

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. Date le dichiarazioni:

```
Boolean [] c;
Object [] [] s;
Object [] w;
```

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

- A. `c = (Boolean []) s;`
- B. `s = w;`
- C. `c = w;`
- D. `s = (Object [] []) c;`
- E. `w = (Object []) s;`

2. Qual è l'output di questo codice?

```
public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

3. Data la classe

```
1. class X {
2.     private Object o;
3.     void f() {
4.         o = new Object();
```

```
5.         this.g(o);
6.     }
7.     void g(Object obj) {
8.         obj = null;
9.         obj = new Object();
10.    }
11.    public static void main(String[] a){
12.        new X().f();
13.    }
14. }
```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 8.
- C. Al termine della linea 9.
- D. Al termine della linea 12.
- E. Non si può dire.

4. Qual'è l'output di questo codice?

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

```

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

```

- A. 423Exception in thread "main" MyExc3
- B. 42222222... (ciclo infinito)
- C. Errore a tempo di compilazione
- D. 42Exception in thread "main" MyExc3
- E. Nessuna delle precedenti

5. Date le dichiarazioni:

```

class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B2 c;
C2 s;
C0 x;
c = new B2 ();
x = new C2 ();
s = new C2 ();

```

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

- A. c = (B2) x;
- B. s = (C2) x;
- C. c = (B2) s;
- D. s = (C2) c;
- E. Nessuno dei precedenti

6. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A(). new B();
    }
}

```

- A. 0
- B. 1
- C. Errore a tempo di compilazione

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

7. Quale output si ottiene invocando il metodo p?

```

class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
          i4, i3);
    }
    void m(String s1, Boolean b2,
           Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 000
- B. 101
- C. 100
- D. 011
- E. 001

8. Qual è l'output di questo codice?

```

class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    public static void main( String [] a) {
        new B();
    }
}

```

- A. 0
- B. 1
- 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 A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    B() {
        super( 2 );
    }
    public static void main( String [] a ) {
        new B();
    }
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

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

class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

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

    public static void main(String[] args){
        A a = new B();
        a.m();
    }
}
```

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

11. Qual'è l'output di questo codice?

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

- A. 111... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. Exception in thread "main" MyExc2
- D. 2Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

12. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
 - B. Ai metodi **static** non si applica il *dynamic method dispatch*
 - C. Una classe non può essere dichiarata **final**
 - D. Un array vuoto può avere riferimento **null**
 - E. L'operatore **new** non restituisce un riferimento all'oggetto appena creato
-

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

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

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

14. Qual è l'output di questo codice?

```
public interface I {  
    void m();  
}  
  
abstract class A implements I {  
    public void f() {  
        System.out.println("A.f");  
    }  
}  
  
class B extends A {  
    public void m() {  
        System.out.println("B.m");  
    }  
    public void f() { m(); }  
  
    public static void main(String[] args){  
        I i = new B();  
        i.m();  
    }  
}
```

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

15. Se nella classe C è dichiarato il metodo

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

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

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

16. 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 h ) {  
            System.out.print(2);  
        }  
    }  
    static void q() throws Exception {  
        try {  
            throw( new MyExc2() );  
        }  
        catch( Exception w ) {  
            System.out.print(3);  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. Errore a tempo di compilazione
 - B. 42
 - C. 341
 - D. 41
 - 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. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Ai metodi `static` non si applica il *dynamic method dispatch*
- C. Una classe non può essere dichiarata `final`
- D. Un array vuoto può avere riferimento `null`
- E. L'operatore `new` non restituisce un riferimento all'oggetto appena creato

```

}
if(b2 == b1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(i1 == i2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
}
    
```

- A. 000
- B. 101
- C. 100
- D. 011
- E. 001

2. Se nella classe C è dichiarato il metodo

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

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

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

4. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A().new B();
    }
}
    
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

3. Quale output si ottiene invocando il metodo p?

```

class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
          i4, i3);
    }
    void m(String s1, Boolean b2,
           Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
    
```

5. Data la classe

```

1. class X {
2.     private Object o;
3.     void f() {
4.         o = new Object();
5.         this.g(o);
6.     }
7.     void g(Object obj) {
8.         obj = null;
9.         obj = new Object();
10.    }
11.    public static void main(String[] a) {
12.        new X().f();
13.    }
14. }

```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 8.
- C. Al termine della linea 9.
- D. Al termine della linea 12.
- E. Non si può dire.

6. Date le dichiarazioni:

```

Boolean [] c;
Object [] [] s;
Object [] w;

```

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

- A. `c = (Boolean []) s;`
- B. `s = w;`
- C. `c = w;`
- D. `s = (Object [] []) c;`
- E. `w = (Object []) s;`

7. Date le dichiarazioni:

```

class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B2 c;
C2 s;
C0 x;
c = new B2();
x = new C2();
s = new C2();

```

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

- A. `c = (B2) x;`
- B. `s = (C2) x;`
- C. `c = (B2) s;`
- D. `s = (C2) c;`
- E. Nessuno dei precedenti

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

```

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

```

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

9. Qual è l'output di questo codice?

```

class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    B() {
        super( 2 );
    }
    public static void main( String [] a) {
        new B();
    }
}

```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?


```

public interface I {
    void m();
}

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

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

    public static void main(String[] args){
        I i = new B();
        i.m();
    }
}

```

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

11. Qual è l'output di questo codice?

```

class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    public static void main( String [] a) {
        new B();
    }
}

```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

12. Qual'è l'output di questo codice?

```

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

```

- A. 423Exception in thread "main" MyExc3
- B. 4222222... (ciclo infinito)
- C. Errore a tempo di compilazione
- D. 42Exception in thread "main" MyExc3
- E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```

public interface I {
    void m();
}

class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

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

    public static void main(String[] args){
        A a = new B();
        a.m();
    }
}

```

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

14. Qual'è l'output di questo codice?

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

- A. Errore a tempo di compilazione
- B. 42
- C. 341
- D. 41
- E. Nessuna delle precedenti

15. Qual'è l'output di questo codice?

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

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

- A. 111... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. Exception in thread "main" MyExc2
- D. 2Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```
public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}
```

- A. 0
- B. 1
- 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. 3

1. Quale output si ottiene invocando il metodo p?

```
class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
            i4, i3);
    }
    void m(String s1, Boolean b2,
        Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 000
- B. 101
- C. 100
- D. 011
- E. 001

2. Qual è l'output di questo codice?

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

class A implements I {
    public void f() {
```

```
        System.out.println("A.f");
    }
}

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

    public static void main(String[] args){
        A a = new B();
        a.m();
    }
}
```

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

3. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Ai metodi static non si applica il *dynamic method dispatch*
- C. Una classe non può essere dichiarata final
- D. Un array vuoto può avere riferimento null
- E. L'operatore new non restituisce un riferimento all'oggetto appena creato

4. Qual'è l'output di questo codice?

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

```

static void q() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc2 d ) {
        System.out.print(1);
        throw( new MyExc3() );
    }
    catch( Exception h ) {
        throw( new MyExc2() );
    }
    catch( MyExc3 w ) {
    }
    finally {
        System.out.print(2);
    }
}
}

```

- A. 111... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. Exception in thread "main" MyExc2
- D. 2Exception in thread "main" MyExc2
- 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 h ) {
            System.out.print(2);
        }
    }
    static void q() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( Exception w ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 42
- C. 341

D. 41

E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```

class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    public static void main( String [] a ) {
        new B();
    }
}

```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

7. Date le dichiarazioni:

```

Boolean [] c;
Object [] [] s;
Object [] w;

```

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

- A. c = (Boolean []) s;
- B. s = w;
- C. c = w;
- D. s = (Object [] []) c;
- E. w = (Object []) s;

8. Se nella classe C è dichiarato il metodo

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

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

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

9. Qual è l'output di questo codice?

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

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

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

    public static void main(String[] args) {
        I i = new B();
        i.m();
    }
}
```

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

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

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

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

11. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
    }
}
```

```
    }
    catch( MyExc3 r ) {
        System.out.print(2);
        throw( new MyExc3() );
    }
    finally {
        System.out.print(3);
    }
}
static void p() {
    try {
        System.out.print(4);
        throw( new Exception() );
    }
    catch( MyExc3 a ) {
    }
    finally {
        throw( new MyExc3() );
    }
}
}
```

- A. 423Exception in thread "main" MyExc3
- B. 4222222... (ciclo infinito)
- C. Errore a tempo di compilazione
- D. 42Exception in thread "main" MyExc3
- E. Nessuna delle precedenti

12. Data la classe

```
1. class X {
2.     private Object o;
3.     void f() {
4.         o = new Object();
5.         this.g(o);
6.     }
7.     void g(Object obj) {
8.         obj = null;
9.         obj = new Object();
10.    }
11.    public static void main(String[] a) {
12.        new X().f();
13.    }
14. }
```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 8.
- C. Al termine della linea 9.
- D. Al termine della linea 12.
- E. Non si può dire.

13. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}

```

- A. 0
 - B. 1
 - 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 A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    B() {
        super( 2 );
    }
    public static void main( String [] a) {
        new B();
    }
}

```

- A. 0
 - B. 1
 - C. Errore a tempo di compilazione
 - D. Errore a tempo di esecuzione
 - E. Nessuna delle precedenti
-

15. Date le dichiarazioni:

```

class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B2 c;
C2 s;
C0 x;
c = new B2 ();
x = new C2 ();
s = new C2 ();

```

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

- A. c = (B2) x;
 - B. s = (C2) x;
 - C. c = (B2) s;
 - D. s = (C2) c;
 - E. Nessuno dei precedenti
-

16. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A(). new B();
    }
}

```

- A. 0
 - B. 1
 - 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. Data la classe

```

1. class X {
2.     private Object o;
3.     void f() {
4.         o = new Object();
5.         this.g(o);
6.     }
7.     void g(Object obj) {
8.         obj = null;
9.         obj = new Object();
10.    }
11.    public static void main(String[] a){
12.        new X().f();
13.    }
14. }
```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 8.
- C. Al termine della linea 9.
- D. Al termine della linea 12.
- E. Non si può dire.

2. Se nella classe C è dichiarato il metodo

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

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

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

3. Qual'è 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 h ) {
            System.out.print(2);
        }
    }
    static void q() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( Exception w ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 42
- C. 341
- D. 41
- E. Nessuna delle precedenti

4. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Ai metodi static non si applica il *dynamic method dispatch*
- C. Una classe non può essere dichiarata final
- D. Un array vuoto può avere riferimento null
- E. L'operatore new non restituisce un riferimento all'oggetto appena creato

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

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

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

6. Qual è l'output di questo codice?

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

class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

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

    public static void main(String[] args){
        A a = new B();
        a.m();
    }
}
```

- A. A.f
- B. B.m
- 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 A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}
```

```
}

public class B extends A {
    B() {
        super( 2 );
    }
    public static void main( String [] a) {
        new B();
    }
}
```

- A. 0
- B. 1
- 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 A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    public static void main( String [] a) {
        new B();
    }
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

9. Quale output si ottiene invocando il metodo p?

```
class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
            i4, i3);
    }
    void m(String s1, Boolean b2,
        Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```

    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(i1 == i2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 000
- B. 101
- C. 100
- D. 011
- E. 001

10. Qual è l'output di questo codice?

```

public interface I {
    void m();
}

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

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

    public static void main(String[] args){
        I i = new B();
        i.m();
    }
}

```

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

11. Qual'è l'output di questo codice?

```

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

```

- A. 423Exception in thread "main" MyExc3
- B. 4222222... (ciclo infinito)
- C. Errore a tempo di compilazione
- D. 42Exception in thread "main" MyExc3
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}

```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

E. Nessuna delle precedenti

13. Date le dichiarazioni:

```
Boolean [] c;  
Object [] [] s;  
Object [] w;
```

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

- A. `c = (Boolean []) s;`
- B. `s = w;`
- C. `c = w;`
- D. `s = (Object [] []) c;`
- E. `w = (Object []) s;`

14. Qual'è l'output di questo codice?

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

- A. 111... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. Exception in thread "main" MyExc2
- D. 2Exception in thread "main" MyExc2

15. Date le dichiarazioni:

```
class C0 extends Object {...}  
class C2 extends C0 {...}  
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B2 c;  
C2 s;  
C0 x;  
c = new B2();  
x = new C2();  
s = new C2();
```

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

- A. `c = (B2) x;`
- B. `s = (C2) x;`
- C. `c = (B2) s;`
- D. `s = (C2) c;`
- E. Nessuno dei precedenti

16. Qual è l'output di questo codice?

```
public class A {  
    class B {  
        B() {  
            System.out.println("B");  
        }  
    }  
    A() {  
        System.out.println("A");  
    }  
    public static void main(String [] a) {  
        new A().new B();  
    }  
}
```

- A. 0
 - B. 1
 - 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. Data la classe

```

1. class X {
2.     private Object o;
3.     void f() {
4.         o = new Object();
5.         this.g(o);
6.     }
7.     void g(Object obj) {
8.         obj = null;
9.         obj = new Object();
10.    }
11.    public static void main(String[] a) {
12.        new X().f();
13.    }
14. }
```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 8.
- C. Al termine della linea 9.
- D. Al termine della linea 12.
- E. Non si può dire.

2. Qual è l'output di questo codice?

```

public interface I {
    void m();
}

class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

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

    public static void main(String[] args) {
        A a = new B();
    }
}
```

```

        a.m();
    }
}
```

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

3. Date le dichiarazioni:

```

class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```

B2 c;
C2 s;
C0 x;
c = new B2();
x = new C2();
s = new C2();
```

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

- A. c = (B2) x;
- B. s = (C2) x;
- C. c = (B2) s;
- D. s = (C2) c;
- E. Nessuno dei precedenti

4. Qual è l'output di questo codice?

```

class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}
```

```
public class B extends A {
    public static void main( String [] a) {
        new B();
    }
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

5. Qual'è l'output di questo codice?

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

- A. 423Exception in thread "main" MyExc3
- B. 4222222... (ciclo infinito)
- C. Errore a tempo di compilazione
- D. 42Exception in thread "main" MyExc3
- E. Nessuna delle precedenti

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

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

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

7. Qual'è 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 h ) {
            System.out.print(2);
        }
    }
    static void q() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( Exception w ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 42
- C. 341
- D. 41
- E. Nessuna delle precedenti

8. Date le dichiarazioni:

```
Boolean [] c;
Object [] [] s;
Object [] w;
```

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

- A. c = (Boolean []) s;
- B. s = w;
- C. c = w;

- D. `s = (Object [] []) c;`
- E. `w = (Object []) s;`

9. Qual'è l'output di questo codice?

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

- A. 111... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. Exception in thread "main" MyExc2
- D. 2Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```
public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A(). new B();
    }
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```
class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    B() {
        super( 2 );
    }
    public static void main( String [] a) {
        new B();
    }
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

12. Quale output si ottiene invocando il metodo p?

```
class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
            i4, i3);
    }
    void m(String s1, Boolean b2,
        Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i1 == i2) {
            System.out.print(1);
        }
    }
}
```



```

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

```

- A. 000
- B. 101
- C. 100
- D. 011
- E. 001

13. Se nella classe C è dichiarato il metodo

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

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

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

14. Qual è l'output di questo codice?

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

```

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

```

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

    public static void main(String[] args){
        I i = new B();
    }
}

```

```

        i.m();
    }
}

```

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

15. Qual è l'output di questo codice?

```
public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}

```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

16. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Ai metodi static non si applica il *dynamic method dispatch*
- C. Una classe non può essere dichiarata final
- D. Un array vuoto può avere riferimento null
- E. L'operatore new non restituisce un riferimento all'oggetto appena creato

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

Prova n. 6

1. Qual'è l'output di questo codice?

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

- A. Errore a tempo di compilazione
- B. 42
- C. 341
- D. 41
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}
```

```
public class B extends A {
    public static void main( String [] a) {
        new B();
    }
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

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

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

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

4. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Ai metodi *static* non si applica il *dynamic method dispatch*
- C. Una classe non può essere dichiarata *final*
- D. Un array vuoto può avere riferimento *null*
- E. L'operatore *new* non restituisce un riferimento all'oggetto appena creato

5. Qual'è l'output di questo codice?

```

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

```

- A. 423Exception in thread "main" MyExc3
- B. 42222222... (ciclo infinito)
- C. Errore a tempo di compilazione
- D. 42Exception in thread "main" MyExc3
- E. Nessuna delle precedenti

6. Date le dichiarazioni:

```

class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B2 c;
C2 s;
C0 x;
c = new B2 ();
x = new C2 ();
s = new C2 ();

```

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

- A. c = (B2) x;
- B. s = (C2) x;
- C. c = (B2) s;

- D. s = (C2) c;
- E. Nessuno dei precedenti

7. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A().new B();
    }
}

```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

8. Data la classe

```

1. class X {
2.     private Object o;
3.     void f() {
4.         o = new Object();
5.         this.g(o);
6.     }
7.     void g(Object obj) {
8.         obj = null;
9.         obj = new Object();
10.    }
11.    public static void main(String[] a){
12.        new X().f();
13.    }
14. }

```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 8.
- C. Al termine della linea 9.
- D. Al termine della linea 12.
- E. Non si può dire.

9. Date le dichiarazioni:

```

Boolean [] c;
Object [] [] s;
Object [] w;

```

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

- A. `c = (Boolean []) s;`
 - B. `s = w;`
 - C. `c = w;`
 - D. `s = (Object [] []) c;`
 - E. `w = (Object []) s;`
-

10. Qual è l'output di questo codice?

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

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

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

    public static void main(String[] args) {
        I i = new B();
        i.m();
    }
}
```

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

11. Qual è l'output di questo codice?

```
class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    B() {
        super( 2 );
    }
}
```

```
public static void main( String [] a) {
    new B();
}
}
```

- A. 0
 - B. 1
 - C. Errore a tempo di compilazione
 - D. Errore a tempo di esecuzione
 - E. Nessuna delle precedenti
-

12. Se nella classe C è dichiarato il metodo

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

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

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

13. Qual'è l'output di questo codice?

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

- A. 111... (ciclo infinito)
- B. Errore a tempo di compilazione

- C. Exception in thread "main" MyExc2
 - D. 2Exception in thread "main" MyExc2
 - E. Nessuna delle precedenti
-

14. Qual è l'output di questo codice?

```
public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}
```

- A. 0
 - B. 1
 - C. Errore a tempo di compilazione
 - D. Errore a tempo di esecuzione
 - E. Nessuna delle precedenti
-

15. Quale output si ottiene invocando il metodo p?

```
class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
            i4, i3);
    }
    void m(String s1, Boolean b2,
        Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        }
    }
}
```

```
} else {
    System.out.print(0);
}
if(i1 == i2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
```

- A. 000
 - B. 101
 - C. 100
 - D. 011
 - E. 001
-

16. Qual è l'output di questo codice?

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

class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

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

    public static void main(String[] args){
        A a = new B();
        a.m();
    }
}
```

- A. A.f
 - B. B.m
 - 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. Quale output si ottiene invocando il metodo p?

```
class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
            i4, i3);
    }
    void m(String s1, Boolean b2,
        Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 000
- B. 101
- C. 100
- D. 011
- E. 001

2. Qual è l'output di questo codice?

```
public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
}
```

```
A() {
    System.out.println("A");
}
public static void main(String [] a) {
    new B();
}
}
```

- A. 0
- B. 1
- 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 D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 r ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 a ) {
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. 423Exception in thread "main" MyExc3
- B. 42222222... (ciclo infinito)

- C. Errore a tempo di compilazione
 - D. 42Exception in thread "main" MyExc3
 - E. Nessuna delle precedenti
-

4. Qual'è l'output di questo codice?

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

- A. 111... (ciclo infinito)
 - B. Errore a tempo di compilazione
 - C. Exception in thread "main" MyExc2
 - D. 2Exception in thread "main" MyExc2
 - 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 h ) {
            System.out.print(2);
        }
    }
}
```

```
}
static void q() throws Exception {
    try {
        throw( new MyExc2() );
    }
    catch( Exception w ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
    }
}
}
```

- A. Errore a tempo di compilazione
 - B. 42
 - C. 341
 - D. 41
 - E. Nessuna delle precedenti
-

6. Qual è l'output di questo codice?

```
class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}
public class B extends A {
    public static void main( String [] a ) {
        new B();
    }
}
```

- A. 0
 - B. 1
 - C. Errore a tempo di compilazione
 - D. Errore a tempo di esecuzione
 - E. Nessuna delle precedenti
-

7. Qual è l'output di questo codice?

```
public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A(). new B();
    }
}
```

- A. 0
 - B. 1
 - C. Errore a tempo di compilazione
 - D. Errore a tempo di esecuzione
 - E. Nessuna delle precedenti
-

8. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
 - B. Ai metodi `static` non si applica il *dynamic method dispatch*
 - C. Una classe non può essere dichiarata `final`
 - D. Un array vuoto può avere riferimento `null`
 - E. L'operatore `new` non restituisce un riferimento all'oggetto appena creato
-

9. Date le dichiarazioni:

```
Boolean [] c;  
Object [] [] s;  
Object [] w;
```

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

- A. `c = (Boolean []) s;`
 - B. `s = w;`
 - C. `c = w;`
 - D. `s = (Object [] []) c;`
 - E. `w = (Object []) s;`
-

10. Qual è l'output di questo codice?

```
public interface I {  
    void m();  
}  
  
class A implements I {  
    public void f() {  
        System.out.println("A.f");  
    }  
}  
  
class B extends A {  
    public void m() {  
        System.out.println("B.m");  
    }  
    public void f() { m(); }  
  
    public static void main(String[] args) {  
        A a = new B();  
        a.m();  
    }  
}
```

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

11. Data la classe

```
1. class X {  
2.     private Object o;  
3.     void f() {  
4.         o = new Object();  
5.         this.g(o);  
6.     }  
7.     void g(Object obj) {  
8.         obj = null;  
9.         obj = new Object();  
10.    }  
11.    public static void main(String[] a) {  
12.        new X().f();  
13.    }  
14. }
```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
 - B. Al termine della linea 8.
 - C. Al termine della linea 9.
 - D. Al termine della linea 12.
 - E. Non si può dire.
-

12. Date le dichiarazioni:

```
class C0 extends Object {...}  
class C2 extends C0 {...}  
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B2 c;  
C2 s;  
C0 x;  
c = new B2();  
x = new C2();  
s = new C2();
```

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

- A. `c = (B2) x;`
- B. `s = (C2) x;`
- C. `c = (B2) s;`

- D. `s = (C2) c;`
 - E. Nessuno dei precedenti
-

13. Se nella classe C è dichiarato il metodo

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

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

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

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

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

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

15. Qual è l'output di questo codice?

```
class A {  
    A() {  
        System.out.println(0);  
    }  
    A( int i ) {  
        System.out.println(i);  
    }  
}
```

```
public class B extends A {
```

```
B() {  
    super( 2 );  
}  
public static void main( String [] a) {  
    new B();  
}  
}
```

- A. 0
 - B. 1
 - C. Errore a tempo di compilazione
 - D. Errore a tempo di esecuzione
 - E. Nessuna delle precedenti
-

16. Qual è l'output di questo codice?

```
public interface I {  
    void m();  
}  
  
abstract class A implements I {  
    public void f() {  
        System.out.println("A.f");  
    }  
}  
  
class B extends A {  
    public void m() {  
        System.out.println("B.m");  
    }  
    public void f() { m(); }  
  
    public static void main(String[] args){  
        I i = new B();  
        i.m();  
    }  
}
```

- A. A.f
 - B. B.m
 - 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?

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

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

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

    public static void main(String[] args){
        I i = new B();
        i.m();
    }
}
```

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

2. Date le dichiarazioni:

```
class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B2 c;
C2 s;
C0 x;
c = new B2();
x = new C2();
s = new C2();
```

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

- A. c = (B2) x;
- B. s = (C2) x;
- C. c = (B2) s;
- D. s = (C2) c;
- E. Nessuno dei precedenti

3. Qual è l'output di questo codice?

```
public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A().new B();
    }
}
```

- A. 0
- B. 1
- 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 A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    B() {
```

```

    super( 2 );
}
public static void main( String [] a) {
    new B();
}
}

```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

5. Se nella classe C è dichiarato il metodo

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

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

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

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

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

```

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

7. Quale output si ottiene invocando il metodo p?

```
class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
            i4, i3);
    }
    void m(String s1, Boolean b2,

```

```

    Integer i1, Integer i2) {
    Boolean b1 = new Boolean(true);
    if(s1 == s2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(i1 == i2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 000
- B. 101
- C. 100
- D. 011
- E. 001

8. Qual è l'output di questo codice?

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

class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

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

public static void main(String[] args){
    A a = new B();
    a.m();
}
}

```

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

9. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}

```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

10. Date le dichiarazioni:

```

Boolean [] c;
Object [] [] s;
Object [] w;

```

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

- A. `c = (Boolean []) s;`
- B. `s = w;`
- C. `c = w;`
- D. `s = (Object [] []) c;`
- E. `w = (Object []) s;`

11. Qual'è l'output di questo codice?

```

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

```

```

catch( Exception h ) {
    throw( new MyExc2() );
}
catch( MyExc3 w ) {
}
finally {
    System.out.print(2);
}
}
}

```

- A. 111... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. Exception in thread "main" MyExc2
- D. 2Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

12. Data la classe

```

1. class X {
2.     private Object o;
3.     void f() {
4.         o = new Object();
5.         this.g(o);
6.     }
7.     void g(Object obj) {
8.         obj = null;
9.         obj = new Object();
10.    }
11.    public static void main(String[] a){
12.        new X().f();
13.    }
14. }

```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 8.
- C. Al termine della linea 9.
- D. Al termine della linea 12.
- E. Non si può dire.

13. Qual'è l'output di questo codice?

```

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

```

```

        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 a ) {
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 423Exception in thread "main" MyExc3
- B. 42222222... (ciclo infinito)
- C. Errore a tempo di compilazione
- D. 42Exception in thread "main" MyExc3
- E. Nessuna delle precedenti

14. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Ai metodi **static** non si applica il *dynamic method dispatch*
- C. Una classe non può essere dichiarata **final**
- D. Un array vuoto può avere riferimento **null**
- E. L'operatore **new** non restituisce un riferimento all'oggetto appena creato

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 h ) {
        System.out.print(2);
    }
}
static void q() throws Exception {
    try {
        throw( new MyExc2() );
    }
    catch( Exception w ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
    }
}
}
}

```

- A. Errore a tempo di compilazione
- B. 42
- C. 341
- D. 41
- E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```

class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    public static void main( String [] a ) {
        new B();
    }
}

```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

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

Prova n. 9

1. Qual'è l'output di questo codice?

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

- A. 111... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. Exception in thread "main" MyExc2
- D. 2Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}
```

}

```
public class B extends A {
    public static void main( String [] a) {
        new B();
    }
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

3. Data la classe

```
1. class X {
2.     private Object o;
3.     void f() {
4.         o = new Object();
5.         this.g(o);
6.     }
7.     void g(Object obj) {
8.         obj = null;
9.         obj = new Object();
10.    }
11.    public static void main(String[] a){
12.        new X().f();
13.    }
14. }
```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 8.
- C. Al termine della linea 9.
- D. Al termine della linea 12.
- E. Non si può dire.

4. Quale output si ottiene invocando il metodo p?

```

class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
            i4, i3);
    }
    void m(String s1, Boolean b2,
        Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 000
- B. 101
- C. 100
- D. 011
- E. 001

5. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A(). new B();
    }
}

```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

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

```

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

```

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

7. Qual è l'output di questo codice?

```

class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

```

```

public class B extends A {
    B() {
        super( 2 );
    }
    public static void main( String [] a) {
        new B();
    }
}

```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}

```

- A. 0
 - B. 1
 - C. Errore a tempo di compilazione
 - D. Errore a tempo di esecuzione
 - E. Nessuna delle precedenti
-

9. Date le dichiarazioni:

```
Boolean [] c;  
Object [] [] s;  
Object [] w;
```

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

- A. `c = (Boolean []) s;`
 - B. `s = w;`
 - C. `c = w;`
 - D. `s = (Object [] []) c;`
 - E. `w = (Object []) s;`
-

10. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
 - B. Ai metodi `static` non si applica il *dynamic method dispatch*
 - C. Una classe non può essere dichiarata `final`
 - D. Un array vuoto può avere riferimento `null`
 - E. L'operatore `new` non restituisce un riferimento all'oggetto appena creato
-

11. Date le dichiarazioni:

```
class C0 extends Object {...}  
class C2 extends C0 {...}  
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B2 c;  
C2 s;  
C0 x;  
c = new B2 ();  
x = new C2 ();  
s = new C2 ();
```

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

- A. `c = (B2) x;`
- B. `s = (C2) x;`

- C. `c = (B2) s;`
 - D. `s = (C2) c;`
 - E. Nessuno dei precedenti
-

12. Qual è l'output di questo codice?

```
public interface I {  
    void m();  
}  
  
abstract class A implements I {  
    public void f() {  
        System.out.println("A.f");  
    }  
}  
  
class B extends A {  
    public void m() {  
        System.out.println("B.m");  
    }  
    public void f() { m(); }  
  
    public static void main(String[] args){  
        I i = new B();  
        i.m();  
    }  
}
```

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

13. Qual è l'output di questo codice?

```
public interface I {  
    void m();  
}  
  
class A implements I {  
    public void f() {  
        System.out.println("A.f");  
    }  
}  
  
class B extends A {  
    public void m() {  
        System.out.println("B.m");  
    }  
    public void f() { m(); }  
  
    public static void main(String[] args){  
        A a = new B();  
        a.m();  
    }  
}
```

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

14. Qual'è l'output di questo codice?

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

- A. Errore a tempo di compilazione
- B. 42
- C. 341
- D. 41
- E. Nessuna delle precedenti

15. Qual'è l'output di questo codice?

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

```
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 r ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 a ) {
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. 423Exception in thread "main" MyExc3
- B. 4222222... (ciclo infinito)
- C. Errore a tempo di compilazione
- D. 42Exception in thread "main" MyExc3
- E. Nessuna delle precedenti

16. Se nella classe C è dichiarato il metodo

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

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

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

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

Prova n. 10

1. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Ai metodi `static` non si applica il *dynamic method dispatch*
- C. Una classe non può essere dichiarata `final`
- D. Un array vuoto può avere riferimento `null`
- E. L'operatore `new` non restituisce un riferimento all'oggetto appena creato

```
B2 c;
C2 s;
C0 x;
c = new B2 ();
x = new C2 ();
s = new C2 ();
```

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

- A. `c = (B2) x;`
- B. `s = (C2) x;`
- C. `c = (B2) s;`
- D. `s = (C2) c;`
- E. Nessuno dei precedenti

2. Qual è l'output di questo codice?

```
class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    public static void main( String [] a ) {
        new B();
    }
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

3. Date le dichiarazioni:

```
class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

4. Se nella classe C è dichiarato il metodo

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

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

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

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

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

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

6. Qual è l'output di questo codice?

```

public interface I {
    void m();
}

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

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

    public static void main(String[] args) {
        I i = new B();
        i.m();
    }
}

```

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

7. Date le dichiarazioni:

```

Boolean [] c;
Object [] [] s;
Object [] w;

```

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

- A. `c = (Boolean []) s;`
- B. `s = w;`
- C. `c = w;`
- D. `s = (Object [] []) c;`
- E. `w = (Object []) s;`

8. Data la classe

```

1. class X {
2.     private Object o;
3.     void f() {
4.         o = new Object();
5.         this.g(o);
6.     }
7.     void g(Object obj) {
8.         obj = null;
9.         obj = new Object();

```

```

10.     }
11.     public static void main(String[] a) {
12.         new X().f();
13.     }
14. }

```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 8.
- C. Al termine della linea 9.
- D. Al termine della linea 12.
- E. Non si può dire.

9. Qual è l'output di questo codice?

```

public interface I {
    void m();
}

class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

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

    public static void main(String[] args) {
        A a = new B();
        a.m();
    }
}

```

- A. A.f
- B. B.m
- 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 A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    B() {

```



```

    super( 2 );
}
public static void main( String [] a) {
    new B();
}
}

```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

11. Qual'è l'output di questo codice?

```

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

```

- A. 111... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. Exception in thread "main" MyExc2
- D. 2Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {

```

```

        System.out.println("B");
    }
}
A() {
    System.out.println("A");
}
public static void main(String [] a) {
    new A(). new B();
}
}

```

- A. 0
- B. 1
- 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 h ) {
            System.out.print(2);
        }
    }
    static void q() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( Exception w ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 42
- C. 341
- D. 41
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}

```

- A. 000
- B. 101
- C. 100
- D. 011
- E. 001

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

15. Quale output si ottiene invocando il metodo p?

```

class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
          i4, i3);
    }
    void m(String s1, Boolean b2,
           Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

16. Qual'è l'output di questo codice?

```

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

```

- A. 423Exception in thread "main" MyExc3
- B. 4222222... (ciclo infinito)
- C. Errore a tempo di compilazione
- D. 42Exception in thread "main" MyExc3
- E. Nessuna delle precedenti

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

Prova n. 11

1. Qual'è l'output di questo codice?

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

- A. 111... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. Exception in thread "main" MyExc2
- D. 2Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

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

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

- A. public A(String b) {}
- B. void A() {}

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

3. Se nella classe C è dichiarato il metodo

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

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

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

4. Data la classe

```
1. class X {
2.     private Object o;
3.     void f() {
4.         o = new Object();
5.         this.g(o);
6.     }
7.     void g(Object obj) {
8.         obj = null;
9.         obj = new Object();
10.    }
11.    public static void main(String[] a){
12.        new X().f();
13.    }
14. }
```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 8.
- C. Al termine della linea 9.
- D. Al termine della linea 12.
- E. Non si può dire.

5. Date le dichiarazioni:

```
Boolean [] c;  
Object [] [] s;  
Object [] w;
```

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

- A. `c = (Boolean []) s;`
- B. `s = w;`
- C. `c = w;`
- D. `s = (Object [] []) c;`
- E. `w = (Object []) s;`

6. Qual è l'output di questo codice?

```
class A {  
    A() {  
        System.out.println(0);  
    }  
    A( int i ) {  
        System.out.println(i);  
    }  
}  
  
public class B extends A {  
    B() {  
        super( 2 );  
    }  
    public static void main( String [] a ) {  
        new B();  
    }  
}
```

- A. 0
- B. 1
- 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 A {  
    A() {  
        System.out.println(0);  
    }  
    A( int i ) {  
        System.out.println(i);  
    }  
}  
  
public class B extends A {  
    public static void main( String [] a ) {  
        new B();  
    }  
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```
public interface I {  
    void m();  
}  
  
abstract class A implements I {  
    public void f() {  
        System.out.println("A.f");  
    }  
}  
  
class B extends A {  
    public void m() {  
        System.out.println("B.m");  
    }  
    public void f() { m(); }  
  
    public static void main(String[] args) {  
        I i = new B();  
        i.m();  
    }  
}
```

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

9. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Ai metodi `static` non si applica il *dynamic method dispatch*
- C. Una classe non può essere dichiarata `final`
- D. Un array vuoto può avere riferimento `null`
- E. L'operatore `new` non restituisce un riferimento all'oggetto appena creato

10. Date le dichiarazioni:

```

class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B2 c;
C2 s;
C0 x;
c = new B2 ();
x = new C2 ();
s = new C2 ();

```

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

- A. c = (B2) x;
- B. s = (C2) x;
- C. c = (B2) s;
- D. s = (C2) c;
- E. Nessuno dei precedenti

11. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}

```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```

public interface I {
    void m();
}

class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

```

```

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

    public static void main(String[] args){
        A a = new B();
        a.m();
    }
}

```

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

13. Quale output si ottiene invocando il metodo p?

```

class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
            i4, i3);
    }
    void m(String s1, Boolean b2,
        Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 000
- B. 101
- C. 100
- D. 011
- E. 001

14. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A(). new B();
    }
}

```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

15. Qual'è l'output di questo codice?

```

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

```

```

    }
}

```

- A. 423Exception in thread "main" MyExc3
- B. 42222222... (ciclo infinito)
- C. Errore a tempo di compilazione
- D. 42Exception in thread "main" MyExc3
- 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 h ) {
            System.out.print(2);
        }
    }
    static void q() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( Exception w ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 42
- C. 341
- D. 41
- 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. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Ai metodi `static` non si applica il *dynamic method dispatch*
- C. Una classe non può essere dichiarata `final`
- D. Un array vuoto può avere riferimento `null`
- E. L'operatore `new` non restituisce un riferimento all'oggetto appena creato

- A. `c = (B2) x;`
- B. `s = (C2) x;`
- C. `c = (B2) s;`
- D. `s = (C2) c;`
- E. Nessuno dei precedenti

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

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

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

4. Qual è l'output di questo codice?

```
public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A().new B();
    }
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

3. Date le dichiarazioni:

```
class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B2 c;
C2 s;
C0 x;
c = new B2();
x = new C2();
s = new C2();
```

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

5. Quale output si ottiene invocando il metodo `p`?

```
class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
          i4, i3);
    }
    void m(String s1, Boolean b2,
           Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```

    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(i1 == i2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 000
- B. 101
- C. 100
- D. 011
- E. 001

6. Qual è l'output di questo codice?

```

class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    public static void main( String [] a) {
        new B();
    }
}

```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

7. Date le dichiarazioni:

```

Boolean [] c;
Object [] [] s;
Object [] w;

```

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

- A. c = (Boolean []) s;
- B. s = w;
- C. c = w;

- D. s = (Object [] []) c;
- E. w = (Object []) s;

8. Qual è l'output di questo codice?

```

public interface I {
    void m();
}

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

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

    public static void main(String[] args){
        I i = new B();
        i.m();
    }
}

```

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

9. Qual è l'output di questo codice?

```

public interface I {
    void m();
}

class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

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

    public static void main(String[] args){
        A a = new B();
        a.m();
    }
}

```

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

10. Se nella classe C è dichiarato il metodo

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

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

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

11. Qual è l'output di questo codice?

```
public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}
```

- A. 0
 - B. 1
 - C. Errore a tempo di compilazione
 - D. Errore a tempo di esecuzione
 - E. Nessuna delle precedenti
-

12. Data la classe

```
1. class X {
2.     private Object o;
3.     void f() {
4.         o = new Object();
5.         this.g(o);
6.     }
7.     void g(Object obj) {
8.         obj = null;
9.         obj = new Object();
```

```
10.     }
11.     public static void main(String[] a) {
12.         new X().f();
13.     }
14. }
```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
 - B. Al termine della linea 8.
 - C. Al termine della linea 9.
 - D. Al termine della linea 12.
 - E. Non si può dire.
-

13. Qual'è l'output di questo codice?

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

- A. 111... (ciclo infinito)
 - B. Errore a tempo di compilazione
 - C. Exception in thread "main" MyExc2
 - D. 2Exception in thread "main" MyExc2
 - E. Nessuna delle precedenti
-

14. Qual'è l'output di questo codice?

```

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

```

- A. Errore a tempo di compilazione
- B. 42
- C. 341
- D. 41
- E. Nessuna delle precedenti

15. Qual è l'output di questo codice?

```

class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    B() {
        super( 2 );
    }
    public static void main( String [] a ) {
        new B();
    }
}

```

```

}
}

```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

16. Qual'è l'output di questo codice?

```

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

```

- A. 423Exception in thread "main" MyExc3
- B. 42222222... (ciclo infinito)
- C. Errore a tempo di compilazione
- D. 42Exception in thread "main" MyExc3
- 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. Data la classe

```

1. class X {
2.     private Object o;
3.     void f() {
4.         o = new Object();
5.         this.g(o);
6.     }
7.     void g(Object obj) {
8.         obj = null;
9.         obj = new Object();
10.    }
11.    public static void main(String[] a){
12.        new X().f();
13.    }
14. }
```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 8.
- C. Al termine della linea 9.
- D. Al termine della linea 12.
- E. Non si può dire.

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

```

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

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

3. Se nella classe C è dichiarato il metodo

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

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

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

4. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A().new B();
    }
}
```

- A. 0
- B. 1
- 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. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Ai metodi `static` non si applica il *dynamic method dispatch*
- C. Una classe non può essere dichiarata `final`
- D. Un array vuoto può avere riferimento `null`
- E. L'operatore `new` non restituisce un riferimento all'oggetto appena creato

6. Quale output si ottiene invocando il metodo p?

```
class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
          i4, i3);
    }
    void m(String s1, Boolean b2,
           Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 000
- B. 101
- C. 100
- D. 011
- E. 001

7. Qual è l'output di questo codice?

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

class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

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

    public static void main(String[] args){
        A a = new B();
```

```
        a.m();
    }
}
```

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

8. Date le dichiarazioni:

```
class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B2 c;
C2 s;
C0 x;
c = new B2();
x = new C2();
s = new C2();
```

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

- A. c = (B2) x;
- B. s = (C2) x;
- C. c = (B2) s;
- D. s = (C2) c;
- E. Nessuno dei precedenti

9. Qual è l'output di questo codice?

```
class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    B() {
        super( 2 );
    }
    public static void main( String [] a) {
        new B();
    }
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuna delle precedenti

10. Qual'è l'output di questo codice?

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

A. 423Exception in thread "main" MyExc3

B. 4222222... (ciclo infinito)

C. Errore a tempo di compilazione

D. 42Exception in thread "main" MyExc3

E. Nessuna delle precedenti

11. Date le dichiarazioni:

```
Boolean [] c;
Object [] [] s;
Object [] w;
```

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

A. c = (Boolean []) s;

B. s = w;

C. c = w;

D. s = (Object [] []) c;

E. w = (Object []) s;

12. Qual è l'output di questo codice?

```
class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    public static void main( String [] a ) {
        new B();
    }
}
```

A. 0

B. 1

C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

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

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

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

public static void main(String[] args){
    I i = new B();
    i.m();
}
```

A. A.f

B. B.m

C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuna delle precedenti

14. Qual'è l'output di questo codice?


```

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

```

- A. Errore a tempo di compilazione
- B. 42
- C. 341
- D. 41
- E. Nessuna delle precedenti

15. Qual'è l'output di questo codice?

```

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

```

```

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

```

- A. 111... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. Exception in thread "main" MyExc2
- D. 2Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}

```

- A. 0
- B. 1
- 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. Se nella classe C è dichiarato il metodo

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

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

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

2. Quale output si ottiene invocando il metodo p?

```
class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
          i4, i3);
    }
    void m(String s1, Boolean b2,
           Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 000

- B. 101
- C. 100
- D. 011
- E. 001

3. Qual è l'output di questo codice?

```
class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}
```

```
public class B extends A {
    public static void main( String [] a) {
        new B();
    }
}
```

- A. 0
- B. 1
- 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 Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
    }
    static void q() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc2 d ) {
            System.out.print(1);
        }
    }
}
```

```

        throw( new MyExc3() );
    }
    catch( Exception h ) {
        throw( new MyExc2() );
    }
    catch( MyExc3 w ) {
    }
    finally {
        System.out.print(2);
    }
}
}

```

- A. 111... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. Exception in thread "main" MyExc2
- D. 2Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A(). new B();
    }
}

```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```

public interface I {
    void m();
}

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

class B extends A {

```

```

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

    public static void main(String[] args){
        I i = new B();
        i.m();
    }
}

```

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

7. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}

```

- A. 0
- B. 1
- 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 D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 r ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
}

```

```

static void p() {
    try {
        System.out.print(4);
        throw( new Exception() );
    }
    catch( MyExc3 a ) {
    }
    finally {
        throw( new MyExc3() );
    }
}

```

- A. 423Exception in thread "main" MyExc3
- B. 42222222... (ciclo infinito)
- C. Errore a tempo di compilazione
- D. 42Exception in thread "main" MyExc3
- E. Nessuna delle precedenti

9. Date le dichiarazioni:

```

class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B2 c;
C2 s;
C0 x;
c = new B2();
x = new C2();
s = new C2();

```

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

- A. c = (B2) x;
- B. s = (C2) x;
- C. c = (B2) s;
- D. s = (C2) c;
- E. Nessuno dei precedenti

10. Qual'è l'output di questo codice?

```

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

```

```

catch( MyExc2 h ) {
    System.out.print(2);
}
}
static void q() throws Exception {
    try {
        throw( new MyExc2() );
    }
    catch( Exception w ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. Errore a tempo di compilazione
- B. 42
- C. 341
- D. 41
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```

class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    B() {
        super( 2 );
    }
    public static void main( String [] a ) {
        new B();
    }
}

```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

12. Date le dichiarazioni:

```

Boolean [] c;
Object [] [] s;
Object [] w;

```

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

- A. `c = (Boolean []) s;`
 - B. `s = w;`
 - C. `c = w;`
 - D. `s = (Object [] []) c;`
 - E. `w = (Object []) s;`
-

13. Data la classe

```
1. class X {
2.     private Object o;
3.     void f() {
4.         o = new Object();
5.         this.g(o);
6.     }
7.     void g(Object obj) {
8.         obj = null;
9.         obj = new Object();
10.    }
11.    public static void main(String[] a){
12.        new X().f();
13.    }
14. }
```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
 - B. Al termine della linea 8.
 - C. Al termine della linea 9.
 - D. Al termine della linea 12.
 - E. Non si può dire.
-

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

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

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

15. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
 - B. Ai metodi `static` non si applica il *dynamic method dispatch*
 - C. Una classe non può essere dichiarata `final`
 - D. Un array vuoto può avere riferimento `null`
 - E. L'operatore `new` non restituisce un riferimento all'oggetto appena creato
-

16. Qual è l'output di questo codice?

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

class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

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

public static void main(String[] args){
    A a = new B();
    a.m();
}
```

- A. A.f
 - B. B.m
 - 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 Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
    }
    static void q() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc2 d ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception h ) {
            throw( new MyExc2() );
        }
        catch( MyExc3 w ) {
        }
        finally {
            System.out.print(2);
        }
    }
}
```

- A. 111... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. Exception in thread "main" MyExc2
- D. 2Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

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

class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}
```

```

    }
}

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

    public static void main(String[] args){
        A a = new B();
        a.m();
    }
}
```

- A. A.f
- B. B.m
- 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 A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    B() {
        super( 2 );
    }
    public static void main( String [] a) {
        new B();
    }
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione

E. Nessuna delle precedenti

4. Se nella classe C è dichiarato il metodo

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

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

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

5. Date le dichiarazioni:

```
class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B2 c;
C2 s;
C0 x;
c = new B2 ();
x = new C2 ();
s = new C2 ();
```

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

- A. `c = (B2) x;`
- B. `s = (C2) x;`
- C. `c = (B2) s;`
- D. `s = (C2) c;`
- E. Nessuno dei precedenti

6. Qual è l'output di questo codice?

```
public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A(). new B();
    }
}
```

A. 0

B. 1

C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuna delle precedenti

7. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Ai metodi **static** non si applica il *dynamic method dispatch*
- C. Una classe non può essere dichiarata **final**
- D. Un array vuoto può avere riferimento **null**
- E. L'operatore **new** non restituisce un riferimento all'oggetto appena creato

8. Date le dichiarazioni:

```
Boolean [] c;
Object [] [] s;
Object [] w;
```

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

- A. `c = (Boolean []) s;`
- B. `s = w;`
- C. `c = w;`
- D. `s = (Object [] []) c;`
- E. `w = (Object []) s;`

9. Data la classe

```
1. class X {
2.     private Object o;
3.     void f() {
4.         o = new Object();
5.         this.g(o);
6.     }
7.     void g(Object obj) {
8.         obj = null;
9.         obj = new Object();
10.    }
11.    public static void main(String[] a) {
12.        new X().f();
13.    }
14. }
```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.

- B. Al termine della linea 8.
- C. Al termine della linea 9.
- D. Al termine della linea 12.
- E. Non si può dire.

10. Qual'è l'output di questo codice?

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

- A. 423Exception in thread "main" MyExc3
- B. 42222222... (ciclo infinito)
- C. Errore a tempo di compilazione
- D. 42Exception in thread "main" MyExc3
- E. Nessuna delle precedenti

11. Qual'è l'output di questo codice?

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

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

- A. Errore a tempo di compilazione
- B. 42
- C. 341
- D. 41
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

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

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

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

    public static void main(String[] args){
        I i = new B();
        i.m();
    }
}
```

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

13. Quale output si ottiene invocando il metodo p?

```

class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
            i4, i3);
    }
    void m(String s1, Boolean b2,
        Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 000
- B. 101
- C. 100
- D. 011
- E. 001

14. Qual è l'output di questo codice?

```

class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    public static void main( String [] a) {
        new B();
    }
}

```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

15. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}

```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

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

```

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

```

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

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. Date le dichiarazioni:

```
class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B2 c;
C2 s;
C0 x;
c = new B2 ();
x = new C2 ();
s = new C2 ();
```

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

- A. c = (B2) x;
- B. s = (C2) x;
- C. c = (B2) s;
- D. s = (C2) c;
- E. Nessuno dei precedenti

2. Qual'è l'output di questo codice?

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

```
        throw( new Exception() );
        }
        catch( MyExc3 a ) {
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. 423Exception in thread "main" MyExc3
- B. 42222222... (ciclo infinito)
- C. Errore a tempo di compilazione
- D. 42Exception in thread "main" MyExc3
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    B() {
        super( 2 );
    }
    public static void main( String [] a) {
        new B();
    }
}
```

- A. 0
- B. 1
- 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 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 h ) {
            System.out.print(2);
        }
    }
    static void q() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( Exception w ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 42
- C. 341
- D. 41
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```

public interface I {
    void m();
}

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

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

    public static void main(String[] args){
        I i = new B();
        i.m();
    }
}

```

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

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

```

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

```

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

7. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A(). new B();
    }
}

```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

8. Data la classe

```

1. class X {
2.     private Object o;
3.     void f() {
4.         o = new Object();
5.         this.g(o);
6.     }
7.     void g(Object obj) {
8.         obj = null;

```

```

9.         obj = new Object ();
10.      }
11.      public static void main(String[] a){
12.          new X().f();
13.      }
14.  }

```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 8.
- C. Al termine della linea 9.
- D. Al termine della linea 12.
- E. Non si può dire.

9. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Ai metodi `static` non si applica il *dynamic method dispatch*
- C. Una classe non può essere dichiarata `final`
- D. Un array vuoto può avere riferimento `null`
- E. L'operatore `new` non restituisce un riferimento all'oggetto appena creato

10. Se nella classe C è dichiarato il metodo

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

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

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

11. Qual è l'output di questo codice?

```

public interface I {
    void m();
}

class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

```

```

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

    public static void main(String[] args){
        A a = new B();
        a.m();
    }
}

```

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

12. Quale output si ottiene invocando il metodo p?

```

class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
            i4, i3);
    }
    void m(String s1, Boolean b2,
        Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 000
- B. 101
- C. 100
- D. 011
- E. 001

13. Date le dichiarazioni:

```
Boolean [] c;
Object [] [] s;
Object [] w;
```

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

- A. `c = (Boolean []) s;`
- B. `s = w;`
- C. `c = w;`
- D. `s = (Object [] []) c;`
- E. `w = (Object []) s;`

14. Qual è l'output di questo codice?

```
class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    public static void main( String [] a ) {
        new B();
    }
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

15. Qual'è l'output di questo codice?

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

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

- A. 111... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. Exception in thread "main" MyExc2
- D. 2Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```
public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}
```

- A. 0
 - B. 1
 - 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. 17

1. Qual'è l'output di questo codice?

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

- A. 111... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. Exception in thread "main" MyExc2
- D. 2Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

2. Se nella classe C è dichiarato il metodo

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

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

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

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

3. Quale output si ottiene invocando il metodo p?

```
class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
            i4, i3);
    }
    void m(String s1, Boolean b2,
        Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 000
- B. 101
- C. 100
- D. 011
- E. 001

4. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Ai metodi `static` non si applica il *dynamic method dispatch*
- C. Una classe non può essere dichiarata `final`
- D. Un array vuoto può avere riferimento `null`
- E. L'operatore `new` non restituisce un riferimento all'oggetto appena creato

5. Qual è l'output di questo codice?

```
class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    B() {
        super( 2 );
    }
    public static void main( String [] a) {
        new B();
    }
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```
public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione

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

7. Qual è l'output di questo codice?

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

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

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

    public static void main(String[] args) {
        I i = new B();
        i.m();
    }
}
```

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

8. Date le dichiarazioni:

```
class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B2 c;
C2 s;
C0 x;
c = new B2();
x = new C2();
s = new C2();
```

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

- A. `c = (B2) x;`
- B. `s = (C2) x;`
- C. `c = (B2) s;`
- D. `s = (C2) c;`
- E. Nessuno dei precedenti

9. Qual è l'output di questo codice?

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

class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

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

    public static void main(String[] args) {
        A a = new B();
        a.m();
    }
}
```

- A. A.f
- B. B.m
- 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 A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    public static void main( String [] a) {
        new B();
    }
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

11. Data la classe

```
1. class X {
2.     private Object o;
3.     void f() {
4.         o = new Object();
5.         this.g(o);
6.     }
7.     void g(Object obj) {
8.         obj = null;
9.         obj = new Object();
10.    }
11.    public static void main(String[] a){
12.        new X().f();
13.    }
14. }
```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 8.
- C. Al termine della linea 9.
- D. Al termine della linea 12.
- E. Non si può dire.

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

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

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

13. Date le dichiarazioni:

```
Boolean [] c;
Object [] [] s;
Object [] w;
```

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

- A. c = (Boolean []) s;
- B. s = w;
- C. c = w;
- D. s = (Object [] []) c;
- E. w = (Object []) s;

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 h ) {
            System.out.print(2);
        }
    }
    static void q() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( Exception w ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 42
- C. 341
- D. 41
- E. Nessuna delle precedenti

15. Qual'è l'output di questo codice?

```

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

```

```

        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 a ) {
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 423Exception in thread "main" MyExc3
- B. 42222222... (ciclo infinito)
- C. Errore a tempo di compilazione
- D. 42Exception in thread "main" MyExc3
- E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A(). new B();
    }
}

```

- A. 0
- B. 1
- 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. 18

1. Date le dichiarazioni:

```
Boolean [] c;
Object [] [] s;
Object [] w;
```

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

- A. `c = (Boolean []) s;`
- B. `s = w;`
- C. `c = w;`
- D. `s = (Object [] []) c;`
- E. `w = (Object []) s;`

2. Qual è l'output di questo codice?

```
class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    public static void main( String [] a ) {
        new B();
    }
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

3. Dire quale delle seguenti affermazioni è vera:

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

- B. Ai metodi `static` non si applica il *dynamic method dispatch*
- C. Una classe non può essere dichiarata `final`
- D. Un array vuoto può avere riferimento `null`
- E. L'operatore `new` non restituisce un riferimento all'oggetto appena creato

4. Qual è l'output di questo codice?

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

class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

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

public static void main(String[] args){
    A a = new B();
    a.m();
}
```

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

5. Quale output si ottiene invocando il metodo `p`?

```
class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
```

```

    i4 = i3;
    m(new String("abcde"), new Boolean(true),
      i4, i3);
}
void m(String s1, Boolean b2,
      Integer i1, Integer i2) {
    Boolean b1 = new Boolean(true);
    if(s1 == s2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(i1 == i2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}
}

```

- A. 000
- B. 101
- C. 100
- D. 011
- E. 001

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

```

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

```

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

7. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);

```

```

        }
        catch( MyExc3 r ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
}
static void p() {
    try {
        System.out.print(4);
        throw( new Exception() );
    }
    catch( MyExc3 a ) {
    }
    finally {
        throw( new MyExc3() );
    }
}
}

```

- A. 423Exception in thread "main" MyExc3
- B. 42222222... (ciclo infinito)
- C. Errore a tempo di compilazione
- D. 42Exception in thread "main" MyExc3
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A(). new B();
    }
}

```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

9. Qual'è l'output di questo codice?


```

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

```

- A. Errore a tempo di compilazione
- B. 42
- C. 341
- D. 41
- E. Nessuna delle precedenti

10. Date le dichiarazioni:

```

class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B2 c;
C2 s;
C0 x;
c = new B2 ();
x = new C2 ();
s = new C2 ();

```

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

- A. c = (B2) x;
- B. s = (C2) x;
- C. c = (B2) s;
- D. s = (C2) c;
- E. Nessuno dei precedenti

11. Qual è l'output di questo codice?

```

public interface I {
    void m();
}

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

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

    public static void main(String[] args){
        I i = new B();
        i.m();
    }
}

```

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

12. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}

```

- A. 0
- B. 1
- 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 A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

```

```

public class B extends A {
    B() {
        super( 2 );
    }
    public static void main( String [] a ) {
        new B();
    }
}

```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

14. Qual'è l'output di questo codice?

```

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

```

}

- A. 111... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. Exception in thread "main" MyExc2
- D. 2Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

15. Se nella classe C è dichiarato il metodo

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

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

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

16. Data la classe

```

1. class X {
2.     private Object o;
3.     void f() {
4.         o = new Object();
5.         this.g(o);
6.     }
7.     void g(Object obj) {
8.         obj = null;
9.         obj = new Object();
10.    }
11.    public static void main(String[] a){
12.        new X().f();
13.    }
14. }

```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 8.
- C. Al termine della linea 9.
- D. Al termine della linea 12.
- E. Non si può dire.

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. Quale output si ottiene invocando il metodo p?

```
class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
            i4, i3);
    }
    void m(String s1, Boolean b2,
        Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 000
- B. 101
- C. 100
- D. 011
- E. 001

2. Qual è l'output di questo codice?

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

class A implements I {
    public void f() {
```

```
        System.out.println("A.f");
    }
}

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

    public static void main(String[] args){
        A a = new B();
        a.m();
    }
}
```

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

3. Qual è l'output di questo codice?

```
public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}
```

- A. 0
- B. 1
- 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 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 h ) {
            System.out.print(2);
        }
    }
    static void q() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( Exception w ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 42
- C. 341
- D. 41
- E. Nessuna delle precedenti

5. Date le dichiarazioni:

```

Boolean [] c;
Object [] [] s;
Object [] w;

```

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

- A. `c = (Boolean []) s;`
- B. `s = w;`
- C. `c = w;`
- D. `s = (Object [] []) c;`
- E. `w = (Object []) s;`

6. Se nella classe C è dichiarato il metodo

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

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

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

7. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A(). new B();
    }
}

```

- A. 0
- B. 1
- 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 D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 r ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 a ) {
        }
        finally {

```

```

        throw( new MyExc3() );
    }
}

```

- A. 423Exception in thread "main" MyExc3
 - B. 42222222... (ciclo infinito)
 - C. Errore a tempo di compilazione
 - D. 42Exception in thread "main" MyExc3
 - E. Nessuna delle precedenti
-

9. Date le dichiarazioni:

```

class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B2 c;
C2 s;
C0 x;
c = new B2 ();
x = new C2 ();
s = new C2 ();

```

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

- A. c = (B2) x;
 - B. s = (C2) x;
 - C. c = (B2) s;
 - D. s = (C2) c;
 - E. Nessuno dei precedenti
-

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

```

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

```

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

11. Data la classe

```

1. class X {
2.     private Object o;
3.     void f() {
4.         o = new Object ();
5.         this.g(o);
6.     }
7.     void g(Object obj) {
8.         obj = null;
9.         obj = new Object ();
10.    }
11.    public static void main(String[] a) {
12.        new X().f ();
13.    }
14. }

```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
 - B. Al termine della linea 8.
 - C. Al termine della linea 9.
 - D. Al termine della linea 12.
 - E. Non si può dire.
-

12. Qual è l'output di questo codice?

```

class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    public static void main( String [] a) {
        new B();
    }
}

```

- A. 0
 - B. 1
 - C. Errore a tempo di compilazione
 - D. Errore a tempo di esecuzione
 - E. Nessuna delle precedenti
-

13. Qual è l'output di questo codice?

```

public interface I {
    void m();
}

abstract class A implements I {
    public void f() {

```

```

        System.out.println("A.f");
    }
}

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

    public static void main(String[] args){
        I i = new B();
        i.m();
    }
}

```

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

14. Qual'è l'output di questo codice?

```

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

```

- A. 111... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. Exception in thread "main" MyExc2
- D. 2Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

15. Qual è l'output di questo codice?

```

class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    B() {
        super( 2 );
    }
    public static void main( String [] a ) {
        new B();
    }
}

```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

16. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Ai metodi **static** non si applica il *dynamic method dispatch*
- C. Una classe non può essere dichiarata **final**
- D. Un array vuoto può avere riferimento **null**
- E. L'operatore **new** non restituisce un riferimento all'oggetto appena creato

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

Prova n. 20

1. Qual'è l'output di questo codice?

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

- A. 111... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. Exception in thread "main" MyExc2
- D. 2Exception in thread "main" MyExc2
- 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 h ) {
            System.out.print(2);
        }
    }
    static void q() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( Exception w ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 42
- C. 341
- D. 41
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

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

class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}
```

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

    public static void main(String[] args){
        A a = new B();
```

```

        a.m();
    }
}

```

- A. A.f
- B. B.m
- 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 A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    B() {
        super( 2 );
    }
    public static void main( String [] a ) {
        new B();
    }
}

```

- A. 0
- B. 1
- 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. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Ai metodi `static` non si applica il *dynamic method dispatch*
- C. Una classe non può essere dichiarata `final`
- D. Un array vuoto può avere riferimento `null`
- E. L'operatore `new` non restituisce un riferimento all'oggetto appena creato

6. Se nella classe C è dichiarato il metodo

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

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

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

7. Date le dichiarazioni:

```

Boolean [] c;
Object [] [] s;
Object [] w;

```

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

- A. `c = (Boolean []) s;`
- B. `s = w;`
- C. `c = w;`
- D. `s = (Object [] []) c;`
- E. `w = (Object []) s;`

8. Qual è l'output di questo codice?

```

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

```

- A. 423Exception in thread "main" MyExc3
- B. 42222222... (ciclo infinito)

- C. Errore a tempo di compilazione
- D. 42Exception in thread "main" MyExc3
- E. Nessuna delle precedenti

E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

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

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

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

    public static void main(String[] args) {
        I i = new B();
        i.m();
    }
}
```

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

10. Qual è l'output di questo codice?

```
public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A(). new B();
    }
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione

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

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

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

12. Date le dichiarazioni:

```
class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B2 c;
C2 s;
C0 x;
c = new B2();
x = new C2();
s = new C2();
```

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

- A. c = (B2) x;
- B. s = (C2) x;
- C. c = (B2) s;
- D. s = (C2) c;
- E. Nessuno dei precedenti

13. Qual è l'output di questo codice?

```
class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    public static void main( String [] a) {
        new B();
    }
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

14. Data la classe

```
1. class X {
2.     private Object o;
3.     void f() {
4.         o = new Object();
5.         this.g(o);
6.     }
7.     void g(Object obj) {
8.         obj = null;
9.         obj = new Object();
10.    }
11.    public static void main(String[] a){
12.        new X().f();
13.    }
14. }
```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 8.
- C. Al termine della linea 9.
- D. Al termine della linea 12.
- E. Non si può dire.

15. Qual è l'output di questo codice?

```
public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}
```

16. Quale output si ottiene invocando il metodo p?

```
class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
            i4, i3);
    }
    void m(String s1, Boolean b2,
        Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 000
 - B. 101
 - C. 100
 - D. 011
 - E. 001
-

Prova n. 1

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1

Prova d'esame

prof. Piero A. Bonatti

12 gennaio 2017

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

12 gennaio 2017

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: