LAB # 12

EXCEPTION HANDLING

OBJECTIVE: Constructing a fault tolerant program by implementing exception handling techniques.

Lab Task:

Question#01

Write a program that meets the following requirements:

- Creates an array with 10 randomly chosen integers.
- Prompts the user to enter the index of the array, then displays the corresponding element value. If the specified index is out of bounds, display the message Out of Bounds. (ArrayIndexOutOfBoundsException)

```
import java.util.Random;
import java.util.Scanner;
public class lab12 {
    public static void main(String[] args) {

        // Create an array of size 10
        int[] arr = new int[10];
        Random rand = new Random();

        // Fill array with random integers
        for (int i = 0; i < arr.length; i++) {
            arr[i] = rand.nextInt(100); // random numbers from 0 to 99
        }
        // Take index input from user
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter an index (0-9): ");

        try {
            int index = sc.nextInt();
            // Display the value at the given index
            System.out.println("Element at index " + index + " is: " + arr[index]);
        } catch (ArrayIndexOutOfBoundsException e) {
            // Handle out of bounds index
            System.out.println("Out of Bounds");
        }
        sc.close();
    }
}
```

Output:

```
<terminated> lab12 [Java Application] C:\e
Enter an index (0-9):
10
Out of Bounds
```

Question#02

Suppose you are developing a game for kids in which they are learning the division operation in math .Your game will take input from kids (2 integers) and then perform the division and displays the answer. Think and apply exception handling in this scenario. For e.g. Arithmetic Exception might occur here. Also suppose, this game has the limitation that it only performs division between integers so if it gets a decimal number as input, it **throws** an exception stating that the input is invalid, please give integer number etc.

```java
import java.util.Scanner;
// Custom exception for decimal input
class DecimalInputException extends Exception {
    public DecimalInputException(String message) {
        super(message);
    }
}
public class lab12b {

    // Method to check and convert input to integer
    public static int parseInteger(String input) throws DecimalInputException {
        if (input.contains(".")) {
            throw new DecimalInputException("Invalid input! Please enter an integer number only.");
        }
        return Integer.parseInt(input);
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        try {
            System.out.print("Enter first number: ");
            String input1 = sc.next();

            System.out.print("Enter second number: ");
            String input2 = sc.next();

            int num1 = parseInteger(input1);
            int num2 = parseInteger(input2);

            // Division
            int result = num1 / num2;

            System.out.println("Result: " + result);

        } catch (DecimalInputException e) {
            System.out.println(e.getMessage());

        } catch (ArithmeticException e) {
            System.out.println("Error! Division by zero is not allowed.");

        } catch (NumberFormatException e) {
            System.out.println("Invalid input! Please enter numbers only.");

        } finally {
            System.out.println("Game ended. Try again!");
            sc.close();
        }
    }
}
```

Output:

```
<terminated> lab12b [Java Application] C:\e
Enter first number: 5
Enter second number: 2
Result: 2
Game ended. Try again!
```

GitHub: