Assignment #9 (Marked Lab)

Problem Solving and Programming in C++

Department of Computer Science

Old Dominion University

Objectives: In this assignment, students will write a program to overload different kinds of operators, learn how operators can perform differently to simplify the code, give special meaning to some operators, and make user-defined data types behave like the built in data types

Important notes:

- This assignment should be implemented during the lab time of this week.
- Any submission after the lab time will not be accepted for marking.
- Please note that this assignment only worth a 30 marks; (0.3 assignment).

Background: Operators are symbols that represent functions. For example, when we type:

```
int a, b, c;
a = b + c;
```

this is the same as:

```
a.operator= (b.operator+(c));
```

The `operator=` assigns everything that is passed to it (in parenthesis) to the variable that called it (`a`).

The `operator+` gets the value of the variable that is passed to it and adds it to the variable that called it (`b`).

While this makes sense to the compiler for standard data types, the compiler doesn't know how to handle these operators for structs and classes you implement. This is where operator overloading comes into play. You need to tell the compiler how to handle operators for your special data types. Keep in mind that some operators need to be implemented as member functions while others must be standalone functions. For example, operator< to compare two dates would be something like:

```
bool operator<(Date date2){
        if (y < date2.y)
                return true;
        else if (y == date2.y && m < date2.m)
                return true;
        else if (y == date2.y && m == date2.m && d < date2.d)
                return true;
        return false;}
```

Problem Description: (30 marks)

Task-1: Rational fractions are of the form a / b, in which a and b are integers and $b \neq 0$. In this exercise, by ‘‘fractions’’ we mean rational fractions. Suppose a / b and c / d are fractions. Arithmetic operations and relational operations on fractions are defined by the following rules:

Arithmetic Operations:

$$a/b + c/d = (ad + bc)/bd$$

$$a/b - c/d = (ad - bc)/bd$$

$$a/b \text{ x } c/d = ac/bd$$

$$(a/b)/(c/d) = ad/bc, \text{ in which } c/d \neq 0$$

Relational Operations:

$$(a/b<=c/d) = (ad<=bc)$$

$$(a/b>=c/d) = (ad>=bc)$$

$$(a/b==c/d) = (ad==bc)$$

$$(a/b!=c/d) = (ad!=bc)$$

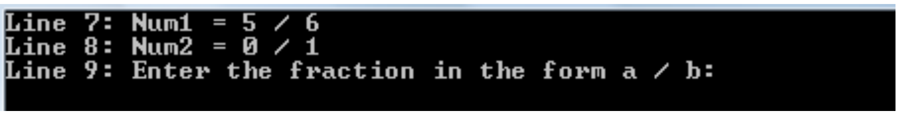
Fractions are compared as follows: a / b **op** c / d if ad **op** bc, in which **op** is any of the relational operations. For example, a / b < c / d **if** ad < bc. In this assignment, you will need to design a class— say, `fractionType` which performs the arithmetic and relational operations on fractions. Overload the arithmetic and relational operators so that the appropriate symbols can be used to perform the operation. Also, overload the stream insertion and stream extraction operators for easy input and output. Write a C++ program that, using the class `fractionType`, performs operations on fractions.

Among other things, test the following: Suppose x, y, and z are objects of type fractionType. If the input is 2/ 3, the statement: `cin >> x;` should store 2 / 3 in x.

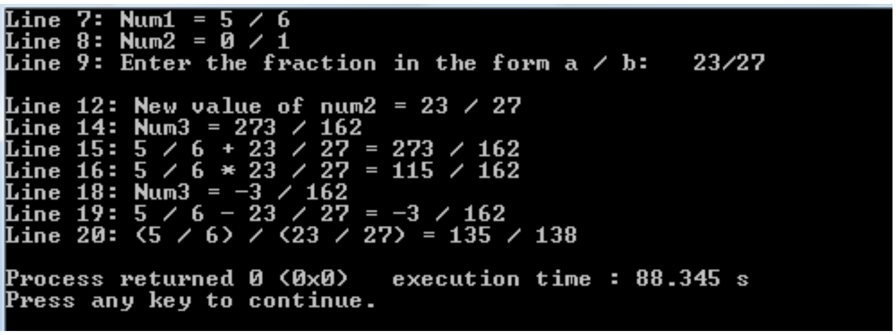
The statement: `cout << x + y << endl;` should output the value of `x + y` in fraction form.

The statement: `z = x + y;` should store the sum of `x` and `y` in `z` in fraction form.

You are given the main driver for this program “`MainProgram.cpp`”. A sample run of the program is provided below:



```
Line 7: Num1 = 5 / 6
Line 8: Num2 = 0 / 1
Line 9: Enter the fraction in the form a / b:
```



```
Line 7: Num1 = 5 / 6
Line 8: Num2 = 0 / 1
Line 9: Enter the fraction in the form a / b:   23/27

Line 12: New value of num2 = 23 / 27
Line 14: Num3 = 273 / 162
Line 15: 5 / 6 + 23 / 27 = 273 / 162
Line 16: 5 / 6 * 23 / 27 = 115 / 162
Line 18: Num3 = -3 / 162
Line 19: 5 / 6 - 23 / 27 = -3 / 162
Line 20: (5 / 6) / (23 / 27) = 135 / 138

Process returned 0 (0x0)   execution time : 88.345 s
Press any key to continue.
```

***Comment**: In the provided run, it was assumed that the user of the program will be asked to enter only one fraction which is the value for **num2**. The value of **num1** was always assumed to be **5/6**.*

Submission notes:

- Submit all files from your project including the .cpp file(s).
- Zip all files and name it as "Assg9-Task1_cslogin", where the cslogin is your login ID for the computers at the Department of Computer Science at ODU.
- Submit the zipped file in the respective Blackboard link.

Task-2: ***(Application – Ingredient Management System Process)***

Assume you were hired by the head chef at a big restaurant (like Freemason Abbey, USA family restaurant, Eleven Madison Park, Paramount, Shaya, *etc*.) to write a computer program to automate/expedite the ingredient management process. The chef needs to cook two similar dishes. These dishes use the same ingredients with slight differences to accommodate various dietary restrictions:

- Assume you have 3 tomatoes, a garlic, 2 onions, 1 butter bar, 2 lbs of ground beef and 2 lbs of beans.
- The first dish requires a half tomato, 3/4 of a garlic, 1/4 of an onion, 1/8 of a butter bar, 1/2 lbs of ground beef, 2/3 lbs of beans.
- The second dish requires 3/5 of a tomato, 1/5 of a garlic, 2/5 of an onion, 1/5 of butter bar, 7/10 lbs of ground beef, 9/10 lbs of beans.

The chef likes to keep the fridge stocked so you need to have a list of:

- The total amount of each ingredient that you will have left so the chef knows when to order more ingredients.
- Which of these two dishes is going to weigh more than the other (Disregard the weight of tomatoes, garlic, onions and butter) so the chef can change the prices accordingly.

In order to write this program, you could just type the formulas above to find the answers. Or you could create functions that will hold the formulas above and call those. However, the easiest way to do it would be to use the operators you overloaded for Task-1 in the fractionType module. For this assignment, you need to use the overloaded operators.

Submission notes:

- Submit all files from your project including the .cpp file(s).
- Zip all files and name it as “Assg9-Task2_cslogin”, where the cslogin is your login ID for the computers at the Department of Computer Science at ODU.
- Submit the zipped file in the respective Blackboard link.