

1)

Get two objects representing a numerator and denominator + Add two Fractions + Output as fraction

```
12 public class Week4ProjectClient {
13
14     /**
15     * @param args the command line arguments
16     */
17     public static void main(String[] args) {
18
19         Fraction firstFrac = new Fraction();
20         Fraction secondFrac = new Fraction(); //NOTE: WHY DID I CREATE ANOTHER OBJECT? Because frac4 stores a different numerator and denom than frac5
21         Fraction thirdFrac = new Fraction();
22
23         do { //Keep looping until fraction is a zero
24             firstFrac.input();
25             firstFrac.display();
26             secondFrac.input();
27             secondFrac.display();
28
29
30             thirdFrac = firstFrac.add(secondFrac); //Add two fractions
31
32
33             thirdFrac.display();
34
35         } while (!firstFrac.isZero());
36     }
37
38 }
```

Client program-

Class-

```
1 public Fraction add(Fraction otherFrac) { //NOTE: WHY did I return Fraction? Because it returns a data type: Fraction
2     //NOTE: WHY didn't I include static? Because the the instance variables for the different objects must be different
3
4     Fraction addFraction = new Fraction(); //Note: WHT Didn't I just use the instance variables of Object: thirdFrac FROM Client class I created? Because we are returning a data type: Fraction
5
6     addFraction.numerator = (this.numerator*otherFrac.denominator)+(this.denominator*otherFrac.numerator); //Note: WHY couldn't I use a varfiabile called addNumerator = ...? Because the instance variable
7     //a new object: Fracion is called numerator
8     addFraction.denominator = (this.denominator*otherFrac.denominator);
9
10    return addFraction;
11
12 }
```

2)

Get two objects representing a numerator and denominator + Determine if fraction is zero + Output as fraction

```
12 public class Week4ProjectClient {
13
14     /**
15     * @param args the command line arguments
16     */
17     public static void main(String[] args) {
18
19         Fraction firstFrac = new Fraction();
20         Fraction secondFrac = new Fraction(); //NOTE: WHY DID I CREATE ANOTHER OBJECT? Because frac4 stores a different numerator and denom than frac5
21         Fraction thirdFrac = new Fraction();
22
23         do { //Keep looping until fraction is a zero
24             firstFrac.input();
25             firstFrac.display();
26             secondFrac.input();
27             secondFrac.display();
28
29             boolean equalFrac = firstFrac.isEqual(secondFrac); //firstFrac is the calling object. Compares two fraction objects
30             thirdFrac = firstFrac.add(secondFrac);
31
32             firstFrac.dspIsEqual(equalFrac); //Display the results of the comparision
33             thirdFrac.display();
34
35         } while (!firstFrac.isZero());
36     }
37
38 }
```

Client program-

```
65     /**
66     * Pre-condition: numerator is an integer (ask tutor whether correct)
67     * Post-condition: returns a true or false representing whether numerator is zero
68     *
69     */
70
71     public boolean isZero() {
72
73         if(numerator == 0) { //Check whether numerator entered is zero. Because numerator is 0 leads to fraction being zero
74             return true;
75         } else {
76             return false;
77         }
78
79     }
```

Class-

4)

Get two objects representing a numerator and denominator + Convert fraction (int) to decimal

Client program-

```
2 public class Week4Question4ProjectClient {
3
4     public static void main(String[] args) {
5         double fracInDecimal = 0;
6
7         Fraction4 firstFrac = new Fraction4();
8
9         do { //Keep looping until fraction is a zero
10            firstFrac.input();
11            firstFrac.display();
12
13
14            fracInDecimal = firstFrac.dblValue();
15            System.out.printf("Decimal value: %f", fracInDecimal); //Very poor. Don't do it. Make method in class program
16
17        } while (!firstFrac.isZero());
18
19    }
20
21 }
22
```

Class-

```
145 public double dblValue() {
146
147     double dblCalc = 0;
148     double dblNumerator = 0;
149     double dblDenominator = 0;
150
151     dblNumerator = (double) this.numerator;
152     dblDenominator = (double) this.denominator;
153
154     dblCalc = dblNumerator/dblDenominator;
155
156     return dblCalc;
157 }
```

6)

Get object representing a numerator and denominator + Simplify fraction

Client program-

```
Fraction frac1 = new Fraction();
```

```
frac1.Input() ← The input method calls the private GCD
```

```
1 private int GCD() { //NOTE: I
2
3     int i = 0;
4     int maxDenominator = 0;
5     int maxNumerator = 0;
6     int smallNum = 0;
7     boolean findDivisor = true;
8     int divisor = 0;
9
10    maxDenominator = this.denominator;
11    maxNumerator = this.numerator;
12
13    boolean neg = (maxNumerator < 0); //Turn negative numerator to positive
14    if(neg) {
15        maxNumerator = -maxNumerator;
16    }
17
18    if(maxNumerator > maxDenominator) { //Compare numerator and denominator. Assign smallest to a variable. Because from smallest number ---> to even smaller number
19        smallNum = maxDenominator; //smallNum = largest possible factor
20    } else {
21        smallNum = maxNumerator;
22    }
23
24
25    for(i = smallNum; i > 0 && findDivisor; i--) { //Keep looping until the highest common divisor is found
26        //Start from smallest number/largest possible factor (numerator or denominator) and get smaller and smaller
27
28        boolean divisorOfNumerator = (maxDenominator % i == 0); //Number factor of numerator?
29        boolean divisorOfDenominator = (maxNumerator % i == 0); //Number factor of denominator?
30
31        if(divisorOfNumerator && divisorOfDenominator) { //Check if the number can be divide evenly with the numerator and denominator i.e factor of both
32            divisor = i;
33            findDivisor = false;
34        }
35    }
36
37
38
39    return divisor;
40 }
```

Class-

7)

Fraction
- numerator: int - denominator: int
+ input(): void + display(): void + isZero(): boolean

+isEqual(): void?