**Lab 5: More Practice with C++ Classes**

Overview

In this lab you will complete a partially-written C++ class, writing the following components:

1. Non-default constructor
2. An overloaded function

In addition, you will write a program that uses the new class.

Class Description

In this lab you will complete a class to handle complex numbers. The class has data members to hold the real and imaginary parts of the number, both of which are of typedouble.

To complete the Complex class, you must add

1. A non-default constructor that creates a complex number given its real and imaginary parts.
2. Two overloaded Multiply() functions:
   * Multiply a complex number by another complex number.
   * Multiply a complex number by a double.

You will create a source file lab5.cpp to contain your main() function. The class declaration and definitions must be in the files Complex.h and Complex.cpp, respectively.

Non-default Constructor  
A complex number consists of real and imaginary parts. The imaginary part is a real multiple of the imaginary unit i, the square root of -1. In the Complex class, the complex number is stored as a real part (in m\_real) and an imaginary part (inm\_imag).

To begin, you need to copy the skeletal program files (lab5.cpp, Complex.h, andComplex.cpp) to your working directory:

cp /afs/umbc.edu/users/c/m/cmarron/pub/cmsc202/lab5.zip .

unzip lab5.zip

In Complex.cpp, implement the non-default constructor consistent with the declaration in Complex.h.

Overloaded Multiply() Functions

Now you will write two Multiply() functions.

The product of two complex numbers a + b i and c + d i is the complex number(a\*c - b\*d) + (a\*d + b\*c) i, that is, the complex number with real part a\*c - b\*d and imaginary part a\*d + b\*c. Write a version of the Multiply() function to multiply a Complex number by a second Complex number. You will need to add the declaration to Complex.h and the implementaion to Complex.cpp.

Once you have completed the first Multiply() function, implement an overloaded function to multiply a Complex number by a double. The product of a complex numbera + b i and a real number r is (a\*r) + (b\*r) i.

Consider whether the Multiply() functions should be const.

The main() Function

Now that you have completed the constructor and Multiply() functions, you just need to complete the main() function in lab5.cpp. Start with the skeletal program provided and add the required elements. The program must

1. Create at least three Complex numbers using the non-default constructor.
2. Demonstrate that both versions of the Multiply() function work correctly by computing products and printing them to the screen using the providedToString() function.

Building your Program

You can build your program with the following command:

g++ -ansi -Wall lab5.cpp Complex.cpp -o lab5

There are really two distinct steps in the program build:

1. lab5.cpp and Complex.cpp are *compiled* to produce the object files lab5.o andComplex.o. Object files consist of machine-readable code, but are not executable from the command line.
2. The objects files are *linked* to produce the executable lab5.

We can execute these separate steps explicitly as follows:

g++ -c lab5.cpp

g++ -c Complex.cpp

g++ lab5.o Complex.o -o lab5

Understanding this explict form of the build process is necessary to be able to write makefiles. Complete the following makefile, which uses the explicit build steps.

CPPFLAGS = -ansi -Wall

lab5: lab5.o Complex.o

g++ lab5.o Complex.o -o lab5

lab5.o: lab5.cpp Complex.h

WHAT GOES HERE?

Complex.o: WHAT GOES HERE...

AND HERE?