










ECOP03 – Programação Orientada a Objetos


Primeira Lista de Exercícios – Prof. João Paulo R. R. Leite


 Explique a diferença básica entre as linguagens C e C++. Dê alguns exemplos de ferramentas disponíveis em C++ que não estão presentes na linguagem C.

02) Diga se a frase é verdadeira ou falsa:


-  () Todo código escrito em C++ precisa, obrigatoriamente, ser colocado dentro de uma classe.
-  () Quando houver opção, devemos sempre preferir utilizar funcionalidades da biblioteca-padrão ao invés de construir o próprio código.
-  () `<stdio.h>` é a biblioteca padrão de entrada e saída de dados do C++.
-  () Em C++, o programador pode definir seus próprios tipos de dados, ao invés de somente utilizar os tipos nativos como `int` e `float`.
-  () `double d2 {2.3};` é uma inicialização válida para um número real em C++11.
-  () `nullptr` é uma palavra reservada de C++11.
-  () Apenas `private` e `public` são especificadores de acesso em C++.

 **03)** Como funciona a estrutura de repetição range-for? Explique e dê um exemplo de sua utilização.

 **04)** Qual a diferença entre um ponteiro e uma referência? Explique textualmente e dê um exemplo de código.

 **05)** Uma equação quadrática é da forma $ax^2 + bx + c = 0$, onde $a > 0$. Escreva uma classe que modele uma equação do segundo grau e tenha os coeficientes a , b e c como membros de dados. A seguir, escreva uma aplicação para ler três números como os respectivos coeficientes (a , b e c), e chame funções-membro desta classe para computar o discriminante (Δ) e imprimir as soluções.

Dica: Use as seguintes regras: se o discriminante = 0, a raiz é única; se o discriminante < 0, não há solução real; finalmente, se o discriminante > 0, há duas soluções reais distintas.

 **06)** Escreva uma classe que modele um triângulo e tenha como membros de dados os valores de seus três ângulos (a , b e c). Escreva uma aplicação onde o usuário é capaz de entrar com os valores para os ângulos e chamar funções-membro da classe para verificar se os valores de ângulo são válidos (conseguem formar um triângulo), e se for, qual o tipo de triângulo formado: isósceles, escaleno ou equilátero.

07) Imagine que você será contratado para desenvolver um programa orientado a objetos para controle de caronas na universidade. É preciso definir um vocabulário para o sistema, contendo informações sobre veículos, passageiros, destinos e tudo mais. Que classes você criaria para tal sistema? Liste pelo menos seis classes. Escolha as três principais e liste os membros de dados e funções-membro que cada uma delas teria. Não é preciso escrever código.



08) Escreva uma classe que modele uma data. Ela deverá ter como atributos valores para dia, mês e ano e deve ter métodos que verifiquem se a data é válida e imprimam a data no formato: 6 de abril de 2016.

09) Inclua uma função membro na classe da questão anterior que receba uma quantidade de dias como parâmetro e retorne uma nova data, que é a data atual acrescida do número de dias passado. Teste com valores entre 1 e 1000.



10) Escreva uma classe para representar um funcionário, com membros de dados para armazenar o nome, as horas trabalhadas e valor da hora. Adicione uma função membro para calcular o salário, utilizando os valores de horas trabalhadas e valor da hora, e outro para imprimir suas informações na tela. Escreva um programa que use a classe escrita para criar três funcionários com nomes Marcos, Mateus e Lucas. Marcos trabalhou 20 horas a R\$3,50 por hora, Mateus trabalhou 40 horas a R\$5,50 por hora, e Lucas trabalhou 40 horas por R\$13,50 a hora. Por fim, imprima os salários e nomes dos três funcionários.



11) No que consiste o conceito de encapsulamento?



12) Por que é importante definir uma boa interface pública para utilização da classe por terceiros?



13) Qual a diferença entre os especificadores de acesso *public*, *private* e *protected*? Quando cada um deles deve ser utilizado?



14) Existe alguma maneira de se acessar um membro *private* de uma determinada classe fora de seu escopo? Se houver, dê um exemplo em código.

15) Escreva uma função amiga (*friend*) da classe *Data* da questão 08 que imprima a data textualmente, como o seguinte exemplo: “seis de abril de dois mil e dezesseis”.



16) Escreva uma classe que modele um paciente com atributos que representem seu nome, peso e altura. A seguir escreva métodos para calcular seu índice de massa corporal (IMC), que é calculado pelo peso dividido pela altura ao quadrado, e para imprimir suas informações na tela (incluindo IMC). Crie, no programa principal, dois pacientes, inicialize seus dados e imprima as

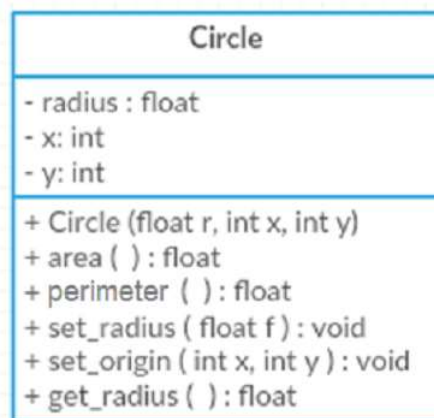
informações. Crie também uma função membro que indique se o paciente precisa fazer dieta ($IMC > 27$) e teste sua utilização no programa principal.



17) Crie uma classe `Conta_corrente` que modele dados bancários e possua os membros de dados nome, senha e saldo. Crie também os métodos `cadastrar_senha`, que recebe uma senha como parâmetro e altera o atributo senha, e o método `alterar_senha`, que recebe como parâmetro a senha antiga e a nova senha. Ele alterará o atributo senha apenas se a senha antiga for correta. Os métodos `debitar` e `creditar` também deverão ser desenvolvidos, e receberão como parâmetro o valor a ser debitado ou creditado e a senha. Eles só deverão efetuar a operação caso a senha for correta. Observe as regras de encapsulamento e escreva um programa de teste para a classe.



18) Escreva o código para a classe `Circle` abaixo com todos os seus membros de dados e funções membro. Não deixe de criar um arquivo para o cabeçalho (.h), um para a implementação (.cpp) e um para teste (.cpp).



19) Qual a função dos Include Guards?



20) O que é o ponteiro this? Que objeto ele referencia?



21) Escreva um programa que simule um jogo de tênis. Um jogo de tênis possui uma localidade, um tipo de piso, as pontuações de dois jogadores e os dois jogadores em si. Cada jogador possui seus próprios atributos, que são nome, o seu ranking atual, e o país que representa. Faça métodos de acesso aos atributos, métodos de impressão e um método que retorna o tenista favorito para ganhar o jogo, baseando-se no ranking atual de cada um.


22) Escreva um programa que possua as seguintes classes:


- Uma classe `Pessoa` com atributos nome (string) e sobrenome (string). A classe `Pessoa` ainda deve ter o método `nome_completo`, que retorna a concatenação do atributo nome com o atributo sobrenome. Além

disso, a classe deve possuir um construtor que recebe como parâmetros o nome e o sobrenome e os armazena nos atributos nome e sobrenome.

- b) Uma subclasse de Pessoa, chamada Funcionario. A classe Funcionario deve ter os atributos matricula (int) e salário (double). Considere que o método modificador `set_salario` somente atribui ao atributo salário o valor recebido como parâmetro se este valor não for negativo. Todo funcionário recebe seu salário em duas parcelas, sendo 60% na primeira parcela e 40% na segunda parcela. Assim, escreva os métodos `primeira_parcela`, que retorna um número real com o valor da primeira parcela do salário (60%) e `segunda_parcela`, que retorna um número real com o valor da segunda parcela do salário (40%).
- c) Uma subclasse de Funcionário, chamada Professor. Todo professor recebe seu salário em uma única parcela. Assim, deve-se sobrescrever os métodos `primeira_parcela` e `segunda_parcela`, sendo que o primeiro deve retornar o valor integral do salário do professor e o segundo retorna o valor zero.

Por fim, crie funcionários e professores fictícios no programa principal e teste cada um dos métodos criados.

 Defina um tipo Ângulo através de uma classe e sobrecarregue os operadores `==`, `!=`, `<`, `>`, `<=`, `>=`, `+`, `-`, além dos operadores de impressão e aquisição de dados `<<` e `>>` e da negação (`!`), que retorna o complemento do ângulo dado.

 Defina um tipo de dados que modele uma fração. Este tipo, ou classe, deverá conter como membros de dados o numerador e denominador de uma fração, do tipo inteiro, e deve implementar a sobrecarga do maior número de operadores matemáticos possível (`+`, `-`, `*` e `/` são obrigatórios). Sobrecarregue também operadores relacionais, para que seja possível realizar comparações entre funções (`<`, `>`, `==` são obrigatórios).