

CSC 113

EXCEPTIONS 1



We want to write a program that provides two methods dealing with integers: division and interval.

Exercise: Write class **ExceptionTest** that has the following:

- A static method that receives two integers x and y and returns the results of dividing x by y . This method should throw **ArithmeticException** with the message “Division by zero.” if y is 0:
 - `public static int divide(int x, int y)`
- A static method that receives two integers s and e and prints the interval $[s..e]$. This method should throw **IllegalArgumentException** with the message “End point is smaller than start point.” if e is smaller than s .
 - `public static void printInterval(int s, int e)`

- A main method to test the two methods above and handle their thrown exceptions using try-catch blocks. It should also handle

InputMismatchException:

- Keep prompting the user to enter the dividend and the divisor until a valid division can be made then print the result in the format: dividend / divisor = result. In case there's an exception (input mismatch, or division by zero), catch it and print an appropriate message (see sample run).
- Keep prompting the user to enter the start and end points for the interval until a valid print can be made. In case there's an exception (input mismatch, or illegal argument), catch it and print an appropriate message (see sample run).

Sample run:

Enter the dividend and the divisor: 8 a↵

Caught [InputMismatchException](#): expected an integer.

Enter the dividend and the divisor: 8 0↵

Caught [ArithmeticException](#): Division by zero.

Enter the dividend and the divisor: 8 2↵

8 / 2 = 4

Enter the start and end points: a 9↵

Caught [InputMismatchException](#): expected an integer.

Enter the start and end points: 3 1↵

Caught [IllegalArgumentException](#): End point is smaller than start point.

Enter the start and end points: 3 9↵

[3, 4, 5, 6, 7, 8, 9]

ExceptionsTest.java ×

```
1 import java.util.Scanner;
2 import java.util.InputMismatchException;
3
4 public class ExceptionsTest {
5     public static int divide(int x, int y) throws ArithmeticException {
6         if (y == 0)
7             throw new ArithmeticException("Division by zero.");
8
9         return x / y;
10    }
11
12    public static void printInterval(int s, int e) throws IllegalArgumentException {
13        if(s > e)
14            throw new IllegalArgumentException("End point is smaller than start point.");
15
16        System.out.printf("[%d", s);
17        for(int i = s + 1; i <= e; i++)
18            System.out.printf(", %d", i);
19        System.out.println("]");
20    }
```

```
22 public static void main(String[] args) {
23     Scanner input = new Scanner(System.in);
24
25     while (true) {
26         try {
27             System.out.print("Enter the dividend and the divisor: ");
28             int x = input.nextInt();
29             int y = input.nextInt();
30
31             System.out.printf("%d / %d = %d\n", x, y, divide(x, y));
32
33             break;
34         }
35         catch (InputMismatchException e) {
36             System.err.println("Caught InputMismatchException: expected an integer.");
37             input.nextLine();
38         }
39         catch (ArithmeticException e) {
40             System.err.println("Caught ArithmeticException: " + e.getMessage());
41         }
42     }
43 }
```

ExceptionsTest.java ×

```
44     while (true) {
45         try {
46             System.out.print("Enter the start and end points: ");
47             int s = input.nextInt();
48             int e = input.nextInt();
49
50             printInterval(s, e);
51
52             break;
53         }
54         catch (InputMismatchException e) {
55             System.err.println("Caught InputMismatchException: expected an integer.");
56             input.nextLine();
57         }
58         catch (IllegalArgumentException e) {
59             System.err.println("Caught IllegalArgumentException: " + e.getMessage());
60         }
61     }
62     input.close();
63 }
64 }
65 |
```