Laboratory 10: Chapter 9 Classes: A deeper look, Part 1
Programming Methodologies Lab (14:332:254)
April 9, 2018

Problem Definition (Exercise 9.6 of the Deitel book)

Create a class called Rational for performing arithmetic with fractions. Use integer variables to represent the private data of the class - the numerator and the denominator. Provide a constructor that enables an object of this class to be initialized when it is declared. The constructor should contain default values in case no initializers are provided. Provide public member functions that perform each of the following tasks:

- Adding two rational numbers.
- Subtracting two rational numbers.
- Multiplying two rational numbers.
- Print Rational numbers in the form a/b, where a is the numerator and b is the denominator

The implementation of the class rational is divided in two steps:

Part 1: Class Definition

Implement the class definition for class Rational in a file called Rational.h. The class definition should contain only the prototypes of the member functions Rational, setNumerator, setDenominator, add, subtract, multiply and print.
You also need to include the integer data members numerator and denominator.
Determine if the functions and the data members should be private or public.
You can find an example of the class definition for the class Time in the file Time.h – also available in Chapter 9.2 (page 379 of the Deitel book, ninth edition).

Part 2.1: Member-function definition

Implement the member-function definition of class Rational in a file called Rational.cpp. This .cpp file contains the actual implementation of the member functions of class Rational.
You can find an example of the member-function implementation in the file Time.cpp – also available in Chapter 9.2 (page 380 of the Deitel book).

Part 2.2: Test program

Write a program that tests your class Rational. Call your file testRational.cpp. This program should:

1. Instantiate two objects of class Rational

2. Output the objects’ values (using the function print)
3. Change the value for one of the objects and output the object’s new value
4. Add the two rational numbers and output the result
5. Subtract the two rational numbers and output the result
6. Multiply the two rational numbers and output the result

You can find an example of a test program for the class Time in the file testTime.cpp – also available in Chapter 9.3 (page 384 of the Deitel book).

Grading

Part 1: Class Definition	30%	Required
Part 2: Member-function definition and test program	70%	Required

Submission

Show your work to the TAs (even if you did not finish) before submitting your answers. You are required to submit via sakai one cpp file for each of the parts in this lab. Name your files:

```
Rational.h
Rational.cpp
testRational.cpp
```

Remember to include following information at the beginning of `testRational.cpp` file as a comment.

```
// Course Number and section: 14:332:254:XX
// Lab Instructor: Kazem Cheshmi
// Date Performed: 04/XX/2018
// Date Submitted: 04/XX/2018
// Submitted by: YOUR NAME, RUID
```

NOTE – You need to **Upload** your files and then **Submit** them. The TAs will have no access to your files if you forget to submit them.

If you do not submit your files to sakai, you will not receive a grade.

Software Copying Policy

If your lab assignment is found to be a copy of another student's lab, you will not be given credit for this assignment. If this happens more than once, you may be in jeopardy of failing the course/lab.

The Software Copying Policy can be found on sakai under “Course Content and Related Materials /Calendar & Course Information”.