

OOP Lab 4

Topic: Operator overloading (arithmetic operators)

Marks: 25

Release Date: 15-Oct-20, Thursday

Submission Date & Time: 17-Oct-20, 11:59 PM, Saturday

You may send your lab queries at: asktoknow42@gmail.com

Problem 0:

Overload * operator for matrix multiplication as discussed in lecture 9 on Thursday, October 15, 2020.

(NOTE: The score of this problem will replace/improve the score of matrix multiplication in lab 2)

Problem 1: (25)

Create a class called **Complex** for performing arithmetic with complex numbers. Write a program to test your class. Complex numbers have the form

realPart + imaginaryPart * i , where i is $\sqrt{-1}$

Use integer variables to represent the private data **realPart** and **imaginaryPart** of the class.

Provide following public member function:

- a) Provide a constructor that enables an object of this class to be initialized when it is declared. The constructor should contain default values in case no initializers are provided. **(0 points)**
- b) **Complex(const Complex&)** Copy constructor. **(5 points)**
- c) **Complex operator + (const Complex& c)** Adding two Complex numbers: The real parts are added together and the imaginary parts are added together and thus returning a new complex number. **(5 points)**
- d) **Complex operator - (const Complex& c)** Subtracting two Complex numbers: The real part of the right operand is subtracted from the real part of the left operand, and the imaginary part of the right operand is subtracted from the imaginary part of the left operand. **(5 points)**
- e) **Complex operator * (const Complex& c)** Multiplying two Complex numbers. Multiplication of two complex numbers (a+bi) and (c+di) is defined as (ac – bd) + (bc + ad)i **(5 points)**
- f) **double mod()** Returns Modulus of a complex number. Modulus of complex number is found by first squaring the realPart and the imaginaryPart each. The resulting values are then added together and then square root is taken. (Use sqrt method from math.h library) **(5 points)**
- g) **void print()** Printing Complex numbers in the form (a, b), where a is the real part and b is the imaginary part. **(0 points)**

Now write a driver program to test your **Complex** class. Instantiate several objects. Test that all your member functions work properly.