C

Istruzioni

Staccare questo foglio dal resto del blocchetto, ed utilizzarlo per segnare le risposte (sul retro di questa pagina). Segnare SUBITO nome, cognome e numero di matricola negli appositi spazi. Al termine della prova si dovrà consegnare SOLO questo foglio.

Il codice Java delle domande non mostra gli *import*, che si assumono essere correttamente presenti.

Vi sono tre tipi di risposte:

A) il codice esegue correttamente. Si indichi l'output

TEST x	il codice esegue correttamente, e l'output è →	

B) il codice potrebbe generare errori. Si indichi in quale riga l'errore avviene, e se si tratta di un compile error o di un runtime error. Se non vi sono errori, si indichi l'output. Va cerchiata l'opzione scelta (A, B o C)

TEST x	A	compile error alla riga _____ perchè →	
	B	runtime error alla riga _____ perchè →	
	C	il codice esegue correttamente, e l'output è →	

C) Domande vero/falso. Per questo tipo di domande, le risposte errate SOTTRAGGONO punti. Nel dubbio si consiglia di lasciare in bianco. Si riporti V o F nelle 8 caselle.

TEST x	Riportare la sequenza di V e F								

C

NOME, COGNOME	
NUMERO DI MATRICOLA	

TEST 1	A	compile error alla riga _____ perchè →	
	B	runtime error alla riga _____ perchè →	
	C	il codice esegue correttamente, e l'output è →	

TEST 2	il codice esegue correttamente, e l'output è →	

TEST 3	il codice esegue correttamente, e l'output è →	

TEST 4	A	compile error alla riga _____ perchè →	
	B	runtime error alla riga _____ perchè →	
	C	il codice esegue correttamente, e l'output è →	

TEST 5	A	compile error alla riga _____ perchè →	
	B	runtime error alla riga _____ perchè →	
	C	il codice esegue correttamente, e l'output è →	

TEST 6	il codice esegue correttamente, e l'output è →	

TEST 7	il codice esegue correttamente, e l'output è →	

TEST 8	il codice esegue correttamente, e l'output è →	

TEST 9	Riportare la sequenza di V e F									

C

Test 1

```
00  public class Tre {
01      class A {
02          public A(int k) {System.out.print(k);}
03          public void finalize() {System.out.print("C");}
04      }
05      class B extends A {
06          public B(int k) {System.out.print(k);}
07          public void finalize() {System.out.print("B");}
08      }
09      public static void main (String z[]){
10         new Tre();
11      }
12      Tre(){
13          A a=new B(3);
14          B b=(B)a;
15          a=null;
16          b=new B(3);
17          System.gc(); System.runFinalization();
18  }     }
```

ronchet 12/9/2016 11:32

Comment [1]: `//superclass constructor not found – line 13`

Test 2

```
01  #include <iostream>
02  using namespace std;
03  int x[] = {-2, -1, 0, 1, 2};
04  void f(int* x, int y[]) {
05      x[*y] = -y[*x];
06  }
07  int main(int argc, char** argv) {
08      int * p = x + 1;
09      f(p, p);
10      for (int * s = x; s < x + 5; s++) {
11          cout << *s;
12      }
13      return 0;
14  }
```

ronchet 12/9/2016 11:32

Comment [2]:
Output: **2-1012**

Test 3

```
01  public class F{
02      int x=2;
03      F(int x) {
04         f(x);
05         f();
06         System.out.println(x);
07      }
08      void f() { x++; System.out.print(x);}
09      void f(int x) { this.x++; x--;System.out.print(x);}
10      public static void main(String arg[]) {
11            F x=new F(3);
12      }}
```

ronchet 12/9/2016 11:49

Comment [3]: 243

C

Test 4 java B

```
00 class SuperB{
01     SuperB(){ System.out.print("S");}
02     void f() { System.out.print("Z");}
03 }
04 public class B extends SuperB {
05     B(){ System.out.print("B");}
06     void f() { System.out.print("C");}
07     public static void main(String args) {System.exit(1);}
08     public static void main(String[] args) {
09         SuperB b=new SuperB();
10         B a=(B)b;
11         a.f();
12     }}
```

Marco Ronchetti 12/9/2016 11:21

Comment [4]: Runtime error at line 10 java.lang.ClassCastException: it.unitn.prog2.esame.SuperB cannot be cast to it.unitn.prog2.esame.B

Test 5

```
01 public class Due {
02     Collection<Due> s=new HashSet<Due>();
03     static int k,j;
04     Due(int k, int j) {this.k=k;   this.j=j;}
05     public boolean equals(Object d){
06         return k-j==((Due)d).j-((Due)d).k;
07     }
08     public int hashCode(){return 1;}
09     public static void main(String[] m){
10         s.add(new Due(1,2)); s.add(new Due(0,1));
11         s.add(new Due(2,1)); s.add(new Due(1,0));
12         System.out.print(s.size());
13         for (Due x:s){System.out.print(x.k+" "+x.j);}
14 }   }
```

ronchet 9/6/2014 23:30

Comment [5]: errore alla linea 10 - accessing non-static variable

Test 6 java B

```
00 class SuperB{
01     SuperB(){ System.out.print("D");}
02     void f() { System.out.print("Z");}
03 }
04 public class B extends SuperB {
05     B(){ System.out.print("E");}
06     void f() { System.out.print("O");}
07     public static void main(String pippo) {System.exit(1);}
08     public static void main(String[] args) {
09         B b=new B();
10         SuperB a=(SuperB)b;
11         a.f();
12     }}
```

Marco Ronchetti 12/9/2016 11:48

Comment [6]: DEO

C

Test 7

```
00  public class Sei {
01      char f() { return '6'; }
02      public static void main(String e[]) {
03          Sei a = new Sei();
04          Sei b = new Sette();
05          Sette c = new Sette();
06          System.out.print(a.f() + " " + b.f() + " " + c.f() + " ");
07          char ch[] = {'C', 'A', 'C', 'A', 'C','A'};
08          int i1 = 0, i2 = 2, i3 = 4;
09          if (a.equals(b)) i1++;
10          if (b.equals(a)) i2++;
11          if (c.equals(b)) i3++;
12          System.out.println(ch[i1] + " " + ch[i2] + " " + ch[i3]);
13  }   }
14  class Sette extends Sei {
15      char f() { return '7'; }
16      public boolean equals(Object a) {
17          return (a instanceof Sei);
18      }
19      public int hashCode() { return 3; }
20  }
```

ronchet 12/9/2016 11:46

Comment [7]: `6 7 7 C A A`

Test 8

```
00  public class Uno {
01      static Collection c=new HashSet();
02      public static void main(String a[]) {
03          Collection c=new LinkedList();
04          Uno u=new Uno();
05          c.add(u); c.add(u); c.add(u);
06          u.f();
07          System.out.print(c.size());
08          System.gc();System.runFinalization();
09      }
10      void f() {
11          A a=new A("S");
12          A b=new A("K");
13          c.add(b);
14      }
15      class A {
16          String s="";
17          A(String s) {this.s=s; System.out.print(this);}
18          public String toString(){return s;}
19          public void finalize(){System.out.print(this);}
20  }   }
```

ronchet 12/9/2016 11:47

Comment [8]: `SK3S`

C

Test 9 – scrivere nel campo per l'output del test la sequenza risultante indicando V per le affermazioni vere e F per quelle false

9.1	**Il metodo finalize() chiama automaticamente il corrispondente metodo della superclasse**
9.2	**L'esistenza in una classe di un metodo f(int x), e in una sua superclasse di un metodo f(String s) è un esempio di overriding**
9.3	**L'esistenza in una classe di un metodo f(int x) e di uno f(String s) è un esempio di overloading**
9.4	**Se a.equals(b) è vero, deve essere a.hashCode==b.hashCode**
9.5	**Se a.equals(b) è falso, deve essere a.hashCode!=b.hashCode**
9.6	**Il costruttore chiama automaticamente il costruttore della superclasse con gli stessi parametri. Se nella superclasse non è disponibile un costruttore con la stessa firma, viene chiamato il costruttore vuoto.**
9.7	**int a[] è un oggetto**.
9.8	**In un programma ci possono essere due classi chiamate F**

ronchet 12/9/2016 11:43

Comment [9]:
1-F 2-F 3-V 4-V
5-F 6-F 7-V 8-V