In this lab you will learn overriding, overloading, 'this', default / non-default constructors, and recursion. You will also revisit GUI event handlers. This problem has two parts.

Part1 problem statement: Develop a Fraction class that is capable of doing arithmetic with fractions. Fractions should be held in lowest terms, that is, you should divide out any common multiple of the numerator and denominator. So 12/16 should be reduced to 3/4.

As shown in Fig. 1 to 4, your program should ask for two fractions in terms of numerators and denominators. It then prints the two fractions, adds them up, and prints their sum in their reduced-fraction form, and in decimal form.

Part1 solution design: As shown in Fig. 5, the Fraction class has

1. two member variables: numerator and denominator
2. two constructors:
   a. Default constructor: initializes numerator and denominator to 1.
   b. Non-default constructor: initializes numerator and denominator with the values passed to it.
3. The toString() method returns a string by concatenating numerator and denominator separated by a '/' sign. So if the numerator is 5 and denominator is 7, it will return a string "5/7"
4. The toDecimal() method returns the actual value after dividing the numerator with denominator. So if the fraction is 5/7, it will return 0.7142857142857143. To handle division by zero, convert denominator to double.
5. The add() method takes an argument of type Fraction, adds it up with 'this', and then reduces the result to a fraction in lowest terms using findGCD(). It returns the result as a Fraction.
6. The findGCD(n, d) method finds the greatest common divisor. The algorithm to do this is as follows:
   - If n is 0, return 1 *// base-case*
   - if d is 0, return n *// base-case*
   - else, findGCD(d, n % d); *// recurse with n=d, and d = n%d*

```
********** Input first fraction *********
Numerator:
1
Denominator:
2
********** Input second fraction *********
Numerator:
3
Denominator:
4
f1 = 1/2
f2 = 3/4
f1 + f2 = 5/4
The sum in decimal is: 1.25
```

Figure 1: Regular fraction

```
********** Input first fraction *********
Numerator:
1
Denominator:
4
********** Input second fraction *********
Numerator:
3
Denominator:
0
f1 = 1/4
f2 = 3/0
f1 + f2 = 1/0
The sum in decimal is: Infinity
```

Figure 2: Denominator as 0

```
********** Input first fraction *********
Numerator:
0
Denominator:
1
********** Input second fraction *********
Numerator:
2
Denominator:
3
f1 = 0/1
f2 = 2/3
f1 + f2 = 2/3
The sum in decimal is: 0.6666666666666666
```

Figure 3: Numerator as 0

```
********** Input first fraction *********
Numerator:
0
Denominator:
0
********** Input second fraction *********
Numerator:
0
Denominator:
0
f1 = 0/0
f2 = 0/0
f1 + f2 = 0/0
The sum in decimal is: NaN
```

Figure 4: Both 0

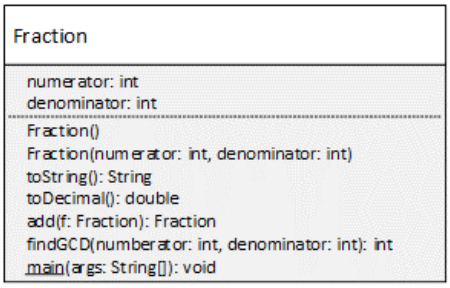

Figure 5: Part1 class diagram

Part2 problem statement: Having worked out the logic of your Fraction program, now attach a GUI frontend that looks as shown below. To save you from tedious creation and alignment of GUI components, I am giving you the code in FractionGUI.java that sets up the screen for you. All you need to do is write and attach event handlers as required. You may create handlers as member-classes or as anonymous classes. Depending on your approach, you may have to make some local variables from setScreen() method into member variables.

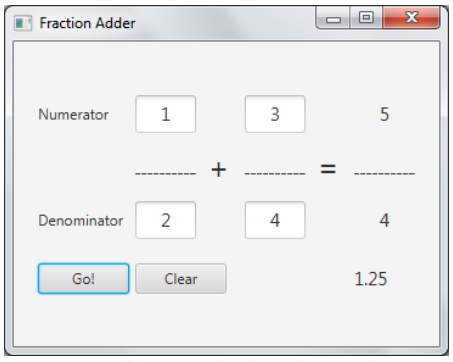

Figure 6: Go! button displays the result

Part2 solution design: As shown in class diagram below, the FractionGUI class has two member classes for the two buttons Go! and Clear. A **JAR[1] file** is provided to you to see the functionality of this application. The Fraction class should not require any change from Part 1 and the FractionGUI should not perform any calculations.

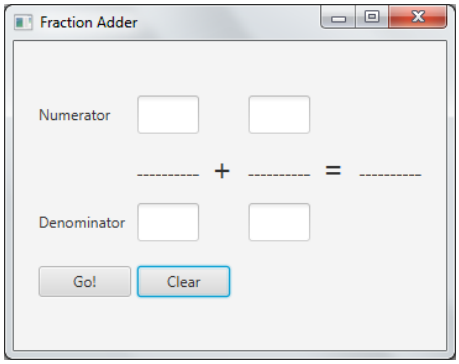

Figure 7: Clear button clears all numbers

Note: As we have not covered error-handling in the course yet, you are not expected to take care of scenarios where user enters non-integer data. The JAR will also throw errors in such cases.

Instructions:

1. Download Fraction.java, TestFraction.java, FractionGUI.java, FractionApp.jar
2. Create a package called lab6 in Labs project.
3. Import the three java files into this package
4. Fill in code for all the methods in Fraction class and test your code using TestFraction.java
5. Now write event handlers in FractionGUI.java. Run your FractionGUI.java and compare it with FractionApp.jar
6. Write your name and Andrew ID in Fraction.java and FractionGUI.java.
7. Zip Fraction.java and FractionGUI.java in a folder named **AndrewID-lab6.zip** and submit the zip file

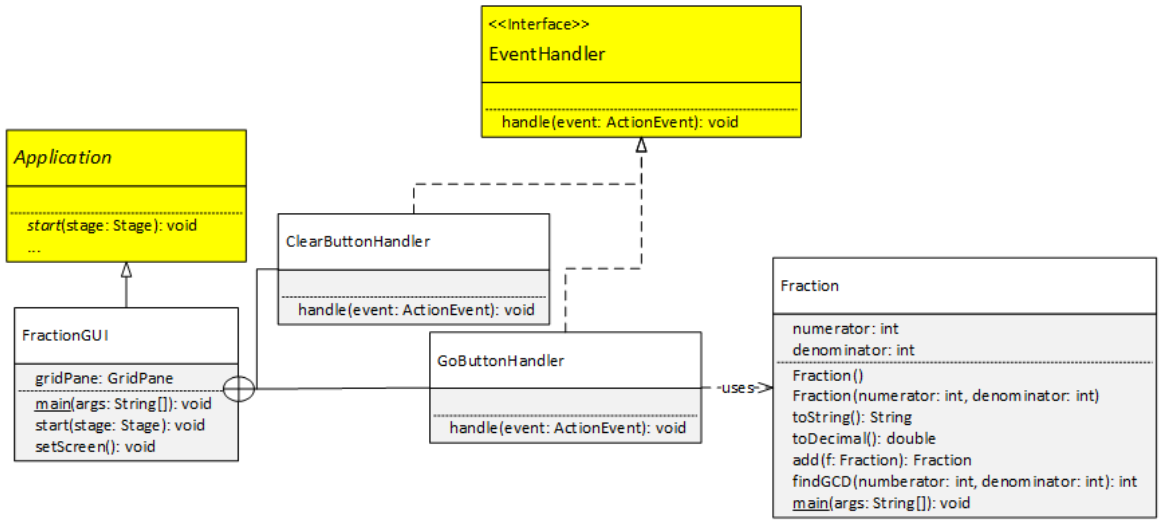

Figure 8: Part2 class diagram

Rubric: 10 Test cases: 5 points (0.5 points each) ; Console output: 2.5 points ; GUI output: 2.5 points

[1] JAR: Java Archive: a collection of class files zipped for distribution. This is a runnable JAR so just download and double click.