
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++] Se una classe non definisce esplicitamente un costruttore

- ☐ a non è possibile istanziare oggetti di quella classe
- ☐ b la classe ha un costruttore di default
- ☐ c viene generato un errore a tempo di compilazione
- ☐ d viene generato un errore a tempo di esecuzione
- ☐ 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 il distruttore e l'operatore di assegnamento
- ☐ c definire esplicitamente anche l'operatore di assegnamento ma non il distruttore
- ☐ d definire esplicitamente anche il distruttore ma non l'operatore di assegnamento
- ☐ e nessuna delle precedenti

3. [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 `foo`

```
int foo(B obj) {...}
```

può accettare come argomenti oggetti

- ☐ a di tipo A, B e `Object`
- ☐ b esclusivamente di tipo B
- ☐ c di tipo B, C e D
- ☐ d di tipo A e B

4. [C++] Si supponga che la classe C contenga il metodo `void f() const {...}`. Il metodo `f` della classe C è un metodo costante. ☐ T ☐ F

5. [C++] Se una classe non definisce esplicitamente il distruttore, ne ha uno di default. ☐ T ☐ F

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

```
A o = new C();  
((C) o).fee();
```

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

7. [Java] Quando un membro di una classe è dichiarato come `protected`, allora il membro può essere accessibile

- ☐ a solo all'interno della stessa classe in cui è dichiarato
- ☐ b solo all'interno della stessa classe e delle sottoclassi della classe in cui è dichiarato
- ☐ c solo all'interno della stessa package in cui è dichiarato
- ☐ d solo all'esterno della stessa package in cui è dichiarato

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

```
A x = new A();  
A y = x;
```

Dopo l'esecuzione del frammento di codice riportato sopra

- ☐ a x e y fanno riferimento allo stesso oggetto
- ☐ b y fa riferimento ad un oggetto che è una copia profonda (*deep copy*) dell'oggetto riferito da x
- ☐ c y fa riferimento ad un oggetto che è una copia leggera (*shallow copy*) dell'oggetto riferito da x
- ☐ d viene sollevata un'eccezione a tempo d'esecuzione
- ☐ e nessuna delle precedenti

9. [Java] L'istruzione `String str;` è equivalente all'istruzione `String str = null;`

☐ T ☐ F

10. [Java] Un parametro di tipo di una classe generica (e.g., `Stack<T>`) non può essere specializzato con tipi primitivi.

☐ T ☐ F