

Curso: Engenharia Eletrônica, **Disciplina:** Fundamentos de Programação II (IF62C), **Turma:** S__
Professores: Hermes Del Monego (), Jean Simão (), Robinson Vida () **Data:** __/__/__.

Aluno: _____ **Código:** _____, **Início** __:__:__ **Fim** __:__:__

1ª Prova

1) [2,5 pt] Um programador criou um pequeno programa para representar a classe Volume (definição de classe e um método main). Esse código, ao ser compilado, apresentará 4 (quatro) erros de compilação. Um desses erros, ocorre mais de uma vez. Quais erros são esses e como eles podem ser corrigidos?

```
#include <iostream>

using namespace std;

class Volume
{
private :
    double comprimento, largura, altura;
public :
    // definicao de metodos de acesso - sets
    void setComprimento(double v){ comprimento = v; }
    void setLargura(double v){ largura = v;}
    void setAltura(double altura) { altura = altura; }

    // definicao de metodos de acesso - gets
    double getComprimento() const {return this->comprimento;}
    double getLargura() const {return this->largura;}
    double setAltura() const {return this->altura;}

    // definicao de construtores e destrutores
    virtual Volume()
    { std::cout << "Primeiro Construtor."; }

    virtual Volume(double $v1 = 1.0
        , double $v2 = 1.0
        , double $v3 = 1): comprimento($v1),
        largura($v2)
        , altura($v3)
    { std::cout << "Segundo Construtor."; }

    virtual ~Volume(){ std::cout << "Destrutor chamado."; }
};

int main()
{
    Volume *v = new Volume();
    delete v;
    return 0;
}
```

Quadro de Perguntas

1) Primeiro erro

- a) Qual é o erro?
- b) Como consertar?

2) Segundo erro

- a) Qual é o erro?
- b) Como consertar?

3) Terceiro erro

- a) Qual é o erro?
- b) Como consertar?

4) Quarto erro

- a) Qual é o erro?
 - b) Como consertar?
-

2) ^[A=2,5,B=2,5 pt] De acordo com o texto abaixo resolver parte A e B em um único código já que ambas se complementam. Usar para tal os bons princípios da Orientação a Objetos. ¹

- A) Faça um programa em C++ que contenha uma classe Funcionário com atributos nome, salario_base e adicional. Particularmente, haverá um método virtual de calculo_salario que soma salario_base e adicional, armazenando isto em um atributo outro chamado salario_efetivo. Crie ainda uma classe Empresa com um atributo nome. Esta classe terá ainda uma coleção de apontamentos de instâncias (i.e., objetos) da classe Funcionário, coleção essa implementada por meio de uma instância da classe vector da STL. Em todas as classes defina os atributos como privados ou protegido e crie métodos públicos para acessar e modificar os atributos. Finalmente, faça um programa que crie uma instância da classe Empresa e adicione a ela apontamentos de instâncias da classe Funcionário.
- B) Complemente o programa anterior para admitir novos tipos de funcionários, os quais serão definidos a partir de novas classes derivadas da classe Funcionário. Serão definidas as subclasses para funcionários do tipo Gerente e Analista. O programa deve ser complementado para:
- b.1) sobrecarregar o método para calculo_salario nas subclasses adicionando respectivamente 20% e 10% de salario_efetivo para as classes Gerente e Analista.
 - b.2) método na classe Empresa para calcular e exibir salário de cada um dos funcionários a partir do seu método calculo_salario. Utilizar técnicas de polimorfismo propriamente dito para tal
 - b.3) Instanciar e adicionar uma instância de Gerente e uma instância de Analista. Isto feito, adicioná-las a coleção de funcionários da instância da classe Empresa, ambas mencionadas na parte A da questão.

3) ^[2,5 pt] Em se tratando de Herança, recorreremos aos princípios de encapsulamento, sobrecarga, polimorfismo e outros bons princípios de OO. Por isso é correto afirmar que:

- I) Em classes, o atributo *private* é definido por padrão, ou seja, os membros que não tenham definidos os seus atributos de acesso explicitamente, serão definidos como privados. Este comportamento revela a necessidade de resguardar os membros de uma classe através de atributos de restrições. (V)

¹Questão adaptada de <http://ptdocz.com/doc/596833/> lista de exercícios feita pelo Prof. Carlos Eduardo Batista, acessado em 17/04/2016].

- II) Os membros de *protected* são semelhantes a membros *private*, exceto pelo fato de que podem ser acessados por qualquer membro de uma classe derivada (sub-classe). (V)
- III) Membros *protected* não podem ser acessados por nenhum objeto declarado fora da classe e de suas subclasses. (F)
- IV) A propriedade *private* será vista e usada apenas dentro da classe e de classes descendentes (F).
- V) Herança privada, indica que todos os membros públicos ou protegidos da classe pai serão privados na classe filho. Apesar da classe filho ter acesso a atributos/métodos da classe pai, seus objetos não o terão. (V)
- VI) Herança protegida, indica que tanto os membros públicos quanto os protegidos da classe pai serão protegidos da classe filha. (V)

Selecione a alternativa correta:

- a) I, II, V e VI são verdadeiras.
- b) I, II, V e VI são falsas.
- c) III, IV e V são verdadeiras
- d) II, IV e VI são falsas
- e) Todas são verdadeiras