Object-Oriented Programming
Assignment 3 – class Rational

April 20, 2017

Objectives

Practice and get familiar with classes in C++ language. In this assignment you will make use of the subject matters about Classes (ch. 6), Constructors and Other Tools (ch. 7), and Operator Overloading, Friends, and References (ch. 8).

Problem Description

Define a class `Rational` for *rational numbers*. A rational number is a number that can be represented as the quotient of two integers. For example, 1/2, 3/4, 64/2, and so forth are all rational numbers. The class `Rational` MUST

1. Represent a rational number as two values of type `int`, one for the numerator and one for the denominator.
2. Define three constructors:
   (a) Default constructor that initializes an object to 0 (i.e., 0/1).
   (b) A constructor with two arguments that can be used to set the member variables of an object to any legitimate values.
   (c) A constructor that has only a single parameter `wholeNumber` of type `int` and initializes an object to the rational number `wholeNumber`/1.
3. Define two *constant* accessor functions: getNumerator() and getDenominator().
4. Overload the input and output operators >> and << as *friend* functions. Numbers are to be input and output in the form 1/2, 15/32, 300/401, and so forth. Note that the numerator, the denominator, or both may contain a minus sign; so $-1/2$, $15/-32$, and $-300/-401$ are also possible inputs.
5. Overload operators +, -, *, /, ==, <, <=, >, >=, and [ ]:

Operator	Overloading	Calculation
+	non-member	
− (subtraction)	non-member	
− (negation)	non-member	
*	friend	
/	friend	
==	friend	$a/b == c/d$ if $ad == cb$
<	member	if $b > 0$ and $d > 0$, $a/b < c/d$ provided $ad < cb$
<=	member	
>	member	
>=	member	
[ ]	member	[0]: numerator; [1]: denominator

6. Need a function to *normalize* the values stored so that, after normalization, the denominator is positive and the numerator and denominator are as small as possible. For example, after normalization $4/-8$ would be represented the same as $-1/2$. 
7. With two rational numbers input by users, test all the above requirements and display respective calculation results for your class in the `main()` function.

Evaluation

- Correctness: 90%
- Styling: 10%

Submission

- **Due: 2017/05/03 (degrade by 10 points for each day delay)**
- Source code (*.cpp)
  - Show your information (Name, Student ID, Dept, Year) as comments in the beginning of your code.
  - Name your file as "Hw3_(#Student ID).cpp".
- Upload the file to eCourse.