



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

**Prova n. 1**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
```

```
public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

3. Qual è l'output di questo codice?

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

- A. 42
  - B. 41
  - C. 4
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Dato il template

```
class Nodo<T> {  
    private T info;  
    Nodo<T> next;  
    Nodo(T i) { info=i; }  
    public T getInfo() { return info; }  
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
  - B. `Nodo<String> n = new Nodo<String>("");`
  - C. `new Nodo<Boolean>("");`
  - D. `new Nodo<Boolean>(true).getInfo().length();`
  - E. Nessuno dei precedenti
- 

5. Date le dichiarazioni:

```
Object [] e;  
Boolean [] f;  
Exception [] u;  
u = new Exception [8];  
e = new Exception [1];  
f = new Boolean [5];
```

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

- A. `f = (Boolean []) e;`
  - B. `f = (Boolean []) u;`
  - C. `u = (Exception []) e;`
  - D. `u = (Exception []) f;`
  - E. Nessuno dei precedenti
- 

6. Quale risultato produce questo codice?

```
class A {  
    int x;  
    A() {  
        x = 1;  
        System.out.print(x);  
    }  
}  
public class B extends A {  
    float x;  
    B() {
```

```
        x = 2;  
        System.out.print(x);  
    }  
    public static void main(String a[]) {  
        new B();  
    }  
}
```

- A. 1
  - B. 2.0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

7. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi `static` non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata `private`
- 

8. Date le dichiarazioni:

```
Boolean e;  
Exception m;  
Object v;
```

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

- A. `m = v;`
  - B. `e = (Boolean) m;`
  - C. `e = v;`
  - D. `m = (Exception) e;`
  - E. `v = e;`
- 

9. Quale espressione va inserita nel punto indicato per stampare l'attributo `x` dichiarato nella classe `A`?

```
class A {  
    int x = 1;  
  
    public class B {  
        protected int x = 2;  
        public void my_print() {  
            System.out.println( _QUI_ );  
        }  
    }  
}
```

- A. `x`
- B. `A.this.x`

- C. A.x
- D. super.x
- E. Nessuna delle precedenti

10. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}
```

- }
- }
- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

12. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

13. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
```

```

        System.out.print(0);
    }
    if(s4 == s3) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

---

14. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

---

15. Date la classe

```

class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}

```

e una variabile `n` di tipo `Nodo` tale che `n.info=""`, dire quale dei loro usi riportati qui sotto è corretto:

- A. `String s = n.getInfo();`
- B. `int l = n.getInfo().length();`
- C. `n=new Nodo(new Integer(0));`
- D. `int v = n.getInfo().intValue();`
- E. Nessuno dei precedenti

---

16. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- 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. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

2. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
```

```
B obj = new B();
obj.x = 1;
obj.set_x(2);
System.out.println( obj.x );
}
```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}
```

- A. 23431
- B. Errore a tempo di compilazione

- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi **static** non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata *private*

4. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}
```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

5. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

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

- A. `f = (Boolean []) e;`
- B. `f = (Boolean []) u;`
- C. `u = (Exception []) e;`
- D. `u = (Exception []) f;`
- E. Nessuno dei precedenti

6. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null

7. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
- B. `Nodo<String> n = new Nodo<String>("");`
- C. `new Nodo<Boolean>("");`
- D. `new Nodo<Boolean>(true).getInfo().length();`
- E. Nessuno dei precedenti

8. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

9. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile `n` di tipo `Nodo` tale che `n.info=""`,  
dire quale dei loro usi riportati qui sotto è corretto:



- A. `String s = n.getInfo();`
- B. `int l = n.getInfo().length();`
- C. `n=new Nodo(new Integer(0));`
- D. `int v = n.getInfo().intValue();`
- E. Nessuno dei precedenti

10. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

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

- A. `m = v;`
- B. `e = (Boolean) m;`
- C. `e = v;`
- D. `m = (Exception) e;`
- E. `v = e;`

11. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. `x`
- B. `A.this.x`
- C. `A.x`
- D. `super.x`
- E. Nessuna delle precedenti

12. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

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

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

14. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- 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 C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
    }
}

```

```

}
catch( MyExc1 g ) {
    System.out.print(4);
}
catch( MyExc3 e ) {
    throw( new MyExc1() );
}
finally {
    System.out.print(5);
}
}
}

```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

16. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - 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 risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

2. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
- B. `Nodo<String> n = new Nodo<String>("");`
- C. `new Nodo<Boolean>("");`
- D. `new Nodo<Boolean>(true).getInfo().length();`
- E. Nessuno dei precedenti

3. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

4. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

---

5. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

---

6. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
- B. int l = n.getInfo().length();
- C. n=new Nodo(new Integer(0));
- D. int v = n.getInfo().intValue();
- E. Nessuno dei precedenti

---

7. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
```

```
            System.out.print(0);
        }
    }
    if(s2 == s1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s4 == s3) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

---

8. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

---

9. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

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

- A. m = v;
- B. e = (Boolean) m;
- C. e = v;
- D. m = (Exception) e;

E. v = e;

---

10. Qual è l'output di questo codice?

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

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

11. Qual è l'output di questo codice?

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

```
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

12. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi **static** non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata *private*

---

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
    }
}
```

```

    catch( Exception e ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
        throw( new Exception() );
    }
}
}

```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

---

14. Quale risultato produce questo codice?

```

class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}

```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

---

15. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

---

16. Date le dichiarazioni:

```

Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];

```

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

- A. f = (Boolean []) e;
  - B. f = (Boolean []) u;
  - C. u = (Exception []) e;
  - D. u = (Exception []) f;
  - E. Nessuno dei precedenti
-





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

**Prova n. 4**

1. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception (8);
e = new Exception (1);
f = new Boolean (5);
```

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

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;
- C. u = (Exception []) e;
- D. u = (Exception []) f;
- E. Nessuno dei precedenti

2. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
- B. int l = n.getInfo().length();
- C. n=new Nodo(new Integer(0));
- D. int v = n.getInfo().intValue();
- E. Nessuno dei precedenti

3. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }
```

```
public static void main(String a[]) {
    C obj = new C( new Integer(1) );
    obj = null; // QUESTA
    System.out.println("done");
}
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

4. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

5. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
```

```

public static void main(String[] s) {
    A a = new B();
    B b = (B) a;
    System.out.println( a.get() + b.get() );
}

```

- A. 12
- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

```

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

```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

#### 6. Dato il template

```

class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}

```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
- B. `Nodo<String> n = new Nodo<String>("");`
- C. `new Nodo<Boolean>("");`
- D. `new Nodo<Boolean>(true).getInfo().length();`
- E. Nessuno dei precedenti

#### 7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
    }
}

```

#### 8. Quale risultato produce questo codice?

```

class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}

```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

#### 9. Qual è l'output di questo codice?

```

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

```

```

        catch( MyExc3 u ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

10. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

11. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);

```

```

        } else {
            System.out.print(0);
        }
    }
    if(s2 == s1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s4 == s3) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452
  - B. 352Exception in thread "main" MyExc2
  - C. 35
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

14. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

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

- A. m = v;
  - B. e = (Boolean) m;
  - C. e = v;
  - D. m = (Exception) e;
  - E. v = e;
- 

15. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi static non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata private
- 

16. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
  - B. 2
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
-



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

**Prova n. 5**

1. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

2. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

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

- A. `m = v;`
- B. `e = (Boolean) m;`
- C. `e = v;`
- D. `m = (Exception) e;`
- E. `v = e;`

3. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile `n` di tipo `Nodo` tale che `n.info=""`, dire quale dei loro usi riportati qui sotto è corretto:

- A. `String s = n.getInfo();`
- B. `int l = n.getInfo().length();`
- C. `n=new Nodo(new Integer(0));`
- D. `int v = n.getInfo().intValue();`
- E. Nessuno dei precedenti

4. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
- B. `Nodo<String> n = new Nodo<String>("");`
- C. `new Nodo<Boolean>("");`
- D. `new Nodo<Boolean>(true).getInfo().length();`
- E. Nessuno dei precedenti

5. Quale espressione va inserita nel punto indicato per stampare l'attributo `x` ereditato dalla classe `A`?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}
```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

7. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
```

```
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}
```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

8. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

9. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
- B. 22
- C. 3

D. 4

E. Nessuno dei precedenti

---

10. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}
```

A. x

B. A.this.x

C. A.x

D. super.x

E. Nessuna delle precedenti

---

11. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

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

A. f = (Boolean []) e;

B. f = (Boolean []) u;

C. u = (Exception []) e;

D. u = (Exception []) f;

E. Nessuno dei precedenti

---

12. Qual è l'output di questo codice?

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

```
    }
    catch( MyExc2 y ) {
        System.out.print(2);
    }
    catch( MyExc3 u ) {
        System.out.print(3);
    }
}
static void m() throws Exception {
    try {
        throw( new MyExc2() );
    }
    finally {
        System.out.print(4);
    }
}
}
```

A. 42

B. 41

C. 4

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

13. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

A. 110

B. 011

C. 001

D. 101



14. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

15. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi **static** non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata *private*

16. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti



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

**Prova n. 6**

1. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

2. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

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

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;
- C. u = (Exception []) e;
- D. u = (Exception []) f;
- E. Nessuno dei precedenti

3. Qual è l'output di questo codice?

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

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

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
```

```

float x;
B() {
    x = 2;
    System.out.print(x);
}
public static void main(String a[]) {
    new B();
}
}

```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

5. Qual è il risultato di questo codice:

```

class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}

```

- A. 12
- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {

```

```

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

```

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

8. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

9. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

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

- A. `m = v;`
- B. `e = (Boolean) m;`
- C. `e = v;`
- D. `m = (Exception) e;`
- E. `v = e;`

10. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi **static** non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata **private**

11. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
```

```
public void my_print() {
    System.out.println( _QUI_ );
}
public static void main(String a[]) {
    B obj = new B();
    obj.x = 1;
    obj.set_x(2);
    obj.my_print();
}
}
```

- A. `x`
- B. `A.this.x`
- C. `A.x`
- D. `super.x`
- E. Nessuna delle precedenti

12. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
- B. `Nodo<String> n = new Nodo<String>("");`
- C. `new Nodo<Boolean>("");`
- D. `new Nodo<Boolean>(true).getInfo().length();`
- E. Nessuno dei precedenti

13. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

#### 14. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile `n` di tipo `Nodo` tale che `n.info=""`,  
dire quale dei loro usi riportati qui sotto è corretto:

- A. `String s = n.getInfo();`
- B. `int l = n.getInfo().length();`
- C. `n=new Nodo(new Integer(0));`
- D. `int v = n.getInfo().intValue();`
- E. Nessuno dei precedenti

---

#### 15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
    }
}
```

```
catch( Exception e ) {
    System.out.print(3);
}
finally {
    System.out.print(4);
    throw( new Exception() );
}
}
```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

---

#### 16. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
  - B. 2
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
-



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

**Prova n. 7**

1. Qual è l'output di questo codice?

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

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
- B. `Nodo<String> n = new Nodo<String>("");`
- C. `new Nodo<Boolean>("");`
- D. `new Nodo<Boolean>(true).getInfo().length();`
- E. Nessuno dei precedenti

3. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

4. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. `String s = n.getInfo();`
- B. `int l = n.getInfo().length();`
- C. `n=new Nodo(new Integer(0));`
- D. `int v = n.getInfo().intValue();`
- E. Nessuno dei precedenti



---

5. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}
```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

---

6. Qual è l'output di questo codice?

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

```
    }
}
static void n() throws Exception {
    try {
        System.out.print(3);
        if (true) throw( new MyExc2() );
        else throw( new MyExc3() );
    }
    catch( MyExc1 g ) {
        System.out.print(4);
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(5);
    }
}
}
```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

7. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

---

8. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
}
```

```

    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}

```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

9. Quale risultato produce questo codice?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}

```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

10. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```

if(s2 == s1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s4 == s3) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
}

```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

11. Date le dichiarazioni:

```

Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];

```

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

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;
- C. u = (Exception []) e;
- D. u = (Exception []) f;
- E. Nessuno dei precedenti

12. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

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

E. Nessuna delle precedenti

---

13. Date le dichiarazioni:

```
Boolean e;  
Exception m;  
Object v;
```

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

- A. `m = v;`
  - B. `e = (Boolean) m;`
  - C. `e = v;`
  - D. `m = (Exception) e;`
  - E. `v = e;`
- 

14. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi `static` non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata `private`
- 

15. Quale risultato produce questo codice?

```
class A {  
    int x;  
    A() {  
        x = 1;  
        System.out.print(x);  
    }  
}  
  
public class B extends A {
```

```
    float x;  
    B() {  
        x = 2;  
        System.out.print(x);  
    }  
    public static void main(String a[]) {  
        new B();  
    }  
}
```

- A. 1
  - B. 2.0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

16. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {  
    String x = "abc";  
    static Integer y;  
  
    C(Integer y) { this.y = y; }  
  
    public static void main(String a[]) {  
        C obj = new C( new Integer(1) );  
        obj = null; // QUESTA  
        System.out.println("done");  
    }  
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - 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. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

2. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
```

```
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}
```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

3. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

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

```

        m();
    }
    catch( Exception x ) {
        System.out.print(1);
    }
    catch( MyExc2 y ) {
        System.out.print(2);
    }
    catch( MyExc3 u ) {
        System.out.print(3);
    }
}
static void m() throws Exception {
    try {
        throw( new MyExc2() );
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}

```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione

- C. Un array possiede dei membri
  - D. Ai metodi `static` non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata `private`
- 

8. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

9. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

10. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

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

- A. `m = v;`
  - B. `e = (Boolean) m;`
  - C. `e = v;`
  - D. `m = (Exception) e;`
  - E. `v = e;`
- 

11. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
  - B. 22
  - C. 3
  - D. 4
  - E. Nessuno dei precedenti
- 

12. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
  - B. `Nodo<String> n = new Nodo<String>("");`
  - C. `new Nodo<Boolean>("");`
  - D. `new Nodo<Boolean>(true).getInfo().length();`
  - E. Nessuno dei precedenti
- 

13. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
  - B. int l = n.getInfo().length();
  - C. n=new Nodo(new Integer(0));
  - D. int v = n.getInfo().intValue();
  - E. Nessuno dei precedenti
- 

14. Date le dichiarazioni:

```
Object [] e;  
Boolean [] f;  
Exception [] u;  
u = new Exception [8];  
e = new Exception [1];  
f = new Boolean [5];
```

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

- A. f = (Boolean []) e;
  - B. f = (Boolean []) u;
  - C. u = (Exception []) e;
  - D. u = (Exception []) f;
  - E. Nessuno dei precedenti
- 

15. Quale risultato produce questo codice?

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}  
public class B extends A {  
    protected int x;  
    public static void main(String a[]) {  
        B obj = new B();  
        obj.x = 1;
```

```
        obj.set_x(2);  
        System.out.println( obj.x );  
    }  
}
```

- A. 1
  - B. 2
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

16. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}  
public class B extends A {  
    protected int x;  
    public void my_print() {  
        System.out.println( _QUI_ );  
    }  
    public static void main(String a[]) {  
        B obj = new B();  
        obj.x = 1;  
        obj.set_x(2);  
        obj.my_print();  
    }  
}
```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - 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. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

- D. super.x  
E. Nessuna delle precedenti

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1  
B. 2  
C. 3  
D. 4  
E. Nessuna delle precedenti

2. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x  
B. A.this.x  
C. A.x

3. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110  
B. 011  
C. 001  
D. 101  
E. 010

4. Date la classe

```
class Nodo {
    private Object info;
```

```

    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}

```

è una variabile n di tipo Nodo tale che n.info="",  
dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
- B. int l = n.getInfo().length();
- C. n=new Nodo(new Integer(0));
- D. int v = n.getInfo().intValue();
- E. Nessuno dei precedenti

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}

```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```

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

```

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Quale risultato produce questo codice?

```

class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}

```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione

E. Nessuno dei precedenti

---

8. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi `static` non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata `private`
- 

9. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
  - B. 22
  - C. 3
  - D. 4
  - E. Nessuno dei precedenti
- 

10. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

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

- A. `m = v;`
  - B. `e = (Boolean) m;`
  - C. `e = v;`
  - D. `m = (Exception) e;`
  - E. `v = e;`
- 

11. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

12. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

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

- A. `f = (Boolean []) e;`
  - B. `f = (Boolean []) u;`
  - C. `u = (Exception []) e;`
  - D. `u = (Exception []) f;`
  - E. Nessuno dei precedenti
- 

13. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
  - B. `Nodo<String> n = new Nodo<String>("");`
  - C. `new Nodo<Boolean>("");`
  - D. `new Nodo<Boolean>(true).getInfo().length();`
  - E. Nessuno dei precedenti
- 

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

15. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

---

16. Quale risultato produce questo codice?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}

```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti



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

**Prova n. 10**

1. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}
```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

2. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
- B. 2
- C. Errore a tempo di compilazione

- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

3. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

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

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;
- C. u = (Exception []) e;
- D. u = (Exception []) f;
- E. Nessuno dei precedenti

4. Qual è l'output di questo codice?

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

```

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

```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

5. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi `static` non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata `private`

6. Quale espressione va inserita nel punto indicato per stampare l'attributo `x` ereditato dalla classe `A`?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}

```

- A. `x`
- B. `A.this.x`
- C. `A.x`
- D. `super.x`
- E. Nessuna delle precedenti

7. Qual è l'output di questo codice?

```

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

```

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

8. Date le dichiarazioni:

```

Boolean e;
Exception m;
Object v;

```

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

- A. `m = v;`
- B. `e = (Boolean) m;`
- C. `e = v;`
- D. `m = (Exception) e;`
- E. `v = e;`

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



```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

10. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

11. Dato il template

```

class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}

```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
- B. `Nodo<String> n = new Nodo<String>("");`
- C. `new Nodo<Boolean>("");`
- D. `new Nodo<Boolean>(true).getInfo().length();`
- E. Nessuno dei precedenti

12. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

13. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
- B. 2

C. 3

D. 4

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 C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

A. 3452

B. 352Exception in thread "main" MyExc2

C. 35

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

15. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile `n` di tipo `Nodo` tale che `n.info=""`,  
dire quale dei loro usi riportati qui sotto è corretto:

A. `String s = n.getInfo();`

B. `int l = n.getInfo().length();`

C. `n=new Nodo(new Integer(0));`

D. `int v = n.getInfo().intValue();`

E. Nessuno dei precedenti

---

16. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

A. 12

B. 22

C. 3

D. 4

E. Nessuno dei precedenti

---



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

**Prova n. 11**

1. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

2. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

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

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;
- C. u = (Exception []) e;
- D. u = (Exception []) f;
- E. Nessuno dei precedenti

3. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
- B. `Nodo<String> n = new Nodo<String>("");`
- C. `new Nodo<Boolean>("");`
- D. `new Nodo<Boolean>(true).getInfo().length();`
- E. Nessuno dei precedenti

4. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

---

5. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi `static` non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata `private`
- 

6. Qual è l'output di questo codice?

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

- A. 42
  - B. 41
  - C. 4
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
```

```
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
  - B. 2
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

8. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
  - B. 22
  - C. 3
  - D. 4
  - E. Nessuno dei precedenti
- 

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
```

```

        throw( new MyExc2() );
    }
    catch( Exception e ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
        throw( new Exception() );
    }
}
}

```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

10. Date la classe

```

class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}

```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
- B. int l = n.getInfo().length();
- C. n=new Nodo(new Integer(0));
- D. int v = n.getInfo().intValue();
- E. Nessuno dei precedenti

11. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}

```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

12. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

13. Quale risultato produce questo codice?

```

class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}

```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

14. Date le dichiarazioni:

```
Boolean e;  
Exception m;  
Object v;
```

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

- A. `m = v;`
  - B. `e = (Boolean) m;`
  - C. `e = v;`
  - D. `m = (Exception) e;`
  - E. `v = e;`
- 

15. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {  
    String x = "abc";  
    Integer y;  
  
    C(Integer y) { this.y = y; }  
  
    public static void main(String a[]) {  
        C obj = new C( new Integer(1) );  
        obj = null; // QUESTA  
        System.out.println("done");  
    }  
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

16. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class C1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            n();  
        }  
        catch( MyExc3 j ) {  
            System.out.print(1);  
        }  
        finally {  
            System.out.print(2);  
        }  
    }  
    static void n() throws Exception {  
        try {  
            System.out.print(3);  
            if (true) throw( new MyExc2() );  
            else throw( new MyExc3() );  
        }  
        catch( MyExc1 g ) {  
            System.out.print(4);  
        }  
        catch( MyExc3 e ) {  
            throw( new MyExc1() );  
        }  
        finally {  
            System.out.print(5);  
        }  
    }  
}
```

- A. 3452
  - B. 352Exception in thread "main" MyExc2
  - C. 35
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-





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

**Prova n. 12**

1. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

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

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;
- C. u = (Exception []) e;
- D. u = (Exception []) f;
- E. Nessuno dei precedenti

2. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

3. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

4. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
- B. int l = n.getInfo().length();
- C. n=new Nodo(new Integer(0));
- D. int v = n.getInfo().intValue();
- E. Nessuno dei precedenti

5. Qual è il risultato di questo codice:

```

class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}

```

- A. 12
- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

6. Quale risultato produce questo codice?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}

```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

7. Quale risultato produce questo codice?

```

class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
}

```

```

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

```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

8. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi static non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata private

9. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

10. Dato il template

```

class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}

```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`

- B. `Nodo<String> n = new Nodo<String>("");`
- C. `new Nodo<Boolean>("");`
- D. `new Nodo<Boolean>(true).getInfo().length();`
- E. Nessuno dei precedenti

11. Qual è l'output di questo codice?

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

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
}
```

```
    }
}
static void p() {
    try {
        System.out.print(2);
        throw( new MyExc3() );
    }
    catch( MyExc2 k ) {
        throw( new MyExc2() );
    }
    catch( Exception e ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
        throw( new Exception() );
    }
}
}
```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.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 C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {

```

```

        System.out.print(5);
    }
}

```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

14. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 011

- C. 001
- D. 101
- E. 010

15. Date le dichiarazioni:

```

Boolean e;
Exception m;
Object v;

```

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

- A. m = v;
- B. e = (Boolean) m;
- C. e = v;
- D. m = (Exception) e;
- E. v = e;

16. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- 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. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

2. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi `static` non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata `private`

3. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

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

- A. `f = (Boolean []) e;`
- B. `f = (Boolean []) u;`
- C. `u = (Exception []) e;`
- D. `u = (Exception []) f;`
- E. Nessuno dei precedenti

4. Quale espressione va inserita nel punto indicato per stampare l'attributo `x` ereditato dalla classe `A`?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}

public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }

    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. `x`
- B. `A.this.x`
- C. `A.x`
- D. `super.x`
- E. Nessuna delle precedenti

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

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
```

```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s2 == s1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s4 == s3) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class Cl {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Date le dichiarazioni:

```

Boolean e;
Exception m;
Object v;

```

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

- A. m = v;
- B. e = (Boolean) m;
- C. e = v;
- D. m = (Exception) e;
- E. v = e;

8. Date la classe

```

class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}

```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
- B. int l = n.getInfo().length();
- C. n=new Nodo(new Integer(0));
- D. int v = n.getInfo().intValue();
- E. Nessuno dei precedenti

9. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

```

- A. x

- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}
```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

11. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

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

```
C obj = new C( new Integer(1) );
obj = null; // QUESTA
System.out.println("done");
}
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

12. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
- B. `Nodo<String> n = new Nodo<String>("");`
- C. `new Nodo<Boolean>("");`
- D. `new Nodo<Boolean>(true).getInfo().length();`
- E. Nessuno dei precedenti

13. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}
```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

14. Quale risultato produce questo codice?



```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}

```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

---

15. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
            System.out.print(3);
        }
    }
}

```

```

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

```

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

16. Qual è il risultato di questo codice:

```

class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}

```

- A. 12
  - B. 22
  - C. 3
  - D. 4
  - E. Nessuno dei precedenti
-



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

**Prova n. 14**

1. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

2. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
- B. `Nodo<String> n = new Nodo<String>("");`
- C. `new Nodo<Boolean>("");`
- D. `new Nodo<Boolean>(true).getInfo().length();`
- E. Nessuno dei precedenti

3. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

4. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

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

- A. `f = (Boolean []) e;`
- B. `f = (Boolean []) u;`
- C. `u = (Exception []) e;`
- D. `u = (Exception []) f;`
- E. Nessuno dei precedenti

5. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info="",  
dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
  - B. int l = n.getInfo().length();
  - C. n=new Nodo(new Integer(0));
  - D. int v = n.getInfo().intValue();
  - E. Nessuno dei precedenti
- 

6. Qual è l'output di questo codice?

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

- A. 42
  - B. 41
  - C. 4
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
    throws Exception {
        try {
```

```
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 3452
  - B. 352Exception in thread "main" MyExc2
  - C. 35
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

8. Qual è l'output di questo codice?

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

```

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

```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

9. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

10. Qual è il risultato di questo codice:

```

class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}

```

- A. 12
- B. 22
- C. 3

- D. 4
- E. Nessuno dei precedenti

11. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

12. Date le dichiarazioni:

```

Boolean e;
Exception m;
Object v;

```

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

- A. m = v;
- B. e = (Boolean) m;
- C. e = v;
- D. m = (Exception) e;
- E. v = e;

13. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi static non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata private

14. Quale risultato produce questo codice?

```

class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}

```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

15. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {

```

```

        System.out.print(0);
    }
    if(s4 == s3) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

16. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}

```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- 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 Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}
```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

2. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
```

```
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

3. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi **static** non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata *private*

4. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

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

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;
- C. u = (Exception []) e;



- D. u = (Exception []) f;
  - E. Nessuno dei precedenti
- 

5. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
  - B. 22
  - C. 3
  - D. 4
  - E. Nessuno dei precedenti
- 

6. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

7. Qual è l'output di questo codice?

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

```
throws Exception {
    try {
        m();
    }
    catch( Exception x ) {
        System.out.print(1);
    }
    catch( MyExc2 y ) {
        System.out.print(2);
    }
    catch( MyExc3 u ) {
        System.out.print(3);
    }
}
static void m() throws Exception {
    try {
        throw( new MyExc2() );
    }
    finally {
        System.out.print(4);
    }
}
```

- A. 42
  - B. 41
  - C. 4
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

8. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}
```

- A. 1
  - B. 2.0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

9. Dato il template

```

class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}

```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
- B. `Nodo<String> n = new Nodo<String>("");`
- C. `new Nodo<Boolean>("");`
- D. `new Nodo<Boolean>(true).getInfo().length();`
- E. Nessuno dei precedenti

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

11. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

12. Date la classe

```

class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}

```

e una variabile n di tipo Nodo tale che `n.info=""`, dire quale dei loro usi riportati qui sotto è corretto:

- A. `String s = n.getInfo();`
- B. `int l = n.getInfo().length();`
- C. `n=new Nodo(new Integer(0));`
- D. `int v = n.getInfo().intValue();`
- E. Nessuno dei precedenti

13. Date le dichiarazioni:

```
Boolean e;  
Exception m;  
Object v;
```

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

- A. m = v;
- B. e = (Boolean) m;
- C. e = v;
- D. m = (Exception) e;
- E. v = e;

---

14. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {  
    int x = 1;  
  
    public class B {  
        protected int x = 2;  
        public void my_print() {  
            System.out.println( _QUI_ );  
        }  
    }  
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

---

15. Quale risultato produce questo codice?

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}
```

```
public class B extends A {  
    protected int x;  
    public static void main(String a[]) {  
        B obj = new B();  
        obj.x = 1;  
        obj.set_x(2);  
        System.out.println( obj.x );  
    }  
}
```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

---

16. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {  
    String x = "abc";  
    static Integer y;  
  
    C(Integer y) { this.y = y; }  
  
    public static void main(String a[]) {  
        C obj = new C( new Integer(1) );  
        obj = null; // QUESTA  
        System.out.println("done");  
    }  
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
-



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

**Prova n. 16**

1. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi `static` non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata `private`

2. Quale espressione va inserita nel punto indicato per stampare l'attributo `x` dichiarato nella classe `A`?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}
```

- A. `x`
- B. `A.this.x`
- C. `A.x`
- D. `super.x`
- E. Nessuna delle precedenti

3. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

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

- A. `m = v;`
- B. `e = (Boolean) m;`
- C. `e = v;`
- D. `m = (Exception) e;`

E. `v = e;`

4. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

5. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}
```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
```

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

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione

E. Nessuna delle precedenti

9. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

10. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1

B. 2

C. 3

D. 4

E. Nessuna delle precedenti

11. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
- B. `Nodo<String> n = new Nodo<String>("");`
- C. `new Nodo<Boolean>("");`
- D. `new Nodo<Boolean>(true).getInfo().length();`
- E. Nessuno dei precedenti

12. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

13. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info="",  
dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
  - B. int l = n.getInfo().length();
  - C. n=new Nodo(new Integer(0));
  - D. int v = n.getInfo().intValue();
  - E. Nessuno dei precedenti
- 

14. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

15. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

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

- A. f = (Boolean []) e;
  - B. f = (Boolean []) u;
  - C. u = (Exception []) e;
  - D. u = (Exception []) f;
  - E. Nessuno dei precedenti
- 

16. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}
```

- A. 1
  - B. 2.0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
-





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

**Prova n. 17**

1. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

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

- A. `m = v;`
- B. `e = (Boolean) m;`
- C. `e = v;`
- D. `m = (Exception) e;`
- E. `v = e;`

```
C(Integer y) { this.y = y; }

public static void main(String a[]) {
    C obj = new C( new Integer(1) );
    obj = null; // QUESTA
    System.out.println("done");
}
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

2. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

3. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;
```

4. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
- B. `Nodo<String> n = new Nodo<String>("");`
- C. `new Nodo<Boolean>("");`
- D. `new Nodo<Boolean>(true).getInfo().length();`
- E. Nessuno dei precedenti

5. Qual è l'output di questo codice?

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

```

    catch( MyExc2 y ) {
        System.out.print(2);
    }
    catch( MyExc3 u ) {
        System.out.print(3);
    }
}
static void m() throws Exception {
    try {
        throw( new MyExc2() );
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

6. Date la classe

```

class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}

```

e una variabile n di tipo Nodo tale che n.info="",  
dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
- B. int l = n.getInfo().length();
- C. n=new Nodo(new Integer(0));
- D. int v = n.getInfo().intValue();
- E. Nessuno dei precedenti

7. Qual è l'output di questo codice?

```

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

```

```

    finally {
        System.out.print(2);
    }
}
static void n() throws Exception {
    try {
        System.out.print(3);
        if (true) throw( new MyExc2() );
        else throw( new MyExc3() );
    }
    catch( MyExc1 g ) {
        System.out.print(4);
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(5);
    }
}
}

```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

8. Date le dichiarazioni:

```

Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];

```

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

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;
- C. u = (Exception []) e;
- D. u = (Exception []) f;
- E. Nessuno dei precedenti

9. Quale risultato produce questo codice?

```

class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {

```

```

        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}

```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

10. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}

```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

11. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        }
    }
}

```

```

    } else {
        System.out.print(0);
    }
    if(s2 == s1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s4 == s3) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

12. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;
}
public class B {
    protected int x = 2;
    public void my_print() {
        System.out.println( _QUI_ );
    }
}
}

```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {

```

```

try {
    System.out.print(2);
    throw( new MyExc3() );
}
catch( MyExc2 k ) {
    throw( new MyExc2() );
}
catch( Exception e ) {
    System.out.print(3);
}
finally {
    System.out.print(4);
    throw( new Exception() );
}
}
}

```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

14. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1

- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

15. Qual è il risultato di questo codice:

```

class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}

```

- A. 12
- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

16. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi `static` non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata `private`



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. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

2. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
- B. int l = n.getInfo().length();
- C. n=new Nodo(new Integer(0));
- D. int v = n.getInfo().intValue();
- E. Nessuno dei precedenti

3. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
    }
}
```

```
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}
```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

4. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

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

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;
- C. u = (Exception []) e;
- D. u = (Exception []) f;
- E. Nessuno dei precedenti

5. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

6. Qual è il risultato di questo codice:

```

class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}

```

- A. 12
- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

7. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi `static` non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata `private`

8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```

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

```



```

        throw( new MyExc3() );
    }
    catch( MyExc2 k ) {
        throw( new MyExc2() );
    }
    catch( Exception e ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
        throw( new Exception() );
    }
}

```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```

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

```

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione

E. Nessuna delle precedenti

11. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

12. Date le dichiarazioni:

```

Boolean e;
Exception m;
Object v;

```

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

- A. m = v;
- B. e = (Boolean) m;
- C. e = v;
- D. m = (Exception) e;
- E. v = e;

13. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}

```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

---

14. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

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

E. Nessuna delle precedenti

---

15. Dato il template

```

class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}

```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
- B. `Nodo<String> n = new Nodo<String>("");`
- C. `new Nodo<Boolean>("");`
- D. `new Nodo<Boolean>(true).getInfo().length();`
- E. Nessuno dei precedenti

---

16. Quale risultato produce questo codice?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}

```

- A. 1
  - B. 2
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
-



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

**Prova n. 19**

1. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

2. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi **static** non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata **private**

3. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
}
```

```
public static void main(String a[]) {
    B obj = new B();
    obj.x = 1;
    obj.set_x(2);
    obj.my_print();
}
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

4. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}
```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

6. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

```

- A. x
- B. A.this.x
- C. A.x

- D. super.x
- E. Nessuna delle precedenti

7. Date la classe

```

class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}

```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
- B. int l = n.getInfo().length();
- C. n=new Nodo(new Integer(0));
- D. int v = n.getInfo().intValue();
- E. Nessuno dei precedenti

8. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

9. Date le dichiarazioni:

```

Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];

```

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

- A. f = (Boolean []) e;

- B. f = (Boolean []) u;
- C. u = (Exception []) e;
- D. u = (Exception []) f;
- E. Nessuno dei precedenti

- C. 001
- D. 101
- E. 010

---

10. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
- B. `Nodo<String> n = new Nodo<String>("");`
- C. `new Nodo<Boolean>("");`
- D. `new Nodo<Boolean>(true).getInfo().length();`
- E. Nessuno dei precedenti

---

11. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 011

---

12. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

---

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}
```

- A. 23431
  - B. Errore a tempo di compilazione
  - C. 234Exception in thread "main" java.lang.Exception
  - D. 24Exception in thread "main" java.lang.Exception
  - E. Nessuna delle precedenti
- 

14. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
  - B. 2
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

15. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

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

- A. m = v;
- B. e = (Boolean) m;

- C. e = v;
  - D. m = (Exception) e;
  - E. v = e;
- 

16. Qual è l'output di questo codice?

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

- A. 42
  - B. 41
  - C. 4
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-





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

**Prova n. 20**

1. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

2. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile *n* di tipo *Nodo* tale che *n.info=""*, dire quale dei loro usi riportati qui sotto è corretto:

- A. `String s = n.getInfo();`
- B. `int l = n.getInfo().length();`
- C. `n=new Nodo(new Integer(0));`
- D. `int v = n.getInfo().intValue();`
- E. Nessuno dei precedenti

3. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

4. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
- B. `Nodo<String> n = new Nodo<String>("");`
- C. `new Nodo<Boolean>("");`
- D. `new Nodo<Boolean>(true).getInfo().length();`
- E. Nessuno dei precedenti

5. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi `static` non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata `private`

6. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

7. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

8. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
```

```
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}
```

- A. 1
  - B. 2.0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

9. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

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

- A. m = v;
  - B. e = (Boolean) m;
  - C. e = v;
  - D. m = (Exception) e;
  - E. v = e;
- 

10. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}
```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}

```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
            System.out.print(3);
        }
    }
}

```

```

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

```

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

13. Quale risultato produce questo codice?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}

```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

14. Qual è l'output di questo codice?

```

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

```

```

static void n() throws Exception {
    try {
        System.out.print(3);
        if (true) throw( new MyExc2() );
        else throw( new MyExc3() );
    }
    catch( MyExc1 g ) {
        System.out.print(4);
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(5);
    }
}
}

```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

15. Date le dichiarazioni:

```

Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];

```

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

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;
- C. u = (Exception []) e;
- D. u = (Exception []) f;

E. Nessuno dei precedenti

16. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010



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

**Prova n. 21**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}
```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

2. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
```

```
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

3. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110

- B. 011
  - C. 001
  - D. 101
  - E. 010
- 

4. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

5. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
  - B. 22
  - C. 3
  - D. 4
  - E. Nessuno dei precedenti
- 

6. Qual è l'output di questo codice?

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

- A. 42
  - B. 41
  - C. 4
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}
```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione

- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

8. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
- B. `Nodo<String> n = new Nodo<String>("");`
- C. `new Nodo<Boolean>("");`
- D. `new Nodo<Boolean>(true).getInfo().length();`
- E. Nessuno dei precedenti

---

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

---

10. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi static non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata private

---

11. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

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

- A. `m = v;`
- B. `e = (Boolean) m;`
- C. `e = v;`
- D. `m = (Exception) e;`
- E. `v = e;`

---

12. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}
```

- A. `x`
- B. `A.this.x`
- C. `A.x`
- D. `super.x`
- E. Nessuna delle precedenti

---

13. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?



```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

14. Date la classe

```

class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}

```

e una variabile `n` di tipo `Nodo` tale che `n.info=""`,  
dire quale dei loro usi riportati qui sotto è corretto:

- A. `String s = n.getInfo();`
  - B. `int l = n.getInfo().length();`
  - C. `n=new Nodo(new Integer(0));`
  - D. `int v = n.getInfo().intValue();`
  - E. Nessuno dei precedenti
- 

15. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

16. Date le dichiarazioni:

```

Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];

```

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

- A. `f = (Boolean []) e;`
  - B. `f = (Boolean []) u;`
  - C. `u = (Exception []) e;`
  - D. `u = (Exception []) f;`
  - E. Nessuno dei precedenti
-



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

**Prova n. 22**

1. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

2. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }
```

```
public static void main(String a[]) {
    C obj = new C( new Integer(1) );
    obj = null; // QUESTA
    System.out.println("done");
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}
```

- A. 23431
- B. Errore a tempo di compilazione

- C. 234Exception in thread "main" java.lang.Exception
  - D. 24Exception in thread "main" java.lang.Exception
  - E. Nessuna delle precedenti
- 

4. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info="",  
dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
  - B. int l = n.getInfo().length();
  - C. n=new Nodo(new Integer(0));
  - D. int v = n.getInfo().intValue();
  - E. Nessuno dei precedenti
- 

5. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

6. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");
  - B. Nodo<String> n = new Nodo<String>("");
  - C. new Nodo<Boolean>("");
  - D. new Nodo<Boolean>(true).getInfo().length();
  - E. Nessuno dei precedenti
- 

7. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}
```

- A. 1
  - B. 2.0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

8. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x
- B. A.this.x
- C. A.x

- D. super.x
- E. Nessuna delle precedenti

---

9. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi `static` non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata `private`

---

10. Date le dichiarazioni:

```
Object [] e;  
Boolean [] f;  
Exception [] u;  
u = new Exception [8];  
e = new Exception [1];  
f = new Boolean [5];
```

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

- A. `f = (Boolean []) e;`
- B. `f = (Boolean []) u;`
- C. `u = (Exception []) e;`
- D. `u = (Exception []) f;`
- E. Nessuno dei precedenti

---

11. Quale risultato produce questo codice?

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}  
public class B extends A {  
    protected int x;  
    public static void main(String a[]) {  
        B obj = new B();  
        obj.x = 1;  
        obj.set_x(2);  
        System.out.println( obj.x );  
    }  
}
```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

---

12. Date le dichiarazioni:

```
Boolean e;  
Exception m;  
Object v;
```

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

- A. `m = v;`
- B. `e = (Boolean) m;`
- C. `e = v;`
- D. `m = (Exception) e;`
- E. `v = e;`

---

13. Qual è l'output di questo codice?

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

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452
  - B. 352Exception in thread "main" MyExc2
  - C. 35
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

15. Qual è il risultato di questo codice:

```

class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}

```

- A. 12
  - B. 22
  - C. 3
  - D. 4
  - E. Nessuno dei precedenti
- 

16. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
-



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

**Prova n. 23**

1. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

2. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x
- B. A.this.x
- C. A.x

D. super.x

E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. Qual è il risultato di questo codice:



```

class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}

```

- A. 12
- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

5. Quale risultato produce questo codice?

```

class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}

```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

6. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {

```

```

String s4;
s3 = "";
s4 = "";
if(b1 == b2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s2 == s1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s4 == s3) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
}

```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

7. Date le dichiarazioni:

```

Boolean e;
Exception m;
Object v;

```

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

- A. m = v;
- B. e = (Boolean) m;
- C. e = v;
- D. m = (Exception) e;
- E. v = e;

8. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

---

9. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

---

10. Qual è l'output di questo codice?

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

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

11. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

---

12. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

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

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;
- C. u = (Exception []) e;
- D. u = (Exception []) f;
- E. Nessuno dei precedenti

---

13. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`;
  - B. `Nodo<String> n = new Nodo<String>("");`;
  - C. `new Nodo<Boolean>("");`;
  - D. `new Nodo<Boolean>(true).getInfo().length();`;
  - E. Nessuno dei precedenti
- 

14. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi `static` non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata `private`
- 

15. Qual è l'output di questo codice?

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

```
catch( MyExc2 k ) {
    throw( new MyExc2() );
}
catch( Exception e ) {
    System.out.print(3);
}
finally {
    System.out.print(4);
    throw( new Exception() );
}
}
}
```

- A. 23431
  - B. Errore a tempo di compilazione
  - C. 234Exception in thread "main" java.lang.Exception
  - D. 24Exception in thread "main" java.lang.Exception
  - E. Nessuna delle precedenti
- 

16. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile `n` di tipo `Nodo` tale che `n.info=""`, dire quale dei loro usi riportati qui sotto è corretto:

- A. `String s = n.getInfo();`
  - B. `int l = n.getInfo().length();`
  - C. `n=new Nodo(new Integer(0));`
  - D. `int v = n.getInfo().intValue();`
  - E. Nessuno dei precedenti
-



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

**Prova n. 24**

1. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

2. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

3. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

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

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;
- C. u = (Exception []) e;
- D. u = (Exception []) f;
- E. Nessuno dei precedenti

4. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}
```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

5. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}

```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```

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

```

- A. 42
- B. 41
- C. 4

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

7. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

8. Quale risultato produce questo codice?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}

```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

9. Dato il template

```

class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}

```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
- B. `Nodo<String> n = new Nodo<String>("");`
- C. `new Nodo<Boolean>("");`
- D. `new Nodo<Boolean>(true).getInfo().length();`
- E. Nessuno dei precedenti

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

10. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile `n` di tipo `Nodo` tale che `n.info=""`,  
dire quale dei loro usi riportati qui sotto è corretto:

- A. `String s = n.getInfo();`
- B. `int l = n.getInfo().length();`
- C. `n=new Nodo(new Integer(0));`
- D. `int v = n.getInfo().intValue();`
- E. Nessuno dei precedenti

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

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}
```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

13. Quale espressione va inserita nel punto indicato per stampare l'attributo `x` dichiarato nella classe `A`?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
```

```

        System.out.println( _QUI_ );
    }
}

```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

14. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi `static` non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata `private`
- 

15. Qual è l'output di questo codice?

```

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

```

```

static void n() throws Exception {
    try {
        System.out.print(3);
        if (true) throw( new MyExc2() );
        else throw( new MyExc3() );
    }
    catch( MyExc1 g ) {
        System.out.print(4);
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(5);
    }
}

```

- A. 3452
  - B. 352Exception in thread "main" MyExc2
  - C. 35
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Date le dichiarazioni:

```

Boolean e;
Exception m;
Object v;

```

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

- A. `m = v;`
  - B. `e = (Boolean) m;`
  - C. `e = v;`
  - D. `m = (Exception) e;`
  - E. `v = e;`
-





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

**Prova n. 25**

1. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

2. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x

E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}
```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

4. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

5. Date le dichiarazioni:

```

Boolean e;
Exception m;
Object v;

```

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

- A. m = v;
- B. e = (Boolean) m;
- C. e = v;
- D. m = (Exception) e;
- E. v = e;

6. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

7. Quale risultato produce questo codice?

```

class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}

public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}

```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

8. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

```

```
    }  
  }  
}
```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

9. Date le dichiarazioni:

```
Object [] e;  
Boolean [] f;  
Exception [] u;  
u = new Exception [8];  
e = new Exception [1];  
f = new Boolean [5];
```

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

- A. f = (Boolean []) e;
  - B. f = (Boolean []) u;
  - C. u = (Exception []) e;
  - D. u = (Exception []) f;
  - E. Nessuno dei precedenti
- 

10. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {  
    String x = "abc";  
    static Integer y;  
  
    C(Integer y) { this.y = y; }  
  
    public static void main(String a[]) {  
        C obj = new C( new Integer(1) );  
        obj = null; // QUESTA  
        System.out.println("done");  
    }  
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

11. Date la classe

```
class Nodo {  
    private Object info;  
    Nodo next;  
    Nodo(Object i) { info=i; }  
    public Object getInfo() { return info; }  
}
```

è una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
  - B. int l = n.getInfo().length();
  - C. n=new Nodo(new Integer(0));
  - D. int v = n.getInfo().intValue();
  - E. Nessuno dei precedenti
- 

12. Quale risultato produce questo codice?

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}  
public class B extends A {  
    protected int x;  
    public static void main(String a[]) {  
        B obj = new B();  
        obj.x = 1;  
        obj.set_x(2);  
        System.out.println( obj.x );  
    }  
}
```

- A. 1
  - B. 2
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

13. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi static non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata private
- 

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

15. Dato il template

```

class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}

```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
- B. `Nodo<String> n = new Nodo<String>("");`
- C. `new Nodo<Boolean>("");`
- D. `new Nodo<Boolean>(true).getInfo().length();`
- E. Nessuno dei precedenti

16. Qual è l'output di questo codice?

```

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

```

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti



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

**Prova n. 26**

1. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
```

```
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

3. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
```

```

        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}

```

- A. 1
  - B. 2.0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

4. Dato il template

```

class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}

```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
  - B. `Nodo<String> n = new Nodo<String>("");`
  - C. `new Nodo<Boolean>("");`
  - D. `new Nodo<Boolean>(true).getInfo().length();`
  - E. Nessuno dei precedenti
- 

5. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

6. Qual è il risultato di questo codice:

```

class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}

```

- A. 12
  - B. 22
  - C. 3
  - D. 4
  - E. Nessuno dei precedenti
- 

7. Qual è l'output di questo codice?

```

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

```

- A. 42
  - B. 41
  - C. 4
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}
```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

9. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
- B. 2

- C. 3
- D. 4
- E. Nessuna delle precedenti

10. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

11. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

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

- A. m = v;
- B. e = (Boolean) m;
- C. e = v;
- D. m = (Exception) e;
- E. v = e;

12. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

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

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;

- C. u = (Exception []) e;
  - D. u = (Exception []) f;
  - E. Nessuno dei precedenti
- 

13. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info="",  
dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
  - B. int l = n.getInfo().length();
  - C. n=new Nodo(new Integer(0));
  - D. int v = n.getInfo().intValue();
  - E. Nessuno dei precedenti
- 

14. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

15. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}
```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

16. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi static non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata private
-



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

**Prova n. 27**

1. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x  
 B. A.this.x  
 C. A.x  
 D. super.x  
 E. Nessuna delle precedenti

2. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

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

- A. f = (Boolean []) e;  
 B. f = (Boolean []) u;  
 C. u = (Exception []) e;  
 D. u = (Exception []) f;  
 E. Nessuno dei precedenti

3. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();  
 B. int l = n.getInfo().length();  
 C. n=new Nodo(new Integer(0));  
 D. int v = n.getInfo().intValue();  
 E. Nessuno dei precedenti

4. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null  
 B. Un oggetto può essere inizializzato dopo la sua dichiarazione  
 C. Un array possiede dei membri  
 D. Ai metodi static non si applica il *dynamic method dispatch*  
 E. Una classe interna può essere dichiarata private

5. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
```

```

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

```

- A. 42
  - B. 41
  - C. 4
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

6. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

7. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452
  - B. 352Exception in thread "main" MyExc2
  - C. 35
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

9. Quale risultato produce questo codice?

```

class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}

```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

10. Quale risultato produce questo codice?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}

```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();

```

```

        System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}

```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

12. Qual è il risultato di questo codice:

```

class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}

```

- A. 12
- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

13. Date le dichiarazioni:

```

Boolean e;
Exception m;
Object v;

```

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

- A. `m = v;`
- B. `e = (Boolean) m;`
- C. `e = v;`
- D. `m = (Exception) e;`
- E. `v = e;`

---

14. Quale espressione va inserita nel punto indicato per stampare l'attributo `x` dichiarato nella classe `A`?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}
```

- A. `x`
- B. `A.this.x`
- C. `A.x`
- D. `super.x`
- E. Nessuna delle precedenti

---

15. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`

- B. `Nodo<String> n = new Nodo<String>("");`
- C. `new Nodo<Boolean>("");`
- D. `new Nodo<Boolean>(true).getInfo().length();`
- E. Nessuno dei precedenti

---

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

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
  - B. 011
  - C. 001
  - D. 101
  - E. 010
-





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

**Prova n. 28**

1. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

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

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;
- C. u = (Exception []) e;
- D. u = (Exception []) f;
- E. Nessuno dei precedenti

2. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

3. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

4. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

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

- A. m = v;
- B. e = (Boolean) m;
- C. e = v;
- D. m = (Exception) e;
- E. v = e;

5. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. `String s = n.getInfo();`
- B. `int l = n.getInfo().length();`
- C. `n=new Nodo(new Integer(0));`
- D. `int v = n.getInfo().intValue();`
- E. Nessuno dei precedenti

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;
```

```
C(Integer y) { this.y = y; }

public static void main(String a[]) {
    C obj = new C( new Integer(1) );
    obj = null; // QUESTA
    System.out.println("done");
}
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

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

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

9. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

10. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

11. Quale risultato produce questo codice?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}

```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

12. Quale risultato produce questo codice?

```

class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}

```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

13. Dato il template

```

class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}

```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
  - B. `Nodo<String> n = new Nodo<String>("");`
  - C. `new Nodo<Boolean>("");`
  - D. `new Nodo<Boolean>(true).getInfo().length();`
  - E. Nessuno dei precedenti
- 

14. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi `static` non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata `private`
- 

15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
```

```
            throw( new MyExc2() );
        }
    }
    catch( Exception e ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
        throw( new Exception() );
    }
}
```

- A. 23431
  - B. Errore a tempo di compilazione
  - C. `234Exception in thread "main" java.lang.Exception`
  - D. `24Exception in thread "main" java.lang.Exception`
  - E. Nessuna delle precedenti
- 

16. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
  - B. 22
  - C. 3
  - D. 4
  - E. Nessuno dei precedenti
-



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

**Prova n. 29**

1. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile `n` di tipo `Nodo` tale che `n.info=""`,  
dire quale dei loro usi riportati qui sotto è corretto:

- A. `String s = n.getInfo();`
- B. `int l = n.getInfo().length();`
- C. `n=new Nodo(new Integer(0));`
- D. `int v = n.getInfo().intValue();`
- E. Nessuno dei precedenti

2. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

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

- A. `m = v;`
- B. `e = (Boolean) m;`
- C. `e = v;`
- D. `m = (Exception) e;`
- E. `v = e;`

3. Qual è l'output di questo codice?

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

```
catch( Exception x ) {
    System.out.print(1);
}
catch( MyExc2 y ) {
    System.out.print(2);
}
catch( MyExc3 u ) {
    System.out.print(3);
}
}
static void m() throws Exception {
    try {
        throw( new MyExc2() );
    }
    finally {
        System.out.print(4);
    }
}
}
```

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

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

```

        System.out.print(3);
        if (true) throw( new MyExc2() );
        else throw( new MyExc3() );
    }
    catch( MyExc1 g ) {
        System.out.print(4);
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(5);
    }
}

```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

6. Dato il template

```

class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}

```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`

- B. `Nodo<String> n = new Nodo<String>("");`
- C. `new Nodo<Boolean>("");`
- D. `new Nodo<Boolean>(true).getInfo().length();`
- E. Nessuno dei precedenti

7. Date le dichiarazioni:

```

Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];

```

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

- A. `f = (Boolean []) e;`
- B. `f = (Boolean []) u;`
- C. `u = (Exception []) e;`
- D. `u = (Exception []) f;`
- E. Nessuno dei precedenti

8. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}

```

- A. `x`
- B. `A.this.x`
- C. `A.x`
- D. `super.x`
- E. Nessuna delle precedenti

9. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

10. Qual è il risultato di questo codice:

```

class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}

```

- A. 12
- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

11. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

```

- A. x

- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

12. Quale risultato produce questo codice?

```

class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}

```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

13. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi **static** non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata *private*

14. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {

```



```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s2 == s1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s4 == s3) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

15. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {

```

```

            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}

```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

16. Quale risultato produce questo codice?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}

```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti



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

**Prova n. 30**

1. Qual è l'output di questo codice?

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

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }
```

```
public static void main(String a[]) {
    C obj = new C( new Integer(1) );
    obj = null; // QUESTA
    System.out.println("done");
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

3. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

4. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi `static` non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata `private`

---

5. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile `n` di tipo `Nodo` tale che `n.info=""`,  
dire quale dei loro usi riportati qui sotto è corretto:

- A. `String s = n.getInfo();`
  - B. `int l = n.getInfo().length();`
  - C. `n=new Nodo(new Integer(0));`
  - D. `int v = n.getInfo().intValue();`
  - E. Nessuno dei precedenti
- 

6. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
  - B. `Nodo<String> n = new Nodo<String>("");`
  - C. `new Nodo<Boolean>("");`
  - D. `new Nodo<Boolean>(true).getInfo().length();`
  - E. Nessuno dei precedenti
- 

7. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}
```

- A. 1
  - B. 2.0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

8. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

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

- A. `f = (Boolean []) e;`
  - B. `f = (Boolean []) u;`
  - C. `u = (Exception []) e;`
  - D. `u = (Exception []) f;`
  - E. Nessuno dei precedenti
- 

9. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
  - B. 2
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

10. Quale espressione va inserita nel punto indicato per  
stampare l'attributo `x` dichiarato nella classe `A`?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- 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 C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione

E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}

```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

13. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {

```

```

        System.out.print(0);
    }
    if(s2 == s1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s4 == s3) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

---

14. Date le dichiarazioni:

```

Boolean e;
Exception m;
Object v;

```

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

- A. m = v;
- B. e = (Boolean) m;
- C. e = v;
- D. m = (Exception) e;
- E. v = e;

---

15. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}

```

```

}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}

```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

---

16. Qual è il risultato di questo codice:

```

class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}

```

- A. 12
  - B. 22
  - C. 3
  - D. 4
  - E. Nessuno dei precedenti
-



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

**Prova n. 31**

1. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

- B. 011
- C. 001
- D. 101
- E. 010

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

- A. `m = v;`
- B. `e = (Boolean) m;`
- C. `e = v;`
- D. `m = (Exception) e;`
- E. `v = e;`

2. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110

3. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

4. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

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

- A. `f = (Boolean []) e;`
- B. `f = (Boolean []) u;`
- C. `u = (Exception []) e;`
- D. `u = (Exception []) f;`
- E. Nessuno dei precedenti



---

5. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

---

6. Qual è l'output di questo codice?

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

- A. 42
- B. 41
- C. 4

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

7. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
- B. `Nodo<String> n = new Nodo<String>("");`
- C. `new Nodo<Boolean>("");`
- D. `new Nodo<Boolean>(true).getInfo().length();`
- E. Nessuno dei precedenti

---

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}
```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception

D. 24Exception in thread "main" java.lang.Exception

E. Nessuna delle precedenti

---

9. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}
```

A. 1

B. 2.0

C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuno dei precedenti

---

10. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

A. 1

B. 2

C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuno dei precedenti

---

11. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

A. x

B. A.this.x

C. A.x

D. super.x

E. Nessuna delle precedenti

---

12. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}
```

A. x

B. A.this.x

C. A.x

D. super.x

E. Nessuna delle precedenti

---

13. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

---

14. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

15. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile `n` di tipo `Nodo` tale che `n.info=""`,  
dire quale dei loro usi riportati qui sotto è corretto:

- A. `String s = n.getInfo();`
- B. `int l = n.getInfo().length();`
- C. `n=new Nodo(new Integer(0));`
- D. `int v = n.getInfo().intValue();`
- E. Nessuno dei precedenti

---

16. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi `static` non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata `private`
-



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

**Prova n. 32**

1. Qual è l'output di questo codice?

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

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
- B. int l = n.getInfo().length();
- C. n=new Nodo(new Integer(0));
- D. int v = n.getInfo().intValue();
- E. Nessuno dei precedenti

3. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

4. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi static non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata private

---

5. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

---

6. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

---

7. Qual è l'output di questo codice?

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

```
catch( MyExc3 h ) {
    throw( new MyExc2() );
}
}
static void p() {
    try {
        System.out.print(2);
        throw( new MyExc3() );
    }
    catch( MyExc2 k ) {
        throw( new MyExc2() );
    }
    catch( Exception e ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
        throw( new Exception() );
    }
}
}
```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

---

8. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

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

- A. m = v;
- B. e = (Boolean) m;
- C. e = v;
- D. m = (Exception) e;
- E. v = e;

---

9. Qual è l'output di questo codice?

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

```

    }
    finally {
        System.out.print(2);
    }
}
static void n() throws Exception {
    try {
        System.out.print(3);
        if (true) throw( new MyExc2() );
        else throw( new MyExc3() );
    }
    catch( MyExc1 g ) {
        System.out.print(4);
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(5);
    }
}
}

```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

10. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

11. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

12. Date le dichiarazioni:

```

Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];

```

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

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;
- C. u = (Exception []) e;
- D. u = (Exception []) f;
- E. Nessuno dei precedenti

13. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

14. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
- B. `Nodo<String> n = new Nodo<String>("");`
- C. `new Nodo<Boolean>("");`
- D. `new Nodo<Boolean>(true).getInfo().length();`
- E. Nessuno dei precedenti

15. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
```

```
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

16. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}
```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti





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

**Prova n. 33**

1. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}
```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

2. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
- B. 2
- C. 3

D. 4

E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}
```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

4. Date le dichiarazioni:

```

Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];

```

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

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;
- C. u = (Exception []) e;
- D. u = (Exception []) f;
- E. Nessuno dei precedenti

#### 5. Dato il template

```

class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}

```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");
- B. Nodo<String> n = new Nodo<String>("");
- C. new Nodo<Boolean>("");
- D. new Nodo<Boolean>(true).getInfo().length();
- E. Nessuno dei precedenti

#### 6. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

#### 7. Date la classe

```

class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}

```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
- B. int l = n.getInfo().length();
- C. n=new Nodo(new Integer(0));
- D. int v = n.getInfo().intValue();
- E. Nessuno dei precedenti

#### 8. Qual è l'output di questo codice?

```

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

```

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

#### 9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

10. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi `static` non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata `private`

11. Quale espressione va inserita nel punto indicato per stampare l'attributo `x` ereditato dalla classe `A`?

```

class A {
    protected int x;

```

```

        public void set_x(int v) { x=v; }
    }
    public class B extends A {
        protected int x;
        public void my_print() {
            System.out.println( _QUI_ );
        }
        public static void main(String a[]) {
            B obj = new B();
            obj.x = 1;
            obj.set_x(2);
            obj.my_print();
        }
    }

```

- A. `x`
- B. `A.this.x`
- C. `A.x`
- D. `super.x`
- E. Nessuna delle precedenti

12. Quale risultato produce questo codice?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}

```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

13. Date le dichiarazioni:

```

Boolean e;
Exception m;
Object v;

```

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

- A. `m = v;`
- B. `e = (Boolean) m;`
- C. `e = v;`
- D. `m = (Exception) e;`
- E. `v = e;`

---

14. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

---

15. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
- B. 2
- C. 3

D. 4

E. Nessuna delle precedenti

---

16. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
  - B. 011
  - C. 001
  - D. 101
  - E. 010
-



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

**Prova n. 34**

1. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
- B. `Nodo<String> n = new Nodo<String>("");`
- C. `new Nodo<Boolean>("");`
- D. `new Nodo<Boolean>(true).getInfo().length();`
- E. Nessuno dei precedenti

4. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

2. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

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

- A. `m = v;`
- B. `e = (Boolean) m;`
- C. `e = v;`
- D. `m = (Exception) e;`
- E. `v = e;`

3. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}
```

5. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12

- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

6. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

7. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi **static** non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata **private**

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
```

```
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

9. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}
```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

10. Qual è l'output di questo codice?



```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}

```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

11. Date la classe

```

class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}

```

e una variabile n di tipo Nodo tale che n.info="",  
dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
- B. int l = n.getInfo().length();
- C. n=new Nodo(new Integer(0));
- D. int v = n.getInfo().intValue();
- E. Nessuno dei precedenti

12. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

13. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

14. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
    }
}

```

```

if(b1 == b2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s2 == s1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s4 == s3) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
}

```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

15. Qual è l'output di questo codice?

```

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

```

```

catch( MyExc3 u ) {
    System.out.print(3);
}
}
static void m() throws Exception {
    try {
        throw( new MyExc2() );
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

16. Date le dichiarazioni:

```

Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];

```

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

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;
- C. u = (Exception []) e;
- D. u = (Exception []) f;
- E. Nessuno dei precedenti



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

**Prova n. 35**

1. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

2. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
```

```
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

3. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

4. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info="",  
dire quale dei loro usi riportati qui sotto è corretto:

- A. `String s = n.getInfo();`
  - B. `int l = n.getInfo().length();`
  - C. `n=new Nodo(new Integer(0));`
  - D. `int v = n.getInfo().intValue();`
  - E. Nessuno dei precedenti
- 

5. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi `static` non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata `private`
- 

6. Quale espressione va inserita nel punto indicato per stampare l'attributo `x` dichiarato nella classe `A`?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}
```

- A. `x`
  - B. `A.this.x`
  - C. `A.x`
  - D. `super.x`
  - E. Nessuna delle precedenti
- 

7. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1

- B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

8. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

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

- A. `m = v;`
  - B. `e = (Boolean) m;`
  - C. `e = v;`
  - D. `m = (Exception) e;`
  - E. `v = e;`
- 

9. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
  - B. `Nodo<String> n = new Nodo<String>("");`
  - C. `new Nodo<Boolean>("");`
  - D. `new Nodo<Boolean>(true).getInfo().length();`
  - E. Nessuno dei precedenti
- 

10. Quale espressione va inserita nel punto indicato per stampare l'attributo `x` ereditato dalla classe `A`?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}
```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

12. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }
```

```
public static void main(String a[]) {
    C obj = new C( new Integer(1) );
    obj = null; // QUESTA
    System.out.println("done");
}
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

13. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}
```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

14. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
            System.out.print(3);
        }
    }
}
```

```

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

```

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

15. Qual è l'output di questo codice?

```

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

```

```

        else throw( new MyExc3() );
    }
    catch( MyExc1 g ) {
        System.out.print(4);
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(5);
    }
}
}

```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

16. Date le dichiarazioni:

```

Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];

```

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

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;
- C. u = (Exception []) e;
- D. u = (Exception []) f;
- E. Nessuno dei precedenti





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

**Prova n. 36**

1. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}
```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

2. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi `static` non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata `private`

3. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

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

- A. `m = v;`
- B. `e = (Boolean) m;`
- C. `e = v;`
- D. `m = (Exception) e;`
- E. `v = e;`

4. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

5. Quale espressione va inserita nel punto indicato per stampare l'attributo `x` ereditato dalla classe `A`?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
```

```

        obj.set_x(2);
        obj.my_print();
    }
}

```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

6. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

7. Dato il template

```

class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}

```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
- B. `Nodo<String> n = new Nodo<String>("");`
- C. `new Nodo<Boolean>("");`
- D. `new Nodo<Boolean>(true).getInfo().length();`
- E. Nessuno dei precedenti

8. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
    }
}

```

```

    catch( Exception e ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
        throw( new Exception() );
    }
}
}

```

A. 23431  
 B. Errore a tempo di compilazione  
 C. 234Exception in thread "main" java.lang.Exception  
 D. 24Exception in thread "main" java.lang.Exception  
 E. Nessuna delle precedenti

10. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1  
 B. 2  
 C. 3  
 D. 4  
 E. Nessuna delle precedenti

11. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

```

- A. x  
 B. A.this.x

- C. A.x  
 D. super.x  
 E. Nessuna delle precedenti

12. Qual è il risultato di questo codice:

```

class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}

```

- A. 12  
 B. 22  
 C. 3  
 D. 4  
 E. Nessuno dei precedenti

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {

```

```

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

```

- A. 3452
  - B. 352Exception in thread "main" MyExc2
  - C. 35
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

14. Qual è l'output di questo codice?

```

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

```

- A. 42
- B. 41

- C. 4
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

15. Date la classe

```

class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}

```

e una variabile *n* di tipo *Nodo* tale che *n.info=""*,  
dire quale dei loro usi riportati qui sotto è corretto:

- A. `String s = n.getInfo();`
  - B. `int l = n.getInfo().length();`
  - C. `n=new Nodo(new Integer(0));`
  - D. `int v = n.getInfo().intValue();`
  - E. Nessuno dei precedenti
- 

16. Date le dichiarazioni:

```

Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];

```

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

- A. `f = (Boolean []) e;`
  - B. `f = (Boolean []) u;`
  - C. `u = (Exception []) e;`
  - D. `u = (Exception []) f;`
  - E. Nessuno dei precedenti
-



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

**Prova n. 37**

1. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

2. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

3. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile `n` di tipo `Nodo` tale che `n.info=""`, dire quale dei loro usi riportati qui sotto è corretto:

- A. `String s = n.getInfo();`
- B. `int l = n.getInfo().length();`
- C. `n=new Nodo(new Integer(0));`
- D. `int v = n.getInfo().intValue();`
- E. Nessuno dei precedenti

4. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

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

- A. `m = v;`
- B. `e = (Boolean) m;`
- C. `e = v;`
- D. `m = (Exception) e;`
- E. `v = e;`

5. Quale espressione va inserita nel punto indicato per stampare l'attributo `x` dichiarato nella classe `A`?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

6. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

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

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;
- C. u = (Exception []) e;
- D. u = (Exception []) f;
- E. Nessuno dei precedenti

7. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

8. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }
```

```
public static void main(String a[]) {
    C obj = new C( new Integer(1) );
    obj = null; // QUESTA
    System.out.println("done");
}
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

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

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

10. Dato il template

```

class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}

```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
- B. `Nodo<String> n = new Nodo<String>("");`
- C. `new Nodo<Boolean>("");`
- D. `new Nodo<Boolean>(true).getInfo().length();`
- E. Nessuno dei precedenti

11. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi static non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata private

12. Quale risultato produce questo codice?

```

class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}

```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

14. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```



```

if(s2 == s1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s4 == s3) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
}

```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

15. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}

```

- A. x
- B. A.this.x
- C. A.x
- D. super.x

E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}

```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti



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

**Prova n. 38**

1. Qual è l'output di questo codice?

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

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

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

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;
- C. u = (Exception []) e;
- D. u = (Exception []) f;
- E. Nessuno dei precedenti

3. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

4. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}
```

- A. x

- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

5. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

6. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile *n* di tipo *Nodo* tale che *n.info=""*, dire quale dei loro usi riportati qui sotto è corretto:

- A. `String s = n.getInfo();`
- B. `int l = n.getInfo().length();`
- C. `n=new Nodo(new Integer(0));`
- D. `int v = n.getInfo().intValue();`
- E. Nessuno dei precedenti

7. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi `static` non si applica il *dynamic method dispatch*

E. Una classe interna può essere dichiarata `private`

8. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

```

    }
}
}

```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

10. Quale risultato produce questo codice?

```

class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}

```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

11. Qual è l'output di questo codice?

```

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

```

```

}
catch( MyExc2 k ) {
    throw( new MyExc2() );
}
catch( Exception e ) {
    System.out.print(3);
}
finally {
    System.out.print(4);
    throw( new Exception() );
}
}
}

```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

12. Dato il template

```

class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}

```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
- B. `Nodo<String> n = new Nodo<String>("");`
- C. `new Nodo<Boolean>("");`
- D. `new Nodo<Boolean>(true).getInfo().length();`
- E. Nessuno dei precedenti

13. Quale risultato produce questo codice?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}

```

- A. 1
- B. 2
- C. Errore a tempo di compilazione

- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

---

14. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

---

15. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

---

16. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

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

- A. m = v;
- B. e = (Boolean) m;
- C. e = v;
- D. m = (Exception) e;
- E. v = e;



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

**Prova n. 39**

1. Qual è l'output di questo codice?

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

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

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

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;
- C. u = (Exception []) e;
- D. u = (Exception []) f;
- E. Nessuno dei precedenti

3. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010



4. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

5. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
  - B. 2
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

6. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
  - B. int l = n.getInfo().length();
  - C. n=new Nodo(new Integer(0));
  - D. int v = n.getInfo().intValue();
  - E. Nessuno dei precedenti
- 

7. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

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

- A. m = v;
  - B. e = (Boolean) m;
  - C. e = v;
  - D. m = (Exception) e;
  - E. v = e;
- 

8. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
  - B. 22
  - C. 3
  - D. 4
  - E. Nessuno dei precedenti
- 

9. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

10. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- 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 C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);

```

```

        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}

```

- A. 23431
  - B. Errore a tempo di compilazione
  - C. 234Exception in thread "main" java.lang.Exception
  - D. 24Exception in thread "main" java.lang.Exception
  - E. Nessuna delle precedenti
- 

13. Dato il template

```
class Nodo<T> {  
    private T info;  
    Nodo<T> next;  
    Nodo(T i) { info=i; }  
    public T getInfo() { return info; }  
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
  - B. `Nodo<String> n = new Nodo<String>("");`
  - C. `new Nodo<Boolean>("");`
  - D. `new Nodo<Boolean>(true).getInfo().length();` }
  - E. Nessuno dei precedenti
- 

14. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {  
    String x = "abc";  
    static Integer y;  
  
    C(Integer y) { this.y = y; }  
  
    public static void main(String a[]) {  
        C obj = new C( new Integer(1) );  
        obj = null; // QUESTA  
        System.out.println("done");  
    }  
}
```

- A. 1
- B. 2

- C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

15. Quale risultato produce questo codice?

```
class A {  
    int x;  
    A() {  
        x = 1;  
        System.out.print(x);  
    }  
}  
  
public class B extends A {  
    float x;  
    B() {  
        x = 2;  
        System.out.print(x);  
    }  
    public static void main(String a[]) {  
        new B();  
    }  
}
```

- A. 1
  - B. 2.0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

16. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi `static` non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata `private`
-



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

**Prova n. 40**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

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

```
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}
```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

3. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`

- B. `Nodo<String> n = new Nodo<String>("");`
- C. `new Nodo<Boolean>("");`
- D. `new Nodo<Boolean>(true).getInfo().length();`
- E. Nessuno dei precedenti

- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile `n` di tipo `Nodo` tale che `n.info=""`,  
dire quale dei loro usi riportati qui sotto è corretto:

- A. `String s = n.getInfo();`
- B. `int l = n.getInfo().length();`
- C. `n=new Nodo(new Integer(0));`
- D. `int v = n.getInfo().intValue();`
- E. Nessuno dei precedenti

5. Qual è l'output di questo codice?

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

- A. 42

6. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi **static** non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata *private*

7. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

8. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

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

- A. `f = (Boolean []) e;`
- B. `f = (Boolean []) u;`
- C. `u = (Exception []) e;`
- D. `u = (Exception []) f;`
- E. Nessuno dei precedenti

---

9. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

---

10. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

---

11. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
```

```
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

---

12. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

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

- A. m = v;
- B. e = (Boolean) m;
- C. e = v;
- D. m = (Exception) e;
- E. v = e;

---

13. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}
```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

14. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {  
    String x = "abc";  
    Integer y;  
  
    C(Integer y) { this.y = y; }  
  
    public static void main(String a[]) {  
        C obj = new C( new Integer(1) );  
        obj = null; // QUESTA  
        System.out.println("done");  
    }  
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

15. Quale risultato produce questo codice?

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}  
public class B extends A {  
    protected int x;  
    public static void main(String a[]) {  
        B obj = new B();
```

```
        obj.x = 1;  
        obj.set_x(2);  
        System.out.println( obj.x );  
    }  
}
```

- A. 1
  - B. 2
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

16. Quale risultato produce questo codice?

```
class A {  
    int x;  
    A() {  
        x = 1;  
        System.out.print(x);  
    }  
}  
public class B extends A {  
    float x;  
    B() {  
        x = 2;  
        System.out.print(x);  
    }  
    public static void main(String a[]) {  
        new B();  
    }  
}
```

- A. 1
  - B. 2.0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
-



Prova n. 1

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

1 marzo 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*

1 marzo 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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E



Studente e matricola:

Ora di inizio:

Ora di consegna:

---

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E



Studente e matricola:

Ora di inizio:

Ora di consegna:

---

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E



1	■				
2			■		
3				■	
4		■			
5			■		
6					■
7	■				
8					■
9			■		
10	■				
11		■			
12				■	
13				■	
14	■				
15			■		
16		■			

Risultato prova n. 1:

1				■	
2	■				
3		■			
4					■
5			■		
6	■				
7		■			
8			■		
9			■		
10					■
11				■	
12	■				
13				■	
14		■			
15	■				
16			■		

Risultato prova n. 2:

1	■				
2		■			
3			■		
4				■	
5			■		
6			■		
7				■	
8		■			
9					■
10				■	
11	■				
12	■				
13		■			
14					■
15	■				
16			■		

Risultato prova n. 3:

1			■		
2			■		
3		■			
4	■				
5			■		
6		■			
7		■			
8					■
9				■	
10			■		
11				■	
12	■				
13				■	
14					■
15	■				
16	■				

Risultato prova n. 4:

1	■				
2					■
3			■		
4		■			
5				■	
6		■			
7					■
8	■				
9			■		
10			■		
11			■		
12				■	
13				■	
14	■				
15	■				
16		■			

Risultato prova n. 5:

1			■		
2			■		
3	■				
4					■
5			■		
6				■	
7				■	
8	■				
9					■
10	■				
11				■	
12		■			
13		■			
14			■		
15		■			
16	■				

Risultato prova n. 6:

1				■	
2		■			
3			■		
4			■		
5		■			
6	■				
7			■		
8				■	
9	■				
10				■	
11			■		
12		■			
13					■
14	■				
15					■
16	■				

Risultato prova n. 7:

1				■	
2					■
3			■		
4				■	
5		■			
6	■				
7	■				
8	■				
9		■			
10					■
11			■		
12		■			
13			■		
14			■		
15	■				
16				■	

Risultato prova n. 8:



1	■				
2				■	
3				■	
4			■		
5		■			
6				■	
7					■
8	■				
9			■		
10					■
11		■			
12			■		
13		■			
14	■				
15			■		
16	■				

Risultato prova n. 9:

1					■
2	■				
3			■		
4		■			
5	■				
6				■	
7				■	
8					■
9				■	
10			■		
11		■			
12		■			
13	■				
14	■				
15			■		
16			■		

Risultato prova n. 10:

1				■	
2			■		
3		■			
4			■		
5	■				
6				■	
7	■				
8			■		
9		■			
10			■		
11				■	
12	■				
13					■
14					■
15		■			
16	■				

Risultato prova n. 11:

1			■		
2	■				
3				■	
4			■		
5			■		
6	■				
7					■
8	■				
9		■			
10		■			
11				■	
12		■			
13	■				
14				■	
15					■
16			■		

Risultato prova n. 12:

1	■				
2	■				
3			■		
4				■	
5				■	
6	■				
7					■
8			■		
9			■		
10		■			
11		■			
12		■			
13					■
14	■				
15				■	
16			■		

Risultato prova n. 13:

1	■				
2		■			
3		■			
4			■		
5			■		
6				■	
7	■				
8		■			
9	■				
10			■		
11			■		
12					■
13	■				
14					■
15				■	
16				■	

Risultato prova n. 14:

1		■		■	
2				■	
3	■				
4			■		
5			■		
6		■			
7				■	
8					■
9		■			
10	■				
11				■	
12			■		
13					■
14			■		
15	■				
16	■				

Risultato prova n. 15:

1	■			
2			■	
3				■
4			■	
5	■			
6		■		
7			■	
8	■			
9			■	
10		■		
11		■		
12			■	
13			■	
14	■			
15			■	
16				■

Risultato prova n. 16:



1					■
2	■				
3	■				
4		■			
5				■	
6			■		
7	■				
8			■		
9					■
10				■	
11				■	
12			■		
13		■			
14		■			
15			■		
16	■				

Risultato prova n. 17:

1			■		
2			■		
3					■
4			■		
5		■			
6			■		
7	■				
8	■				
9		■			
10				■	
11				■	
12					■
13				■	
14	■				
15		■			
16	■				

Risultato prova n. 18:

1		■			
2	■				
3				■	
4					■
5	■				
6			■		
7			■		
8	■				
9			■		
10		■			
11				■	
12			■		
13		■			
14	■				
15					■
16				■	

Risultato prova n. 19:

1			■		
2			■		
3	■				
4		■			
5	■				
6		■			
7				■	
8					■
9					■
10			■		
11		■			
12				■	
13	■				
14	■				
15			■		
16				■	

Risultato prova n. 20:

1		■			
2	■				
3				■	
4				■	
5			■		
6				■	
7					■
8		■			
9	■				
10	■				
11					■
12			■		
13		■			
14			■		
15	■				
16			■		

Risultato prova n. 21:

1				■	
2		■			
3		■			
4			■		
5	■				
6		■			
7					■
8				■	
9	■				
10			■		
11	■				
12					■
13				■	
14	■				
15			■		
16			■		

Risultato prova n. 22:

1	■				
2				■	
3	■				
4			■		
5					■
6				■	
7					■
8		■			
9			■		
10				■	
11	■				
12			■		
13		■			
14	■				
15		■			
16			■		

Risultato prova n. 23:

1			■		
2	■				
3			■		
4					■
5				■	
6				■	
7		■			
8	■				
9		■			
10			■		
11				■	
12		■			
13			■		
14	■				
15	■				
16					■

Risultato prova n. 24:



1			■		
2				■	
3		■			
4				■	
5					■
6		■			
7					■
8			■		
9			■		
10	■				
11			■		
12	■				
13	■				
14	■				
15		■			
16				■	

Risultato prova n. 25:

1				■	
2	■				
3					■
4		■			
5		■			
6			■		
7				■	
8		■			
9	■				
10	■				
11					■
12			■		
13			■		
14				■	
15			■		
16	■				

Risultato prova n. 26:

1				■	
2			■		
3			■		
4	■				
5				■	
6		■			
7	■				
8	■				
9					■
10	■				
11		■			
12			■		
13					■
14			■		
15		■			
16				■	

Risultato prova n. 27:

1			■		
2				■	
3			■		
4					■
5			■		
6	■				
7	■				
8				■	
9		■			
10				■	
11	■				
12					■
13		■			
14	■				
15		■			
16			■		

Risultato prova n. 28:

1			■		
2					■
3				■	
4	■				
5		■			
6		■			
7			■		
8				■	
9	■				
10			■		
11			■		
12					■
13	■				
14				■	
15		■			
16	■				

Risultato prova n. 29:

1				■	
2	■				
3		■			
4	■				
5			■		
6		■			
7					■
8			■		
9	■				
10			■		
11	■				
12		■			
13				■	
14					■
15				■	
16			■		

Risultato prova n. 30:

1					■
2				■	
3	■				
4			■		
5		■			
6				■	
7		■			
8		■			
9					■
10	■				
11				■	
12			■		
13			■		
14	■				
15			■		
16	■				

Risultato prova n. 31:

1				■	
2			■		
3				■	
4	■				
5	■				
6			■		
7		■			
8					■
9	■				
10	■				
11		■			
12			■		
13			■		
14		■			
15				■	
16					■

Risultato prova n. 32:



1					■
2		■			
3		■			
4			■		
5		■			
6			■		
7			■		
8				■	
9	■				
10	■				
11				■	
12	■				
13					■
14			■		
15	■				
16				■	

Risultato prova n. 33:

1		■			
2					■
3			■		
4	■				
5			■		
6				■	
7	■				
8	■				
9					■
10		■			
11			■		
12	■				
13		■			
14				■	
15				■	
16			■		

Risultato prova n. 34:

1	■				
2				■	
3			■		
4			■		
5	■				
6			■		
7		■			
8					■
9		■			
10				■	
11		■			
12	■				
13					■
14				■	
15	■				
16			■		

Risultato prova n. 35:

1					■
2	■				
3					■
4	■				
5				■	
6	■				
7		■			
8				■	
9		■			
10		■			
11			■		
12			■		
13	■				
14				■	
15			■		
16			■		

Risultato prova n. 36:

1	■				
2	■				
3			■		
4					■
5			■		
6			■		
7			■		
8		■			
9				■	
10		■			
11	■				
12					■
13	■				
14				■	
15				■	
16		■			

Risultato prova n. 37:

1				■	
2			■		
3		■			
4			■		
5			■		
6			■		
7	■				
8	■				
9	■				
10					■
11		■			
12		■			
13	■				
14				■	
15				■	
16					■

Risultato prova n. 38:

1				■	
2			■		
3				■	
4				■	
5	■				
6			■		
7					■
8			■		
9			■		
10		■			
11	■				
12		■			
13		■			
14	■				
15					■
16	■				

Risultato prova n. 39:

1	■				
2		■			
3		■			
4			■		
5				■	
6	■				
7	■				
8			■		
9			■		
10				■	
11				■	
12					■
13			■		
14		■			
15	■				
16					■

Risultato prova n. 40: