**Lab Assignment 11**

**CS 1021C-002 – Spring 2014 Annexstein**

## Due: Try to complete this lab assignment and submit it via CascadeLMS during the lab session on Wednesday, April 2. If you do not submit it during the lab session, you must submit it via CascadeLMS by the day of the next lab on Wedneesday, April 9.

## Objective: Gain more experience and practice with classes and objects, including constructors and operator overloading.

### Problem Description: You are to implement the complex number ADT (Abstract Data Type). As a guide, use the rational number ADT that was discussed in class and has been uploaded to Documents on Cascade (file rat.cpp).

**Task 0.**

Understand the complex number ADT. Data consists of two floating point numbers (e.g., of type double) a and b representing the real and imaginary parts of the complex number a + ib. The symbol i represents the square root of -1, or equivalently the square of i is -1. Operations are:

Addition:

(a+bi) + (c+di) = (a+c) + (b+d)i.\ 

Subtraction:

(a+bi) - (c+di) = (a-c) + (b-d)i.\ 

Mulitplication:  
(a+bi) (c+di) = (ac-bd) + (bc+ad)i.\   
Division:

\,\frac{a + bi}{c + di} = \left({ac + bd \over c^2 + d^2}\right) + \left( {bc - ad \over c^2 + d^2} \right)i. 

Convince yourself that these operations are correct using the fact that i2 = -1.

**Task 1.**

Write the C++ class Complex for the complex numbers and its implementation, using the class Rational and its implementation (see rat.cpp in Documents) as a guide. Implement the operations of addition and multiplication and overload the operators + and \*. Also overload the insert operator << and the extract operator >> (using rat.cpp as a guide).

In your main program have the user input two complex numbers, i.e., two pairs of doubles representing the real and imaginary parts of the complex number, and output their sum and product.

**Task 2.**

Empirically test the famous formula of Euler:

eiπ - 1 = 0.

Compute eiπ using objects from your Complex class and employing the following approximation (first few terms of Tayler series) for ez :

ez ≈ 1/0! + z/1! + z2/2! + z3/3! = 1 + z + z2/2 + z3/6.

### Task 3. (extra credit)

Implement the operations of subtraction and division, overloading the operators – and /.