

Cognome \_\_\_\_\_ Nome \_\_\_\_\_ Matricola \_\_\_\_\_

## Fond. di Informatica I (N.O) / Fond. di Informatica (V.O. foglio 1); 12-12-02

**Regole del Gioco: 1. Scrivere le proprie risposte nei riquadri contenuti in questo foglio; solo le risposte scritte nei riquadri saranno valutate. 2. Prima di consegnare il foglio scrivere il proprio nome, cognome e matricola sul foglio stesso. 3. Non si può parlare, non si possono consultare appunti, non si può usare la calcolatrice.**

**Domanda 1:** con riferimento al seguente codice, indicare: variabili locali, variabili di istanza e variabili di classe

```
class Studente{
    private String nome;
    private String cognome;
    private int matricola;
    private static int numStudenti = 0;
    public Studente (String name, String surname){
        this.nome=name;
        this.nome=surname;
        numStudenti ++;
    }
    public boolean equals (Studente s){
        boolean uguale;
        uguale = (this.matricola==s.matricola);
        return uguale;
    }
}
```

**Domanda 2:** Dire cosa visualizza sullo schermo il metodo main della seguente classe:

```
import java.io.*;
class Esercizio{
    public static void main (String[] args){
        String s="1+2+3+\"abc\"";
        System.out.println (s);
        for (int i=1; i<=3; i++)
            s = s + i;
        System.out.println (s);
        System.out.println ((s.toUpperCase()).substring(0,2));
    }
}
```

## Fond. di Informatica (V.O. foglio 2); 2-12-02

**Regole del Gioco:** 1. Scrivere le proprie risposte nei riquadri contenuti in questo foglio; solo le risposte scritte nei riquadri saranno valutate. 2. Prima di consegnare il foglio scrivere il proprio nome, cognome e matricola sul foglio stesso. 3. Non si può parlare, non si possono consultare appunti, non si può usare la calcolatrice.

**Domanda 3:** Il seguente codice contiene una istruzione errata. Riportare l'istruzione errata nel riquadro. Indicare inoltre una versione corretta del metodo :

```
import java.util.*;
class Domanda{
    public static void main (String[] args){
        Vector v = new Vector();
        v.addElement("ci sono");
        v.addElement(10);
        v.addElement("studenti");
        System.out.println (v.elementAt(0));
        System.out.println (v.elementAt(1));
        System.out.println (v.elementAt(2));
    }
}
```

### **Domanda 4:**

Con riferimento agli algoritmi di ordinamento di un array di N interi (risposta esatta = +1, non data = 0, sbagliata = -1)

1. Se gli elementi dell'array sono ordinati il selection sort ha complessità  $O(N)$
  2. Se gli elementi dell'array non sono ordinati il selection sort ha complessità  $O(N^2)$
  3. Il merge sort ha complessità  $O(N)$  nel caso peggiore
  4. Non esiste un algoritmo di ordinamento di complessità inferiore ad  $O(N \log N)$ .