

---

**FONDAMENTI DI PROGRAMMAZIONE B**

---

*Tempo a disposizione: 30 minuti*

Nome ..... Cognome ..... Matricola .....

*Per accedere alla prova di programmazione è necessario rispondere correttamente ad almeno il 70% delle domande*

1. [C++] Si considerino le classi A, B, C e D. La classe B è derivata da A, mentre le classi C e D sono derivate da B. La seguente funzione

```
int move(B r) {...}
```

può accettare come argomenti oggetti

- ☐ a di tipo A e B
  - ☐ b di tipo B, C e D
  - ☐ c di tipo A, B e C
  - ☐ d esclusivamente di tipo B e delle sue superclassi
  - ☐ e nessuna delle precedenti
2. [C++] Se una classe definisce esplicitamente il costruttore di copia, allora è opportuno
- ☐ a definire anche il costruttore senza parametri
  - ☐ b definire esplicitamente anche l'operatore di assegnamento ma non il distruttore
  - ☐ c definire esplicitamente anche il distruttore e l'operatore di assegnamento
  - ☐ d nessuna delle precedenti
3. [C++] Si consideri il template di classe `Set<T>`. La classe `Set<float>` è una superclasse di `Set<int>` ☐ T ☐ F
4. [C++] La memoria allocata nello heap viene deallocata automaticamente dal garbage collector ☐ T ☐ F
5. [C++] Si supponga che la classe C contenga il metodo `void f(int x, int y) const {...}`. Allora i parametri del metodo f sono tutti costanti. ☐ T ☐ F

6. [Java] Si considerino le classi A, B e C. La classe B è derivata da A e la classe C è derivata da B. La classe B definisce un metodo `g` che la classe C **non** ridefinisce. Si consideri il seguente frammento di codice.

```
C o = new C();  
o.g();
```

- ☐ *a* viene sollevata una `ClassCastException`
- ☐ *b* viene invocato il metodo `g` definito nella classe B
- ☐ *c* viene rilevato un errore a tempo di compilazione
- ☐ *d* viene invocato il metodo `hashCode` definito nella classe C
- ☐ *e* nessuna delle precedenti

7. [Java] Si consideri la seguente dichiarazione di attributo all'interno di una classe C:

```
private final int x;
```

Si indichi la risposta corretta.

- ☐ *a* è un attributo d'istanza con visibilità privata e non modificabile (dopo la sua inizializzazione nel costruttore)
- ☐ *b* è un attributo di classe con visibilità privata e non modificabile (dopo la sua inizializzazione nel costruttore)
- ☐ *c* è un attributo di classe con visibilità privata e modificabile
- ☐ *d* è un attributo d'istanza con visibilità di package e modificabile
- ☐ *e* nessuna delle precedenti

8. [Java] Data la classe A, si consideri il seguente frammento di codice

```
A a1 = new A();  
A a2 = a1;
```

Dopo l'esecuzione del frammento di codice riportato sopra

- ☐ *a* `a1` e `a2` fanno riferimento allo stesso oggetto
- ☐ *b* `a2` fa riferimento ad un oggetto che è una copia profonda (*deep copy*) dell'oggetto riferito da `a1`
- ☐ *c* `a2` fa riferimento ad un oggetto che è una copia leggera (*shallow copy*) dell'oggetto riferito da `a1`
- ☐ *d* nessuna delle precedenti

9. [Java] Data una classe A, l'istruzione `A a;` è equivalente all'istruzione `A a = new A();`

☐ T ☐ F

10. [Java] La parola chiave `protected` applicato ad un campo di una classe indica che il campo, una volta inizializzato nel costruttore, non può essere modificato.

☐ T ☐ F