

## Anteprima di test

parte teorica 20 Gennaio 2015

**Data: Mon Jan 26 12:44:27 2015 Punteggi massimi: 38**

### 1. Overriding/Overloading (2) (3 Punti)

**Dato il seguente codice**

```
1class Elaboratore{
2  void setQuantity(int q){}
3}
4
5class Phone extends Elaboratore{
6  void setQuantity(long l){}
7}
```

**Quali di queste sono giuste**

- ☐ Phone fa overriding del metodo setQuantity di Elaboratore *(Selezionato = 0 Punti, Non selezionato = 1 Punto)*
- ☐ Phone contiene un errore: non può definire setQuantity in questo modo! *(Selezionato = 0 Punti, Non selezionato = 1 Punto)*
- ☒ Phone fa overloading del metodo setQuantity di Elaboratore *(Selezionato = 1 Punto, Non selezionato = 0 Punti)*

### 2. overriding di equals (8 Punti)

**Data la seguente classe e gli oggetti definiti come segue**

```
1 public class A {
2     String name;
3
4     public A(String s) {
5         name = s;
6     }
7
8     public boolean equals(A a) {
9         return this.name == a.name;
10    }
11}
12String p1 = new String("pippo");
13String p2 = new String("pippo");
14Object o = new A(p1);
15A a1 = new A(p1);
16A a2 = new A(p2);
```

**Quale è l'output prodotto dalle seguenti istruzioni (metti errore se pensi ci sia un errore)?**

```
System.out.println(p1.equals(o));false (1 Punto)
System.out.println(o.equals(o));true (1 Punto)
System.out.println(o.equals(new Integer(5)));false (1 Punto)
System.out.println(o.equals(a1));false (1 Punto)
System.out.println(a1.equals(a1));true (1 Punto)
System.out.println(a1.equals(a2));false (3 Punti)
```

FALSE  
TRUE  
FALSE  
FALSE  
TRUE  
FALSE

### 3. Passaggio per riferimento in c++ (9 Punti)

**Data la seguente funzione**

```
1void copy(int& x, int& y) {
```

```

2      x = y;
3      x--;
4      y++;
5}

```

Qual'è l'output prodotto dalle seguenti istruzioni? Se pensi conteng un errore, scrivi errore

```

int main() {
int a = 5;
int b = 6;
int& h = a;

copy(a,b);
cout << a << endl;5 (1 Punto)
cout << b << endl;7 (1 Punto)
cout << h << endl;5 (1 Punto)
copy(h,b);
cout << a << endl;6 (1 Punto)
cout << b << endl;8 (1 Punto)
cout << h << endl;6 (1 Punto)
b=7;
cout << a << endl;6 (1 Punto)
cout << b << endl;7 (1 Punto)
cout << h << endl;6 (1 Punto)
return 0;
}

```

$a = b = 6$

5  
7  
5

$h = b = 7$

6  
8

6

6

7

6

#### 4. Dynamic Binding Java (1) (4 Punti)

Date le seguenti dichiarazioni:

```

1 class Elaboratore {
2     void setCPU(int l) {
3         System.out.println("E");
4     }
5 }
6
7 class Phone extends Elaboratore {
8     void setCPU(int l) {
9         System.out.println("P");
10    }
11}
12
13class Computer extends Elaboratore {
14    void setCPU(short l) {
15        System.out.println("C");
16    }
17}
18
19...
20Object oe = new Elaboratore ();
21Elaboratore ee = new Elaboratore ();
22Elaboratore ep = new Phone ();
23Elaboratore ec = new Computer ();
24short myfreq = 30;

```

Quale è l' input prodotto dalle sequenti istruzioni (errore se pensi ci sia un errore)?

```

oe.setCPU(myfreq) errore (1 Punto)
ee.setCPU(myfreq) E (1 Punto)
ep.setCPU(myfreq) P (1 Punto)

```

ERRORE

E

P

**ec.setCPU(myfreq) E (1 Punto)**

10

## 5. C++ virtual functions ed ereditarietà (2) (14 Punti)

**Date le seguenti classi**

```
1 class Z{
2 public:
3     virtual void m(){ cout << "Z" << endl;}
4 };
5
6 class ZPRI1: private Z{
7 };
8
9 class ZPRI2: private Z{
10 public:
11     virtual void m(){ cout << "ZPRI2" << endl;}
12 };
13
14 class ZPUB1: public Z{
15 };
16
17 class ZPUB2: public Z{
18 public:
19     virtual void m(){ cout << "ZPUB2" << endl;}
20 };
```

**Scrivi l'output delle seguenti coppie di istruzioni. ERR se pensi ci sia un errore.**

Z a1; a1.m();Z (1 Punto)  
 ZPRI1 a2; a2.m();ERR (1 Punto)  
 ZPRI2 a3; a3.m();ZPRI2 (1 Punto)  
 ZPUB1 a4; a4.m();Z (1 Punto)  
 ZPUB2 a5; a5.m();ZPUB2 (1 Punto)  
 a1 = a2; a1.m();ERR (1 Punto)  
 a1 = a3; a1.m();ERR (1 Punto)  
 a1 = a4; a1.m();Z (1 Punto)  
 a1 = a5; a1.m();Z (1 Punto)  
 Z\* p = &a1; p -> m();Z (1 Punto)  
 p = &a2;p -> m();ERR (1 Punto)  
 p = &a3;p -> m();ERR (1 Punto)  
 p = &a4;p -> m();Z (1 Punto)  
 p = &a5;p -> m();ZPUB2 (1 Punto)

Z  
 ERRONE  
 ZPRI2  
 Z  
 ZPUB2  
 ERRONE  
 ERRONE  
 Z  
 Z  
 ERRONE  
 ERRONE  
 Z  
 ZPUB2