CSS342 Data Structures, Algorithms, and Discrete Mathematics I

Autumn 2017

Practice Lab 1: Objects and Classes

Purpose

This programming assignment exercises how to construct abstract data types through implementing a **Rational** class in C++. It also reviews type conversions for classes, operator overloading, and input/output including the friend concept.

Overview of Rational Class

The **Rational** class presents rational number in **numerator** and **denominator**. For instance, Rational r**(4, 3)** means 4/3. The **Rational** class you will design should have the following features, which are additions to the existing Rational class in rat files.

Member functions
Provide the following two member functions:
getNumerator
returns the numerator data member of this Rational number.
getDenominator
returns the denominator data member of this Rational number.

Math operators
The class must implement binary arithmetic operations such as addition, subtraction, multiplication, and division.
Addition
Add two objects.
Subtraction
Subtract the 2nd object from the 1st one.
Multiplication
Multiply the left-hand-side object by the right-hand-side **Rational** and return a **Rational** object.
Division
Divide the 1st object by the 2nd one and return a **Rational**-type value. Division by zero returns the 1st object without change and print error message
Comparison
The class must implement **==**, **!=**, **<**, **<=**, **>**, and **>=**
Assignment
The class must implement **+=**
Stream I/O
The class must implement the **<<** and **>>** operators:

Input
Take two values as a **numerator** and an **denominator** value.

Output
The format will be: **X/Y**, where **X** and **Y** are a **numerator** and **denominator** value respectively. Of course, if **X** is 0, it should displayed as 0. (not 0/denominator). If **Y** is **0**, it should not be displayed (provide error message).

Statement of Work

Download rat.h rat.cpp and ratdriver.cpp files and modify programs so that rat.h files include the above functions described.

1. Make a lab1 directory and go to the folder

```
mkdir lab1
cd lab1
```

2. Download rat.h, rat.cpp, and ratdriver.cpp files

3. Add the specified functions described above in the rat.h, and rat.cpp.

4. Compile your code.

5. Use ratdriver2.cpp driver file and compile it with your new rat.cpp files.

```
g++ rat.cpp ratdriver2.cpp
./a.out
```

```
===================
List of functions
==================

int getNumerator();

int getDenominator();


friend ostream& operator<<(ostream&, const Rational&);


friend istream& operator>>(istream&, Rational&);

// arithmetic operators
Rational operator+(const Rational &) const; // add 2 Rationals

Rational operator-(const Rational &) const; // subtract 2 Rationals

Rational operator*(const Rational &) const; // multiply 2 Rationals

Rational operator/(const Rational &) const; // divide 2 Rationals

// division by zero terminates
// boolean comparison operators
bool operator>(const Rational &) const; // is object > parameter?

bool operator<(const Rational &) const; // is object < parameter?

bool operator>=(const Rational &) const; // is object >= parameter?

bool operator<=(const Rational &) const; // is object >= parameter?

bool operator==(const Rational &) const; // is object == parameter?

bool operator!=(const Rational &) const; // is object != parameter?
// assignment operators
Rational& operator+=(const Rational &); // current object += parameter
```