Math 3300 Programming Assignment 9

Instructions: Work on the following 2 programs and submit your source code to me via Blackboard. Send me 2 cpp files.

1. Create a **class** named **fraction** which will represent a new data type corresponding to fractions. Your class should create the following private members:

   - Integers **n** and **d** corresponding to the numerator and denominator of the fraction
   - A function **gcd** which returns the greatest common divisor of 2 integers (to help you reduce the fraction).

   And the following public member functions:

   - 3 **constructors** which will create any fraction when declared with 2 integers, will create $a/1$ when declared with 1 integer, and will create $0/1$ when declared with no integers.
   - **num** which returns the numerator of the fraction
   - **denom** which returns the denominator of the fraction
   - **reduce** which reduces the fraction to its lowest terms (using the gcd function above)
   - **convert** which converts the fraction to a decimal (a double)

   You are to overload the following operators: $+, -, *, /, += , -=, *=, /=, <, <=, >, >=, ==, !=$ for use with fractions. For the operators $+, -, *, /, += , -=, *=, /=$, calculate the new fraction in lowest terms (like 3/4 instead of 6/8). Overload + and − for both unary and binary operations.

   Also overload << and >> to accept and display fractions in the form a/b. A fraction like $a/1$ should be displayed as a. You can assume that a fraction will always be input in the form a/b.

2. Create a **class** named **complex** which will represent a new data type corresponding to complex numbers. Your class should create the following private members:

   - Real numbers a and b corresponding to $a + bi$.

And the following public member functions:

- 3 **constructors** which can create any complex number when declared with 2 doubles, will create $a+0\imath$ when declared with one double, and will create 0 (which is $0+0\imath$) when declared with 0 doubles.
- **real** which returns the real part of the complex number (a).
- **imag** which returns the imaginary part of the complex number (b).
- **conjugate** which performs the conjugate of the complex number ($a+b\imath$ becomes $a-b\imath$, i.e. the sign of b changes).
- **modulus** which returns the modulus of the complex number ($\sqrt{a^2+b^2}$).

You are to overload the following operators: $+, -, *, /, +=, -=, *=, /=, ==, !=$ for use with complex numbers. Overload + and − for both unary and binary operations.

As an added requirement, overload the operators << and >> to accept and display complex numbers in the form $a+b\imath$ or $a-b\imath$ (depending on the sign of b).

For a review of how to add/subtract/multiply/divide complex numbers, look at the Wikipedia article on complex numbers.