CSCE 240 HOMEWORK 2 DUE: 15 FEB 2017

Given the Computational Geometry which consistently appears in my research, I have need of a complex number class. I would like you to create such a class for my use. Below I describe my requirements. Keep in mind that if your interface does not match what I have described, my program will not compile.

This assignment is worth 10 Points

The class should behave as follows:

- **constructor 1**: The default constructor sets both the real and imaginary component members to 0.0.
- **constructor 2**: Accepts a double value which is assigned to the imaginary component member. The real component member is set to 0.0.
- **constructor 3**: Accepts two double values for the real and imaginary component members. The first parameter should be assigned to the real and the second parameter assigned to the imaginary member.
- **real**: The overloaded method should return the real portion of the Complex instance or set it if a double parameter is passed.
- **imag**: The overloaded method should return the imaginary portion of the Complex instance or set it if a double parameter is passed.
- **Add**: Class method accepts one parameter and should be overloaded to accept a Complex class instance, double, or int. The method should always return a Complex class instance—the sum of the parameters.
- **operator+**: Class method should be overloaded to allow for Complex class objects to be added to each other or to ints or doubles. Be mindful that an int or double could appear as a left-hand or right-hand side argument to the **+ operator**. In any case, the return value should be a Complex class instance representing the sum of the two sides.
- **Mul**: Class method accepts one parameter and should be overloaded to accept a Complex class instance, double, or int. The method should always return a Complex class instance—the product of the parameters.
- **operator***: Class method should be overloaded to allow for Complex class objects to be multiplied by each other or by ints or doubles. Be mindful that an int or double could appear as a left-hand or right-hand side argument to the *** operator**. In any case, the return value should be a Complex class instance representing the product of the two sides.
- **ToString**: This class method returns a string of the represented complex number as follows:
  - (2 + 2i) in the base case
  - (1.71 + 3.14159i) in the case of a non-whole floating point
  - (2 + i) in the case the imaginary portion is 1.0
  - (2 - 2i) in the case that the imaginary portion is ¡ 0.
  - (2) in the case that the imaginary portion is 0.
  - (2i) in the case that the real portion is 0.
- **operator<<**: This function (friend or not) shall accept an ostream object and Complex object and add the complex number represented to the ostream object. The function should correctly accept and return the ostream object so that it behaves as cout.

- **ToComplex**: This static class method shall accept a string of the type described in ToString and return a Complex class instance
- **operator>>**: This function (friend or not) shall accept an istream object and Complex object. It shall extract a string of the type described in ToString from the stream and return a complex number. The function should correctly accept and return the istream object so that it behaves as cin.

To get credit, you must upload a zipped (not tarred, gzipped, or 7zwhateveritscalled) archive containing your submission files in the the directory tree:

- **username/hw/02**.

The archive file should be named:

- **username.zip.**

In both cases, the username should be your Blackboard login name. Your two files should be named:

- **complex.h**, and
- **complex.cc**.

The End.