



Lab Avaliativo I de ECOP13A - Programação Orientada a Objetos

Nome:		Nº	
--------------	--	-----------	--

Duração aprox. 1:50 h.

Data: 23/05/2025

Instruções:

- Compactar os arquivos de código fonte e da prova para enviar na tarefa aberta no SIGAA.

1ª Questão. (35pts):

Com base na **questão 9** do **lab1** e a classe **Inteiro Longo** com operadores sobrecarregados da **questão 3** do **lab5**:

- Implemente uma função membro que verifique se o objeto da classe **Inteiro Longo** é um palíndromo.
- Implemente uma função membro que verifique se o número **Inteiro Longo** possui uma sequência de com três dígitos iguais, desconsidere os zeros a esquerda.
- Sobrecarregue as formas pré-fixada e pós-fixada do operador **--** para a classe **Inteiro Longo**.
- Sobrecarregue pelo menos um operador **-=** para a classe **Inteiro Longo**.
- Utilizar as funções membro criadas na letra a) e b), em um programa que lê um vetor com 4 objetos **Inteiro Longo**, e verifica se cada um dos elementos lidos pelo usuário são palíndromos, e se possuem 3 dígitos iguais repetidos.
- Utilize os operadores criados em um programa de teste.

2ª Questão. (35pts):

Com base na **questão 1** do **lab6**, a classe **polinômio**,

- Implemente os operadores de comparação (**>**, **<**, **>=** e **<=**) considerando somente a ordem do polinômio, como critério de comparação.
- Implemente os operadores de comparação (**==** e **!=**).
- Implemente o operador **float**, retornando o valor do polinômio para **x=0**;
- Altere o operador de saída **<<** para não imprimir o sinal de **+** caso o coeficiente seja negativo.
- Utilize os operadores criados em um programa de teste.

3ª Questão. (15pts):

Declarar uma classe para representar um objeto *Prova Bimestral* (apenas o .h). Acrescente pelo menos quatro atributos e três métodos diferentes dos construtores e destrutor para a classe a ser criada.



4ª Questão. (15pts) Diga se a frase é verdadeira ou falsa. Caso seja falsa destacar o trecho que a torna falsa.

- a() Em C++ quando se altera um atributo `static` de uma classe, esse valor fica disponível somente para o objeto que realizou essa alteração.
- b() Em C++, o programador pode definir seus próprios tipos de dados, ao invés de somente utilizar os tipos nativos como `int` e `float`.
- c() Em C++ é possível sobrecarregar os operadores de atribuição como funções *membro*.
- d() `double d2 {2.3};` não é uma inicialização válida para um número real em C++11.
- e() Os componentes de uma descrição de classe são: Declaração de Atributos, Declaração de Métodos e Implementação de Métodos.
- f() O construtor de cópia é utilizado na passagem de parâmetros de objetos do tipo de uma classe por valor em funções.
- g() Em orientação a objetos, criar um objeto significa definir uma variável do tipo de uma classe.
- h() Sobrecarregar um operador significa definir uma nova funcionalidade para aquele símbolo na linguagem, sem alterar as propriedades que o compilador conhece daquele símbolo.
- i() O construtor de inicialização serve para copiar um objeto existente para um outro que está sendo criado.
- j() Todo código escrito em C++ precisa, obrigatoriamente, ser colocado dentro de classes.
- k() Somente `private`, `protected` e `public` são especificadores de acesso em C++11.
- l() A propriedade da composição é utilizada para criar novos objetos a partir de outros previamente existentes.
- m() Funções *friend* são utilizadas para ter acesso privilegiado aos membros de uma classe.
- n() Para uma classe denominada `CNomeClasse`, a sintaxe para declarar um construtor de cópia é:

```
CNomeClasse ( const CNomeClasse );
```

- o() O modelo de sintaxe para declarar uma função `template` é:

```
template <typename T> tipoRetorno NomeFunc( Parametros ){ .... }
```