print(3);
        }
        catch( MyExc2 w ) {
            System.out.print(4);
        }
    }
    static void m() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(6);
        }
    }
}

```

- A. 156Exception in thread main MyExc1
- B. 156
- C. 1562
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 f ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc1 f ) {
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 45132222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

14. Data il seguente codice:

```

1. public interface I {
2.     int x = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
- B. abstract int x = 0;
- C. protected int x = 0;
- D. public int x;
- E. Nessuna delle precedenti

15. Qual'è l'output di questo codice?

16. Qual è l'output di questo programma?

```

class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle() {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```


}
}

- A. 125
- B. 135

- C. 1235
 - 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. 34

1. Data la seguente classe:

```
1. public class C {
2.     public void m(int i, String s){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. public int m(int i, String s)
- B. public void m(int i, String n)
- C. public void M(int i, String s)
- D. private void m(int j, String q)
- E. Nessuna delle precedenti

2. Qual è l'output di questo programma?

```
class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

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

3. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print (2);
    }
    Polygon(int i) {
        System.out.print (3);
    }
}
class Rectangle extends Polygon{
    Rectangle() {
        super (6);
        System.out.print (5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print (1);
        Square s = new Square();
    }
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. Qual è l'output di questo programma?

```
abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

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

5. Qual è l'output di questo programma?

```
abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

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

6. Date le dichiarazioni:

```
Boolean p;
Object w;
String z;
```

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

- A. p = (Boolean) z;
- B. p = w;
- C. w = z;
- D. p = z;
- E. z = p;

7. Quale output si ottiene invocando il metodo q?

```
class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {
        String s7 = s4.concat("abc");
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 001
- C. 100
- D. 010
- E. 000

8. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- D. È possibile dichiarare un attributo senza iniziarlo
- E. Una classe non può essere dichiarata final

9. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
}
```

```

    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}

```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

10. Qual'è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 u ) {
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc1 g ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione

E. Nessuna delle precedenti

11. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```

class X {
    static char a;
    static byte b;
    static short c;
    static int d;
    static long e;
    static String s;

    public static void main(String[] args) {
        System.out.println(a+b+c+d+e+s);
    }
}

```

- A. 00000null
- B. 0null
- C. 0
- D. null
- E. Errore di compilazione.

12. Date le dichiarazioni:

```

Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];

```

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

- A. a = (Boolean []) z;
- B. t = (Error []) z;
- C. t = (Error []) a;
- D. a = (Boolean []) t;
- E. Nessuno dei precedenti

13. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}

class B {
    private A a1 = new A("1");
}

```

```

private A a2 = new A(o2");
private A a3 = new A(o3");
private void f() {
    a1.setRef(a2);
    a2.setRef(a1);
    a3.setRef(a3);
    a3 = a1; a2 = a1;
    m();
}
private void m() {}
public static void main (String[] args) {
    new B().f();
}
}

```

- A. Quello con s=o1
- B. Quello con s=o2
- C. Quello con s=o3
- D. Quelli con s uguale a o1 e o2
- E. Nessuna delle precedenti.

14. Data il seguente codice:

```

1. public interface I {
2.     int x = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
- B. abstract int x = 0;
- C. protected int x = 0;
- D. public int x;
- E. Nessuna delle precedenti

15. Qual'è l'output di questo codice?

```

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

```

```

try {
    System.out.print(4);
    throw( new MyExc1() );
}
catch( MyExc1 f ) {
}
finally {
    System.out.print(5);
}
}

```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

16. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception e ) {
            System.out.print(2);
        }
        catch( MyExc3 j ) {
            System.out.print(3);
        }
        catch( MyExc2 w ) {
            System.out.print(4);
        }
    }
    static void m() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(6);
        }
    }
}

```

- A. 156Exception in thread main MyExc1
- B. 156
- C. 1562
- 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. 35

1. Data la seguente classe:

```
1. public class C {
2.     public void m(int i, String s){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. public int m(int i, String s)
- B. public void m(int i, String n)
- C. public void M(int i, String s)
- D. private void m(int j, String q)
- E. Nessuna delle precedenti

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception e ) {
            System.out.print(2);
        }
        catch( MyExc3 j ) {
            System.out.print(3);
        }
        catch( MyExc2 w ) {
            System.out.print(4);
        }
    }
    static void m() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(6);
        }
    }
}
```

- A. 156Exception in thread main MyExc1
- B. 156
- C. 1562
- D. Errore a tempo di compilazione
- 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 D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 u ) {
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc1 g ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 23Exception in thread main MyExc1
- B. 234Exception in thread main MyExc1
- C. 23333333... (ciclo infinito)
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. Qual è l'output di questo programma?


```

abstract class A {
    public void f() {}
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

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

5. Qual è l'output di questo programma?

```

abstract class A {
    public void f() {}
    public static void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}

```

- A. M
- B. F
- 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 MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 f ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc1 f ) {
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 4513222222... (ciclo infinito)
- B. 45132Exception in thread main MyExc3
- C. 41
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Qual è l'output di questo programma? Ricordare che l'operatore + è associativo a sinistra.

```

class X {
    static char a;
    static byte b;
    static short c;
    static int d;
    static long e;
    static String s;

    public static void main(String[] args) {
        System.out.println(a+b+c+d+e+s);
    }
}

```

- A. 00000null
- B. 0null

- C. 0
 - D. null
 - E. Errore di compilazione.
-

8. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle() {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}
```

- A. 125
 - B. 135
 - C. 1235
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

9. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
 - B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
 - C. Un attributo static non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
 - D. È possibile dichiarare un attributo senza inicializzarlo
 - E. Una classe non può essere dichiarata final
-

10. Date le dichiarazioni:

```
Boolean [] a;
Error [] t;
Object [] z;
z = new Boolean [3];
t = new Error [2];
a = new Boolean [8];
```

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

- A. a = (Boolean []) z;
 - B. t = (Error []) z;
 - C. t = (Error []) a;
 - D. a = (Boolean []) t;
 - E. Nessuno dei precedenti
-

11. Qual è l'output di questo programma?

```
class Polygon {
    Polygon() {
        System.out.print(2);
    }
    Polygon(int i) {
        System.out.print(3);
    }
}
class Rectangle extends Polygon{
    Rectangle(int i) {
        super(6);
        System.out.print(5);
    }
}
class Square extends Rectangle {
    public static void main(String argv[]) {
        System.out.print(1);
        Square s = new Square();
    }
}
```

- A. 125
 - B. 135
 - C. 1235
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

12. Date le dichiarazioni:

```
Boolean p;
Object w;
String z;
```

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

- A. p = (Boolean) z;
 - B. p = w;
 - C. w = z;
 - D. p = z;
 - E. z = p;
-

13. Data il seguente codice:

```
1. public interface I {
2.     int x = 0;
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int x = 0;
 - B. abstract int x = 0;
 - C. protected int x = 0;
 - D. public int x;
 - E. Nessuna delle precedenti
-

14. Quale output si ottiene invocando il metodo q?

```
class G {
    private String s1 = "abc";
    private String s2 = s1.concat("d");
    private Boolean b2 = new Boolean(true);
    private String s4 = "";
    void q() {
        String s6;
        s6 = s4.concat("abc");
        m(s2, new Boolean(true), s6);
    }
    void m(String s3, Boolean b1, String s5) {
        String s7 = s4.concat("abc");
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
 - B. 001
 - C. 100
 - D. 010
 - E. 000
-

15. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    private A ref;
    private String s;
    public A(String s) {this.s = s;}
    public void setRef(A a) {ref = a;}
}
```

```
class B {
    private A a1 = new A(o1");
    private A a2 = new A(o2");
    private A a3 = new A(o3");
    private void f() {
        a1.setRef(a2);
        a2.setRef(a1);
        a3.setRef(a3);
        a3 = a1; a2 = a1;
        m();
    }
    private void m() {}
    public static void main (String[] args) {
        new B().f();
    }
}
```

- A. Quello con s=o1
 - B. Quello con s=o2
 - C. Quello con s=o3
 - D. Quelli con s uguale a o1 e o2
 - E. Nessuna delle precedenti.
-

16. Qual è l'output di questo programma?

```
class A {
    public abstract void f();
    public void m() {
        System.out.println("M");
    }
}
class B extends A {
    public static void main(String argv[]) {
        B b = new B();
        b.m();
    }
    public void f() {
        System.out.println("F");
    }
    public void m() {
        f();
    }
}
```

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

Prova n. 1

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1

Prova d'esame

prof. Piero A. Bonatti

18 marzo 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

18 marzo 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

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: