**Name**

**Advanced Programming in Java**

**Lab Exercise 10/4/2023**

In this exercise you will use a Fraction class which I have designed. I have two versions of this class; one that contains static methods for add, subtract, multiply, and divide, and one that has non-static methods for the same. The Fraction class has two private integer fields; numerator and denominator. It also has a private double field that contains the decimal equivalent of the fraction. It also has public get and set methods for all private fields. Additionally, it has static methods for add, subtract, multiply, and divide. It also has a non-static method reciprocate that takes the object and swaps the numerator and denominator as well as updating the decimal equivalent field. The Fraction class also has a constructor that initializes the private fields of the constructed object to some safe value. Initializing the denominator to 0 is not a real good idea. It also have a constructor that initializes the Fraction object to some program specified values ( i.e. Fraction one = new Fraction(2, 3) )

You can download both versions of my Fraction class at the following location:

\\Ada\Data Files\Java\Lab Exercise 10.28.2020

We will **first test the Fraction class that contains static methods**.

1. In order to test our class, we need to make a FractionTest class. In our FractionTest class, we will first start by creating three Fraction objects with one initialized to ½, one initialized to ¼, and the third calling the default constructor.
2. The following code will test addition using the Fraction class with **static** methods:

// Test Addition

System.out.println("Test of addition");

answer = Fraction.add(one, two);

one.Print();

System.out.print(" + ");

two.Print();

System.out.print(" = ");

answer.Print();

System.out.println("\t\t" + answer.getDecimal());

1. Repeat step 2 for multiplication, subtraction, and division
2. Now test the reciprocate method by doing division using reciprocate and multiply methods.
3. Repeat steps 1 – 4 using the Fraction class with **non-static** methods. Here is an example of testing addition:

// Test Addition

System.out.println("Test of addition");

answer = one.add(two);

one.Print();

System.out.print(" + ");

two.Print();

System.out.print(" = ");

answer.Print();

System.out.println("\t\t" + answer.getDecimal());