

NORMAS PARA ENTREGA E AVALIAÇÃO DO TRABALHO 1

1. O trabalho consiste na implementação de uma Classe de números complexos conforme arquivos básicos;
 - a. A estrutura básica do código está disponível no SIGAA e contém: *main.cpp*, *complexo.h* e *complexo.cpp*;
 - b. Os atributos e métodos que definem um número complexo estão escritos na classe *Complexo*, que está disponível nos arquivos *complexo.h* e *complexo.cpp*;
 - c. A classe *Complexo* é composto pelos atributos: “a” como a parte real, e “b” como a parte imaginária, da seguinte forma:
$$a + b\sqrt{-1}$$
 - d. A classe *Complexo* é composta por diversos métodos.
 - e. Os atributos e métodos que serão utilizados para compor a nota são classificados como “Definição Base” (ver arquivo *complexo.h*) e, portanto, não devem ter protótipo alterado.
 - f. Os alunos podem adicionar novos atributos ou métodos. Se o fizer, adicionem no espaço alocado como “Definição complementar”
 - g. O aluno deve:
 - i. Criar makefile completo;
 - ii. Em *complexo.cpp*, implementar:
 1. Construtor *Complexo(Complexo const &i_)*: Constrói um complexo a partir de outro.
 2. Construtor *Complexo(float a_, float b_)*: Constrói um complexo a partir da parte real “a_” e da parte imaginária “b_”;
 3. Método *pow(int i_)*: calculado como a potência do número complexo elevado ao parâmetro inteiro “i_”, ie, *complexoⁱ*;
 4. Sobrecarga de:
 - a. +: sobrecarga do operador SOMA;
 - b. -: sobrecarga do operador SUBTRAÇÃO;
 - c. *: sobrecarga do operador MULTIPLICAÇÃO;
 - d. /: sobrecarga do operador DIVISÃO;
 - e. =: sobrecarga do operador de ATRIBUIÇÃO;
 - f. ==: sobrecarga do operador de comparação de IGUALDADE;
 - g. !=: sobrecarga do operador de comparação de DIFERENÇA;
 - iii. Não modificar código algum em:

1. Construtor *Complexo()*;
 2. Método *imprimir()*;
 - iv. Liberar as variáveis alocadas dinamicamente no destrutor *~Complexo()*;
 - v. Utilizar *main.cpp* para realizar os testes.
 1. O código disponível no SIGAA já possui um exemplo fictício dos testes;
 - h. O aluno não deve:
 - i. Plagiar ou copiar código alheio;
 - ii. Utilizar bibliotecas que resolvem o problema;
 - iii. Modificar o construtor *Complexo()* e o método *imprimir()*;
2. A entrega do trabalho final é considerada completa com:
- a. Entrega da implementação contendo os arquivos
 - i. *makefile*;
 - ii. *main.cpp*;
 - iii. *complexo.cpp*;
 - iv. *complexo.h*;
 - b. Código compilando completamente através do comando *make*;
3. A nota será composto pela execução de diversos testes no *main.cpp*, tal como no código exemplo disponibilizado no SIGAA.
- a. Será contabilizado como sucesso cada teste bem sucedido;
 - b. A nota final será atribuída entre 0 a 10,0 considerando a proporção dos testes bem sucedidos em relação ao total de testes, ou seja,
- $$Nota = 10 * \frac{\text{Testes bem sucedidos}}{\text{Número total de testes}}$$
- c. Será considera nota 0,0:
 - i. O não cumprimento do que é exigido em 2.a e 2.b;
 - ii. Submissão dos arquivos por outro meio diferente do campo específico no SIGAA;
 - iii. Submissão dos arquivos fora do prazo estipulado;
 - iv. Erros de execução durante os testes;
 - v. Alterar atributo(s) ou protótipo de método(s) classificado(s) como “Definição Base”;
 - vi. Alterar implementação do construtor *Complexo()* e do método *imprimir()*;
 - vii. Plágios ou cópias;
 - viii. Utilizar bibliotecas que resolvem o problema;
4. Exceções e casos omissos devem ser tratados diretamente com o Professor da disciplina.