Lab 6

Due Date: Mar 7, 2019
Total Points: 15 points

The purpose of this lab is to practice operator overloading and friend functions.

Consider the RationalNumber class declaration below. You need to write four functions:

- A **stand alone friend function** that **overloads** the << operator and prints a rational number in the form numerator/denominator.
- A **member function** that **overloads** the subtraction operator (-) for the RationalNumber class. The member function subtracts two RationalNumber objects r1 and r2 and assigns the result to object r3.
- A **standalone friend function** that **overloads** the postfix increment operator (++) for the RationalNumber class. The friend function adds one to a RationalNumber object and allows cascaded function calls.
- A **stand alone friend function** that **overloads** the >> stream extraction operator and prompts the user for rational number (numerator and denominator)

```
class RationalNumber
{

        // Make the standalone functions friends of the RationalNumber class

private:
        int numerator;          // private variable numerator
        int denominator;        // private variable denominator

public:
        RationalNumber( int = 0, int = 1 );        // default constructor

        // Include the prototype of the overloaded class method here

};
```

You may use and complete the following main program to test your code.

```
int main()
{
        RationalNumber r1( 7, 3 ), r2( 3, 9 ), r3, r4;

        // Call the overloaded subtraction operator to subtract r2 from r1
        // and store the result in r3

        // Call the << overloaded operator to display the rational number r3


        // Call the overloaded postfix increment to increment r1 and store it in r3
```

```
        // Call the  << overloaded operator to display the rational number r3
        // Call the >> overloaded operator to enter attribute values for r4
        // Call the  << overloaded operator to display the rational number r4
return 0;
}
```