

**CS-183**  
**C++ Program #2**  
**Simple Classes**

**Date Assigned:**      **Friday, March 15, 2002**  
**Date Due:**            **Friday, March 29, 2002**

A rational number is any quantity that can be expressed as  $i/j$  where both  $i$  and  $j$  are integers. Rational addition, subtraction, multiplication, and division are defined as follows:

$$\frac{a}{b} + \frac{c}{d} = \frac{ad + bc}{bd}$$
$$\frac{a}{b} - \frac{c}{d} = \frac{ad - bc}{bd}$$
$$\frac{a}{b} \times \frac{c}{d} = \frac{ac}{bd}$$
$$\frac{a/b}{c/d} = \frac{ad}{bc}$$

**Part 1 (Define Methods (i.e. Member functions) )**

Create a rational number class which implements the above operations using ordinary member functions (named: `add`, `sub`, `mult`, and `div`).

As always, a well commented source code implementation is required, with separate files for the declarations, definitions, and user application code. Be sure to demonstrate all aspects of your implementation in the test cases presented in the user program. Then, write a report that documents how you met the objectives of this assignment and also indicate how you tested your work to verify the results of your program. **Be sure to include your time reporting for the lab in your written report.**

**Part 2 (Overload Operators)**

Once Part 1 is successfully working, use operator overloading to implement the same arithmetic operations as `+`, `-`, `*`, and `/`. Complete the functional abstraction for your rational numbers by overloading the remaining operators `+=`, `-=`, `*=`, `/=`, `++`, `--` (both prefix and postfix forms), and overload the inserter (`<<`) and extractor (`>>`) operators so user programs can read and print rational numbers. Also, implement the logical (`&&`, `|`, and `!`) and relational operators (`<`, `<=`, `>`, `>=`, `==`, and `!=`) so that rational numbers can be compared in the same manner as the built-in data types of the language (i.e. make your Rational class a “first class” data type).

The same criteria applies to your Part 2 submission as for your Part 1 submission (above). Be sure to test all aspects of your implementation by demonstrating them in use in the user program. Also be sure to document how you went about designing & implementing your solution. Of course, keep track of your time spent on this work and report the results in your report.