```java
package myExceptions;

public class myNumberFormatException extends Exception{
    private String s;

    public myNumberFormatException(String s) {
        this.s = s;
    }

    @Override
    public String toString() {
        return "NumberFormatException: " + this.s + " is not a number.";
    }
}

```

```
1  package myExceptions;
2
3  public class ZeroDenominatorException extends Exception{
4      @Override
5      public String toString() {
6          return "ZeroDenominatorException{}";
7      }
8  }
9
```

```
1  import java.util.ArrayList;
2  import java.util.Scanner;
3  import myExceptions.ZeroDenominatorException;
4  import myExceptions.myNumberFormatException;
5
6  public class Driver {
7      public static void main(String[] args) {
8          try {
9              Scanner sc = new Scanner(System.in);
10             System.out.println("Input 2 rational numbers in the form '12/34 56/78
   ': ");
11             String inputs = sc.nextLine();
12             ArrayList<RationalNumber> inputList = inputNumbers(inputs.split(" "));
13
14             String[] choices = {"add", "subtract", "multiply", "divide", "convert to
   double", "convert to float"};
15             for(int i=0; i<choices.length; i++) {
16                 System.out.println("Press " + i + " to " + choices[i]);
17             }
18
19             System.out.println("Enter your choice:");
20             int choice = sc.nextInt();
21
22             System.out.println();
23
24             switch(choice) {
25                 case 0 -> System.out.println(Operations.add(inputList.get(0),
   inputList.get(1)));
26                 case 1 -> System.out.println(Operations.subtract(inputList.get(0),
   inputList.get(1)));
27                 case 2 -> System.out.println(Operations.multiply(inputList.get(0),
   inputList.get(1)));
28                 case 3 -> System.out.println(Operations.divide(inputList.get(0),
   inputList.get(1)));
29                 case 4 -> System.out.println(Operations.toDouble(inputList.get(0
   )) + " " +  Operations.toDouble(inputList.get(1)));
30                 case 5 -> System.out.println(Operations.toFloat(inputList.get(0)) +
   " " +  Operations.toFloat(inputList.get(1)));
31                 default -> System.out.println("Incorrect Choice.");
32             }
33         }catch(myNumberFormatException nfe) {
34             System.out.println(nfe.toString());
35         }catch(ZeroDenominatorException zde){
36             System.out.println(zde.toString());
37         }
38     }
39
40
41     // Asserts the command line arguments
42     private static ArrayList<RationalNumber> inputNumbers(String[] CLAs)
   throws myNumberFormatException, ZeroDenominatorException {
43         String numerator1 = CLAs[0].split("/")[0];
44         String denominator1 = CLAs[0].split("/")[1];
```

```
45          String numerator2 = CLAs[1].split("/")[0];
46          String denominator2 = CLAs[1].split("/")[1];
47
48          if(!isInteger(numerator1)){
49              throw new myNumberFormatException(numerator1);
50          }
51          if(!isInteger(denominator1)){
52              throw new myNumberFormatException(denominator1);
53          }
54          if(!isInteger(numerator2)){
55              throw new myNumberFormatException(numerator2);
56          }
57          if(!isInteger(denominator2)){
58              throw new myNumberFormatException(denominator2);
59          }
60          if(denominator1.compareTo("0") == 0){
61              throw new ZeroDenominatorException();
62          }
63          if(denominator2.compareTo("0") == 0){
64              throw new ZeroDenominatorException();
65          }
66
67          ArrayList<RationalNumber> inputList = new ArrayList<>();
68          inputList.add(new RationalNumber(Integer.parseInt(numerator1), Integer
    .parseInt(denominator1)));
69          inputList.add(new RationalNumber(Integer.parseInt(numerator2), Integer
    .parseInt(denominator2)));
70
71          return inputList;
72      }
73
74
75      // Checks whether a string is integer or not
76      private static boolean isInteger(String s) {
77          for(char c: s.toCharArray()) {
78              if(!Character.isDigit(c))
79                  return false;
80          }
81          return true;
82      }
83  }
84
```

```
1  public class Operations{
2
3      // Add 2 RationalNumber objects
4      public static RationalNumber add(RationalNumber r1, RationalNumber r2) {
5          RationalNumber sum = new RationalNumber();
6          sum.numerator = (r1.numerator * r2.denominator) + (r2.numerator * r1.
   denominator);
7          sum.denominator = r1.denominator * r2.denominator;
8          return sum;
9      }
10
11     // Subtract 2 RationalNumber objects
12     public static RationalNumber subtract(RationalNumber r1, RationalNumber
   r2) {
13         RationalNumber difference = new RationalNumber();
14         difference.numerator = (r1.numerator * r2.denominator) - (r2.numerator *
   r1.denominator);
15         difference.denominator = r1.denominator * r2.denominator;
16         return difference;
17     }
18
19     // Multiply 2 RationalNumber objects
20     public static RationalNumber multiply(RationalNumber r1, RationalNumber
   r2) {
21         RationalNumber product = new RationalNumber();
22         product.numerator = r1.numerator * r2.numerator;
23         product.denominator = r1.denominator * r2.denominator;
24         return product;
25     }
26
27     // Divide 2 RationalNumber objects
28     public static RationalNumber divide(RationalNumber r1, RationalNumber r2
   ) {
29         RationalNumber quotient = new RationalNumber();
30         quotient.numerator = r1.numerator * r2.denominator;
31         quotient.denominator = r1.denominator * r2.numerator;
32         return quotient;
33     }
34
35     // Convert RationalNumber object to double
36     public static double toDouble(RationalNumber r) {
37         return (double)(r.numerator / r.denominator);
38     }
39
40     // Convert RationalNumber object to float
41     public static float toFloat(RationalNumber r) {
42         return (float)(r.numerator / r.denominator);
43     }
44 }
45
```

```java
public class RationalNumber {
    int numerator, denominator;


    public RationalNumber() {
        this.numerator = 0;
        this.denominator = 1;
    }

    public RationalNumber(int numerator, int denominator) {
        this.numerator = numerator;
        this.denominator = denominator;
    }


    @Override
    public String toString() {
        return numerator + "/" + denominator;
    }
}
```

```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.3.1\lib\idea_rt.jar=50350:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.3.1\bin" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath "D:\College\Fourth SEM\java\javA\out\production\javA;C:\Users\Abhishek\.m2\repository\org\jetbrains\annotations\24.0.0\annotations-24.0.0.jar" Driver
Input 2 rational numbers in the form '12/34 56/78':
12/34 78/99
Press 0 to add
Press 1 to subtract
Press 2 to multiply
Press 3 to divide
Press 4 to convert to double
Press 5 to convert to float
Enter your choice:
0

3840/3366

Process finished with exit code 0

```