

Nome: \_\_\_\_\_ Código: \_\_\_\_\_

**Notas:**

- Responda às questões seguintes, indicando a opção correta (em maiúsculas)
- Cada resposta errada vale -15% da cotação da pergunta
- Esta prova é composta por 4 páginas

1. A classe `Estudante` é classe derivada (derivação pública) da classe `Pessoa`. É verdade que a classe `Estudante` pode redefinir:

- A. Qualquer membro-função da classe `Pessoa`
- B. Apenas membros-função da classe `Pessoa` nesta declarados como não privados
- C. Apenas membros-função da classe `Pessoa` nesta declarados como não estáticos
- D. Apenas membros-função da classe `Pessoa` nesta declarados como não constantes
- E. Nenhuma das possibilidades anteriores

Resposta: \_\_\_\_\_

2. Identifique qual dos seguintes fragmentos de código realiza a chamada do construtor de cópia para a classe `ABC`. (assuma que `umaFuncao` possui o argumento de tipo apropriado)

- A. `ABC x1; ABC x2; x1=x2;`
- B. `ABC x; umaFuncao(x);`
- C. `ABC *apt; umaFuncao(apt);`
- D. `ABC x; umaFuncao(&x);`
- E. Nenhuma das possibilidades anteriores

Resposta: \_\_\_\_\_

3. Considere o código seguinte (as classes `Function` e `Double` estão parcialmente definidas):

```
class Function {
public:
    virtual float evaluate(float x)=0;
    string info();
};
Function *f = new Double();
float value = f->evaluate(4.0);
string info = f->info();

class Double: public Function {
public:
    float evaluate(float x);
    string info();
};
```

- A. O código está incorreto, devido à atribuição inválida (`=0`) na declaração de `Function::evaluate`
- B. `value = 0` e `info` é o resultado da invocação de `Function::info()`
- C. `value` é o resultado de `Double::evaluate()` e `info` é o resultado de `Double::info()`
- D. `value` é o resultado de `Double::evaluate()` e `info` é o resultado de `Function::info()`
- E. Nenhuma das possibilidades anteriores

Resposta: \_\_\_\_\_

4. Considerando a classe `OneClass`, é correto afirmar que:

```
class OneClass {  
    static int value;  
    // ...  
};
```

- A. `OneClass` está errada, pois um membro estático não pode ser um membro privado
- B. O membro estático `value`, por ser privado, torna a classe `OneClass` abstrata
- C. Para manipular `value`, é necessário implementar um membro-função estático
- D. O membro-dado `value` nunca será herdado por classes derivadas de `OneClass`, por ser estático
- E. Nenhuma das possibilidades anteriores

Resposta: \_\_\_\_\_

5. Considere o seguinte fragmento de código:

```
class ClassX {  
    //...  
    friend ClassY;  
};
```

- A. O qualificador `friend` produz o mesmo efeito que tornar `ClassY` uma classe derivada de `ClassX`
- B. `ClassY` tem acesso a todos os membros de `ClassX`
- C. `ClassY` tem acesso apenas aos membros-função de `ClassX`
- D. O qualificador `friend` não produz qualquer efeito se `ClassY` já for uma classe derivada de `ClassX`
- E. Nenhuma das possibilidades anteriores

Resposta: \_\_\_\_\_

6. Considere a classe `Person` definida a seguir. Assinale a afirmação verdadeira.

```
1. class Person {  
2.     string name;  
4. public:  
5.     Person(string nm) {  
6.         name=nm;  
7.         if ( nm.length() < 3 ) throw InvalidName(nm);  
8.     }  
9. };
```

- A. Só podem ser construídos objetos da classe `Person` com `name`  $\geq 3$  caracteres
- B. Podem ser construídos objetos da classe `Person` com qualquer `name`, mas é lançada uma exceção quando o número de caracteres é inferior a 3
- C. As linhas 6 e 7 estão trocadas. A atribuição ao membro-dado `name` tem de estar após verificação da exceção.
- D. No construtor da classe não é permitido lançar exceções
- E. Nenhuma das possibilidades anteriores

Resposta: \_\_\_\_\_

Nome: \_\_\_\_\_ Código: \_\_\_\_\_

7. `Shape` é uma classe abstrata. É correto afirmar que:

- A. Todos os membros-função da classe `Shape` devem ser abstratos
- B. Só é possível criar objetos da classe `Shape` se estes forem constantes
- C. `Shape` pode ser uma classe base, mas não uma classe derivada
- D. Não é possível criar objetos da classe `Shape`
- E. Nenhuma das possibilidades anteriores

Resposta: \_\_\_\_\_

8. Considere a classe `Two`:

```
template <class F, class S> class Two {  
    F v1;  
    S v2;  
public:  
    //...  
    Two<S,F> reverse() { return Two<S,F>(v2,v1); }  
};
```

- A. O código está incorreto, porque o template só admite um tipo genérico de dados
- B. O membro-função `reverse` está incorreto, pois a classe `Two` apenas está definida para `Two<F,S>` e não `Two<S,F>`
- C. O código está correto
- D. A classe `Two` é abstrata, pois está definida para tipos genéricos de dados
- E. Nenhuma das possibilidades anteriores

Resposta: \_\_\_\_\_

9. Indique a complexidade temporal da seguinte função:

```
void function1(vector<int> &v1) {  
    for(int i=0; i<v1.size(); i++)  
        for(int j=1; j<v1.size(); j*=2)  
            cout << i << " - " << j << endl;  
}
```

- A.  $O(\log n)$
- B.  $O(n)$
- C.  $O(n \log n)$
- D.  $O(n^2)$
- E. Nenhuma das possibilidades anteriores

Resposta: \_\_\_\_\_

10. A complexidade temporal de um algoritmo é  $O(n^2)$ . Este algoritmo apresentou um tempo de execução de 10 segundos para um conjunto de dados de entrada de tamanho  $n$ . Se o tamanho dos dados de entrada for duplicado, quanto tempo demorará (aproximadamente) a execução deste algoritmo?
- A. 10 segundos
  - B. 20 segundos
  - C. 40 segundos
  - D. 100 segundos
  - E. Nenhuma das possibilidades anteriores

Resposta: \_\_\_\_\_