WEEK-9

ROLL NO:230701235

1) In the following program, an array of integer data is to be initialized.

During the initialization, if a user enters a value other than an integer, it will throw an InputMismatchException exception.

On the occurrence of such an exception, your program should print "You entered bad data."

If there is no such exception it will print the total sum of the array.

For example: Input Result 3 8 5 2 1 2 You entered bad data. 1 g

CODE:

```
import java.util.Scanner; import
java.util.InputMismatchException; class
prog {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
                                         int
length = sc.nextInt();
  // create an array to save user input
                                          int[]
name = new int[length]; int sum=0;//save
the total sum of the array.
  /* Define try-catch block to save user input in the array "name"
If there is an exception then catch the exception otherwise print
the total sum of the array. */
                                 try
   {
```

| | Input | Expected | Got | |
|---|------------|-----------------------|-----------------------|---|
| / | 3 5 2 1 | 8 | 8 | ~ |
| / | 2 1 g | You entered bad data. | You entered bad data. | ~ |

2) Write a Java program to create a method that takes an integer as a parameter and throws an exception if the number is odd.

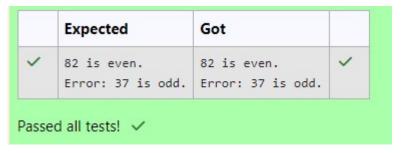
Result 82 is even. Error: 37 is odd.

CODE:

OUTPUT:

class prog {

```
public static void main(String[] args) {
int n = 82;
    tryNumber(n); // Call the tryNumber method
n = 37;
    // Call the tryNumber method for the next number
tryNumber(n);
  }
  public static void tryNumber(int n) {
try {
      // Call the checkEvenNumber method
checkEvenNumber(n);
      System.out.println(n + " is even.");
    } catch (Exception e) {
      // Handle any exceptions thrown
      System.out.println("Error: " + e.getMessage());
    }
  }
  public static void checkEvenNumber(int number) throws Exception {
if (number % 2 != 0) {
      // Throw an exception if the number is odd
throw new Exception(number + " is odd.");
    }
  }
}
OUTPUT:
```



3) Write a Java program to handle ArithmeticException and ArrayIndexOutOfBoundsException.

Create an array, read the input from the user, and store it in the array.

Divide the 0th index element by the 1st index element and store it. if the 1st element is zero, it will throw an exception. if you try to access an element beyond the array limit throws an exception.

For example: Test Input Result 1 6 java.lang.ArithmeticException: / by zero 1 0 4 1 2 8 I am always executed

CODE:

import java.util.Scanner;

```
public class ExceptionHandling {     public
static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        try
{
            // Read the size of the array
int n = sc.nextInt();        int[] arr
= new int[n];
```

// Read the array elements from the user

```
for (int i = 0; i < n; i++) {
arr[i] = sc.nextInt();
      }
      // Attempt to divide the 0th element by the 1st element
int result = arr[0] / arr[1];
      // Attempt to access an element beyond the array limit (e.g., index 3)
System.out.println("Accessing out of bound element: " + arr[3]);
    } catch (ArithmeticException e) {
      // Handle division by zero
      System.out.println("java.lang.ArithmeticException: " + e.getMessage());
    } catch (ArrayIndexOutOfBoundsException e) {
      // Handle array index out of bounds
      System.out.println("java.lang.ArrayIndexOutOfBoundsException: " +
e.getMessage());
    } finally {
      // This block will always be executed
      System.out.println("I am always executed");
    }
  }
}
OUTPUT:
```

| | Test | Input | Expected | Got |
|---|------|----------|--|---------------|
| ~ | 1 | 6 | java.lang.ArithmeticException: / by zero | java.lang.Ari |
| | | 104128 | I am always executed | I am always e |
| ~ | 2 | 3 | java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3 | java.lang.Arr |
| | | 10 20 30 | I am always executed | I am always e |