

Lista 2 – PP

O objetivo desta lista é auxiliar os alunos a fixar o conteúdo e ampliar seus conhecimentos. Não vale nota, porém é recomendado que a façam.

Para efeito de aprendizado do conteúdo de sobrecarga de operadores e templates, esta lista utiliza os códigos criados na lista1.

Implemente uma aplicação simples para testar as funcionalidades implementadas (não precisa realizar entrada pelo usuário, apenas crie objetos, chame os métodos implementados e imprima as saídas resultantes). Não utilize entrada e saída dentro de classe que não são especificar para isso.

1- Usando o código do ex1 da lista1, faça:

- a- Implemente o operador de atribuição e construtor de cópia
- b- Implemente os operadores de soma, subtração, divisão e multiplicação. Estes operadores devem ser aplicados no atributo first dos dois objetos e second dos dois objetos (Ex: obj1(first: 10, second: 2), obj2(first: 3, second: 5), obj1+obj2(first: 10+3, second: 2+5).
- c- Implemente os operadores de incremento e decremento pré e pós-fixados. De forma análoga ao item b, aplique o incremento (ou decremento) em first e second.
- d- Implemente os operadores == e !=. Dois objetos são iguais se first e second, de ambos os objetos, são iguais (são diferentes caso contrário).

2- Usando o código do ex2 da lista1, faça:

- a- Implemente o operador de atribuição e construtor de cópia
- b- Implemente os operadores de soma e multiplicação.

3- (recomendo olhar a questão 4 da lista 1) Implemente um template para representar um arranjo ordenado (para tipos que possuem ordem), que comporte no máximo 10 elementos. Implemente funções para:

- a- busca um elemento no arranjo (implemente usando busca binária)
- b- inserir um elemento (inserção ordenada).
- c- remover o item por valor. Nesse caso, o método deve retornar um booleano: true quando o elemento for encontrado e false caso contrário

4- Usando o código do ex5 da lista1, faça:

- a- Implemente o operador de atribuição e construtor de cópia
- b- Implemente o operador [], para retornar o i-ésimo bit, permitindo a alteração do mesmo.
- c- Implemente os operadores: shift a esquerda(<<), shift a direita (>>), ou exclusivo (^), e bit a bit (&) e ou bit a bit (|).
- d- Implemente o operador de casting (char) e um construtor capaz de inicializar um objeto da classe a partir de uma variável do tipo char (casting de char para sua classe).

5- Idem ao ex 3, para um número qualquer de elementos. Adote a política de dobrar o arranjo quando, na tentativa de inserir um elemento, o arranjo estiver cheio. Reduzir pela metade o tamanho do arranjo, quando na remoção, tivermos metade do arranjo sem elementos.