COSC 4301 – MODERN PROGRAMMING
Program 5 – Fractions Arithmetic

A fraction is a quantity which is not a whole number rather it is formed by using two numbers. A fraction has two parts; numerator and denominator separated by a "/". Examples of fractions are: 15/2, 35/6, and 1/2.

A fraction must always be represented in its reduced (or simplest) form where there are no common multiples of the numerator and the denominator. For example, 56/12 is not in its reduced form since its numerator and denominator still have common multiples. Its reduced form will be 14/3 (dividing both numerator and denominator by 4).

For reducing a fraction to its simplest form, Greatest Common Divisor (GCD) of its numerator and denominator is calculated. GCD is the largest number from the common multiples of numerator and denominator. Write a recursive method named **calculateGcd** to calculate the GCD. You should have studied this in your Data Structures class (COSC 2436).

Allow the user to perform the following:

- Add the fractions and display the results.
- Subtract the fractions and display the results.
- Multiply the fractions and display the results.
- Divide the fractions and display the results.
- Display the original fractions.
- Enter new fractions.

Create a class that depicts a fraction with two instance variables representing the numerator and the denominator of the fraction. The class must have a constructor and setters and getters for initializing the attributes of the class.

The class must contain these methods; addFraction, subtractFraction, multiplyFraction, and divideFraction which receives a Fraction object as an argument which acts as the second fraction. The first fraction is the object on which these methods are called.

The class must also contain a **toString()** method which returns the String version of the Fraction in the form of "numerator / denominator" of fraction.

Write a method named **displayImproper** to calculate the fractions as a whole number and the fraction part.

Write a public test class, named Program5.java to test your other classes. No input, processing or output should happen in the main method. All work should be delegated to other non-static methods.

All classes in this program must be public, non-static and not nested in other classes. Do not use more than three classes to complete this program.

Every method in your program should be limited to performing a single, well-defined task, and the name of the method should express that task effectively. All methods should be non-static unless it is absolutely necessary for it to be static.

Use the modern programming techniques you have learned so far in the class to design and complete this program.

Allow the user to run the program as many times as possible until zeros have been entered for the numerator and denominator.

Run your program and copy and paste the output to a file named **Program5-Output.txt.** Create a folder named, **<YourLastNameFirstName>_Program5**. Copy your source codes and the output file to the folder. **Zip the folder, as a ".zip" file, and upload it to Blackboard**.

Before you upload your program to Blackboard:

- Ensure that your code conforms to the style expectations set out in class and briefly discussed below.

- Make sure your variable names and methods are descriptive and follow standard capitalization conventions.

- Put comments wherever necessary. Comments at the top of each module should include your name, file name, and a description of the module. Comments at the beginning of methods describe what the method does, what the parameters are, and what the return value is. Use comments elsewhere to help your reader follow the flow of your code. **See the ProgramTemplate.java file for more details**.

- *Program readability and elegance are as important as correctness.* After you have written your method, read and re-read it to eliminate any redundant lines of code, and to make sure variables and methods names are intuitive and relevant.

Read the assignment very carefully to ensure that you have followed all instructions and satisfied all requirements.

You will not get full credit for this program if it is not written as instructed even if it works as expected.