Lab Assignment #9

This week's assignment will have you write a program that examines classes further and also introduces constructors. Feel free to work in pairs and ask for help early.

This program will require you to write a `class` called **`Fraction`** that represents a rational number. The class should represent a fraction internally with two private data members: a numerator and a denominator (assume whole numbers only). The public interface of the class must provide the following:

- A *default constructor* that sets the numerator and denominator to 1
- A *constructor* that accepts two parameters: a numerator and a denominator for the new `Fraction` instance. Your code must ensure that:
    - o The denominator is not zero (if it is, output an error and exit)
    - o The denominator is not negative (i.e., if the fraction is supposed to be negative, the numerator should be negative; if the parameters are both negative, the constructor should make the fraction positive)
- Two member functions, **`sumWith`** and **`multiplyWith`**, that perform addition and multiplication of this `Fraction` instance and another `Fraction` instance (so, each will have a single `Fraction` type parameter). The functions should not modify either `Fraction` instance; instead, they should *create and return* a new instance. The resulting `Fraction` instance does not need to be in reduced form (unless you're doing the extra credit).
- A member function called **`print`** that outputs the fraction. It should accept a single `ofstream` argument (by reference) and output the fraction details (without a newline). Here is an example (where N is the numerator value and D is the denominator value):
  `N/D`
- Extra credit: A member function called **`simplify`** that reduces the fraction to its lowest terms (12/15 => 4/5). If you do this, you should add a call of your `simplify` function within the parameterized *constructor* definition.

Your program will consist of three files:

- A header file named `fraction.h`
- An implementation file for the header file, named `fraction.cpp`
- A main program file named `main.cpp`

The main program, `main.cpp`, has been provided for you. You will use this file as-is, except for the addition of your preamble. It is your task to create the header and implementation files and then write the correct function prototypes and definitions to make the main work.

Submitting your work

Each of your source code files must have a preamble at the top (see the programming rules and guidelines document in Blackboard for more on the preamble).

Make sure you are in the folder with your source code files (use the `ls` command) and then run the following to create a zip archive of them:

```
$ mkdir lastname_firstname_lab09
$ cp lab09a.cpp lastname_firstname_lab09/lab09a.cpp
$ cp lab09b.cpp lastname_firstname_lab09/lab09b.cpp
$ zip -r lastname_firstname_lab09.zip lastname_firstname_lab09/
```

You'll need to change `lastname` and `firstname` to your actual last and first names (**use lowercase letters**) in the steps above. Once you have your zipfile, you can use that file as your submission for the assignment in Blackboard.

If you are working in pairs, then you should have both of your names included as comments in the source code file's preamble (see the programming rules and guidelines document in Blackboard for more on the preamble). Also, write both your names to the notes section in the submission form when submitting to Blackboard.

Sample output

```
$ ./lab09
1/2 + 2/3 = 7/6
1/2 * 2/3 = 2/6

-1/2 + -2/6 = -10/12
-1/2 * -2/6 = 2/12

1/2 + -3/5 = -1/10
1/2 * -3/5 = -3/10

3/5 + 2/5 = 25/25
3/5 * 2/5 = 6/25
```

Sample output (with reduction)

```
$ ./lab09
1/2 + 2/3 = 7/6
1/2 * 2/3 = 1/3

-1/2 + -1/3 = -5/6
-1/2 * -1/3 = 1/6

1/2 + -3/5 = -1/10
1/2 * -3/5 = -3/10

3/5 + 2/5 = 1/1
3/5 * 2/5 = 6/25
```