



INSTITUTO FEDERAL DA PARAÍBA  
CAMPUS CAMPINA GRANDE  
BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO  
DISCIPLINA DE POO e LAB. POO  
PROF. VICTOR ANDRÉ PINHO DE OLIVEIRA

Atividade Unidade I - 4 - Introdução à OO

## Instruções

Responda às questões abaixo. Pode usar este próprio documento.

Questões teóricas valem 0,5 ponto. Questões práticas valem 2 pontos.

## Questões

1. Explique a diferença entre classes e objetos.

Resposta: De maneira resumida, classe seria o tipo do objeto, onde possui todos os atributos e métodos, enquanto que o objeto seria, se comparado a programação estruturada, uma variável que dispõem de todos os atributos e métodos da classe criada.

2. Qual a diferença entre uma variável local e um membro de dados (atributo)?

Resposta: Variável local são variáveis que são definidas e “destruídas” dentro de uma escopo que não seja o escopo da função principal, como por exemplo, uma função criada pelo o programador.

Já os membro de dados (atributos), são variáveis que são definidas dentro de uma classe, e que quando um objeto é instanciado com o tipo da classe criada, esse objeto recebe todas as características de seus atributos.

3. Considere o objeto Hamster, um animal fofinho que parece um rato. Descreva 3 atributos e 3 comportamentos.

Resposta: Atributos - possuem pelo, são mamíferos, e são animais de porte pequeno

Comportamentos - Como métodos podemos destacar : Roer, Dormir e Andar

4. Considere o objeto mouse (periférico). Descreva 3 atributos e 2 comportamentos.

Resposta: Atributos: Possui botões, um scroll, e um cabo USB (julgando que seja não seja um mouse sem fio)

Comportamentos: possui os métodos Clicar e Mover

5. Qual a unidade básica de programação de C++?

Resposta: A unidade básica seria a orientação ao objeto.

6. O que é instanciar?

Resposta: Instanciar seria a definição de um objeto, ou seja, a criação de um objeto no escopo principal.

7. De que maneira uma Classe oferece serviços para outrem?

Resposta: Através dos métodos dentro da classe.

8. O que é encapsulamento?

O encapsulamento permite ocultar os atributos de uma classe, por exemplo, para que dessa forma, apenas métodos (ou interface) possam alterar esses atributos de maneira específica, preservando a segurança e sentido do código.

9. O que é UML?

Resposta: UML é uma linguagem de modelagem que modela todo o sistema de um software

10. Como que um objeto envia uma mensagem para outro objeto?

Resposta: Através da herança, onde um objeto recebe as características de outro objeto.

11. Para que serve o especificador de acesso public? E o private?

Resposta: Os especificadores servem para indicar ao compilador quais métodos ou atributos são acessíveis ao usuário, dessa forma, o especificador public indica que o usuário pode utilizar os seus recursos, já o private ocorre um erro caso o usuário tente utilizar algum recurso dentro desse especificador.

12. O que são métodos set? E métodos get?

Resposta: O método set seria a função que atribui um dado para qualquer que seja o atributo, enquanto o método get seria a função que retornaria esse atributo ao usuário.

13. O que é um construtor-padrão?

Resposta: Construtor-padrão é aquele tipo de classe que, quando inicializada, não espera nenhum tipo de argumento, caracterizando um objeto com construtor-padrão.

14. Quando um construtor é invocado?

Resposta: Assim que o objeto é instanciado, caso o construtor não seja padrão, assim que o objeto for inicializado, deve ser passado os parâmetros necessários, que farão parte dos atributos do objeto.

15. Por que é importante separar a Interface da Implementação? Como podemos fazer isso em C++?

Resposta: Um dos motivos que torna esse comportamento importante seria para fazer com que o usuário não veja como foi feita a implementação, mas sim usufrua do método a partir da interface, além disso, é importante fazer essa separação para que não ocorra o acesso dos atributos de maneira errada, sendo apenas possível que o usuário “mude” os atributos a partir dos métodos.

16. Modifique a última versão da **Classe Carro** desenvolvida na “sala” (presente no slide) de modo que:

- Inclua um segundo membro de dados int que representa o ano do carro;
- Forneça uma função set para alterar o ano e uma função get para recuperá-lo;
- Modifique o construtor para aceitar dois parâmetros: um para a marca e outro para o ano;
- Modifique o método displayMessage para que ele apresente na tela a marca e o ano do carro.

Resposta: [main.cpp - Questão16 - Replit](#)

17. Crie uma Classe chamada **ContaBanco** que um banco poderia utilizar para representar contas bancárias dos clientes. Sua classe deve incluir um membro de dados do tipo **double** para representar o **saldo da conta**. Sua Classe deve fornecer um **construtor** que recebe um saldo inicial e o utiliza para inicializar o membro de dados. O construtor deve validar o saldo inicial para assegurar que ele seja maior ou igual a zero. Caso seja menor que zero, o construtor simplesmente deverá setar o saldo para zero. A classe deve fornecer três **funções-membro**. A função-membro *creditar* deve adicionar uma quantia passada como argumento ao saldo atual. A função-membro *debitar* deve retirar a quantia passada como argumento da conta. Se houver tentativa de retirar um valor acima do saldo, então o saldo deverá permanecer inalterado e a função-membro deverá exibir uma mensagem de erro na tela. A função-membro *getSaldo* deve retornar o saldo atual. Escreva um programa que cria dois objetos **ContaBanco** e testa suas funções-membro.

Resposta: [main.cpp - Questão17 - Replit](#)

18. Crie uma Classe chamada **Empregado** que possui 3 **membros de dados** - um **nome**, um **sobrenome** e um **salário mensal**. Sua Classe deve ter um **construtor** que inicialize os 3 membros de dados. Forneça uma função set e uma função get para cada membro de dados. Se o salário mensal não for positivo, configure-o como 0. Escreva um programa de teste que demonstre as capacidades da classe Empregado. Crie dois objetos Empregado e exiba seu salário mensal. Em seguida, dê um aumento de 10% para cada um dos empregados e exiba novamente o salário mensal.

Resposta: [main.cpp - Questão18 - Replit](#)

19. Crie uma Classe Data que inclua 3 partes de informações como atributos - dia (int), mes (int) e ano (int). Sua Classe deve ter um construtor com três parâmetros que utilize os parâmetros para inicializar os 3 membros de dados. Para o propósito deste exercício, assuma que os valores fornecidos para o dia e ano estão corretos, mas certifique-se de que o valor do mês esteja no intervalo 1-12; se não estiver, configure o mês como 1. Forneça uma função set e uma função get para cada membro de dados. Forneça também uma função-membro mostrarData que exiba o dia, o mês e o ano separados por barras (/). Escreva um programa que demonstre as capacidades da sua Classe Date.

Resposta: [main.cpp - Questão19 - Replit](#)

20. Crie uma Classe chamada Cupom que uma loja de suprimentos de informática possa utilizar para representar um cupom de um item vendido na loja. Um Cupom deve incluir quatro membros de dados - id (string), descrição (string), quantidade (int) e o preço do item (float). Sua Classe deve ter um construtor que inicializa os 4 membros de dados. Forneça uma função set e uma função get para cada membro de dados. Além disso, forneça uma função-membro chamada de calculaCupom que calcula o valor total da nota e depois retorna esse valor. Se a quantidade não for positiva, ela deve ser configurada como 0. Se o preço do item não for positivo, ele deve ser configurado com 0. Escreva um programa para ilustrar as capacidades de sua Classe Cupom.

Resposta: [main.cpp - Questão20 - Replit](#)