Lab Practice Week 4

You need to show working versions of your answers to all questions to your tutor. Your tutor will expect to see them by your next session.

What to submit: your answers to exercises 1, 2, 6, 7, and 8.

Note: even though you only need to submit those exercises mentioned above, you should attempt all exercises in each lab practice to help broaden your understanding; this may also help with your assignment work.

Do all the programs in NetBeans IDE.

NOTE: Include internal documentation in your code, if you are not sure, read chapter 2.4 in your textbook and talk to your lecturer

---

This week exercise builds on the Fraction class which you produced for exercises 1, 3, 4, and 5 in last week's lab practice.

Before starting the exercises, make sure you have read the relevant material.

1. Add another public method called *add()* to your Fraction class. This method adds another fraction to the "calling object". Thus, the method will take a Fraction class object as a parameter, add this parameter fraction to the calling object (fraction), and return a Fraction object as a result. HINT: we can use cross multiplication to determine the numerator of the resultant Fraction. The denominator of the resultant Fraction is simply the multiplication of the denominators of the two other Fractions.

2. Write a client program that loops around getting two fractions from the user, output their sum, and state whether or not they are equal. Keep looping until the first fraction entered is a zero fraction.

3. Update your UML diagram from last week to include the added behaviour of your Fraction class.

4. Add another public method *dblValue()* to your Fraction class which returns the double precision approximation value of the fraction. That is, the floating point result of actually dividing numerator by denominator. N.B. this method does **not** do any display itself, but can be called by a client program to be used in an output statement. Eg: if a client has a fraction frac that represents 1 / 2, then a method call to frac.dblValue() should return the double number 0.5. Use your client program to test this functionality; i.e. provide an output statement to display the double value of a fraction.

5. Add a *toString()* method to your Fraction class that returns the fraction as a String in the form "x / y", where x and y are numerator and denominator respectively. This is similar to your display method, but it does not actually do any output itself; it only returns the Fraction as a String -- nothing else. N.B. as the method does **not** do any display itself, the output can be done by a client program that calls the method in an output statement. Use your client program to test this functionality; i.e. provide an output statement to display a fraction as its String representation.

6. Add a **private** method *simplify()* to your Fraction class that converts a fraction to its simplest form. For example, the fraction 20 / 60 should be stored in the class instance variables as 1 / 3 (i.e. numerator = 1, denominator = 3). N.B. you will need a method to determine the Greatest Common Divisor (GCD. Remember, both of these methods (simplify and gcd) **must** be private. As these methods are private, client programs cannot access them. So, how are they to be used? They can only be accessed within the Fraction class. Given their purpose, it would mean that any Fraction class method that modifies the instance variables (e.g.: input, add, constructor, set) should call the *simplify()* method to reduce the instance variables to their minimum values. Thus, these methods are used only for *housekeeping*; they are **not** to be used by client programs.

7. Write a client program that allows the user to add up any number of non-zero fractions. The program should display the running total as an exact fraction (in simplified form as numerator / denominator), and as an approximate double value. The user can finish by entering a fraction that represents a zero fraction.

8. Update your UML diagram from point 3 to include all added behaviour of your Fraction class.