FONDAMENTI DI PROGRAMMAZIONE B

Tempo a disposizione: 30 minuti

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

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

1. **[C++]** Qual è il valore di `y` dopo la chiamata alla funzione `func`?

```
void func(const int& x) {
        x = x + 5;
}

int main() {
        int y = 3;
        func(y);
        cout << y << endl;
}
```

[a] 3 [b] 5 [c] 8 [d] nessuna delle precedenti

2. **[C++]** Si considerino le classi `Persona`, `Studente` e `Docente`. Le classi `Studente` e `Docente` sono derivate da `Persona`. La seguente funzione

```
void f(Persona p) {...}
```

può accettare come argomenti oggetti

[a] esclusivamente di tipo `Persona` e delle sue superclassi
[b] di tipo `Persona`, `Studente` e `Object` ma non `Docente`
[c] di tipo `Persona`, `Studente` e `Docente`
[d] esclusivamente di tipo `Docente` e `Studente`
[e] nessuna delle precedenti

3. **[C++]** È possibile allocare gli oggetti esclusivamentre tramite allocazione dinamica [T] [F]

4. **[C++]** Si consideri il template di classe `Stack⟨T⟩`. Allora `Stack⟨int⟩` è sottotipo di `Stack⟨float⟩` [T] [F]

5. **[C++]** Se una classe `C` alloca memoria dinamica, è opportuno ridefinire il costruttore di copia. [T] [F]

6. **[Java]** Si considerino le classi `Forma`, `Triangolo` e `Quadrato`. Le classi `Quadrato` e `Triangolo` sono derivate da `Forma`. La classe `Forma` contiene un metodo `area` che le classi `Quadrato` e `Triangolo` ridefiniscono. Si consideri il seguente frammento di codice.

```
Forma f = new Triangolo();
((Triangolo) f).area();
```

[a] viene sollevata una `ClassCastException`
[b] viene rilevato un errore a tempo di compilazione
[c] viene invocato il metodo `area` definito nella classe `Triangolo`
[d] viene invocato il metodo `area` definito nella classe `Forma`
[e] nessuna delle precedenti

7. [**Java**] Gli oggetti possono essere passati ad un metodo

- [*a*] per valore oppure per riferimento
- [*b*] esclusivamente per riferimento
- [*c*] esclusivamente per valore
- [*d*] nessuna delle precedenti

8. [**Java**] Data una classe `C`, l'istruzione `C c;` è equivalente all'istruzione `C c = new C();` [T] [F]

9. [**Java**] L'eccezione `public DenominatoreNullo extends Exception { }` è un'eccezione controllata. [T] [F]

10. [**Java**] Il campo `protected int p;` di una classe `C` è visibile esclusivamente nel package della classe `C`. [T] [F]