

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-09-Exception Handling](#) / [Lab-09-Logic Building](#)

<b>Status</b>	Finished
<b>Started</b>	Monday, 21 October 2024, 5:37 PM
<b>Completed</b>	Monday, 21 October 2024, 6:08 PM
<b>Duration</b>	31 mins 5 secs

## Question 1

Correct

Marked out of 5.00

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.

/\* 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. \*/

**Sample Input:**

3  
5 2 1

**Sample Output:**

8

**Sample Input:**

2  
1 g

**Sample Output:**

You entered bad data.

**For example:**

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

**Answer:** (penalty regime: 0 %)

Reset answer

```

1 import java.util.Scanner;
2 import java.util.InputMismatchException;
3 public class ArraySum {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         try {
7             int n = scanner.nextInt();
8             int[] numbers = new int[n];
9             for (int i = 0; i < n; i++) {
10                 numbers[i] = scanner.nextInt();
11             }
12             int sum = 0;
13             for (int num : numbers) {
14                 sum += num;
15             }
16             System.out.println(sum);
17         } catch (InputMismatchException e) {
18             System.out.println("You entered bad data.");
19         } finally {
20             scanner.close();
21         }
22     }
23 }
24 }
25 }
26 }
```

	Input	Expected	Got	
✓	3 5 2 1	8	8	✓
✓	2 1 g	You entered bad data.	You entered bad data.	✓

Passed all tests! ✓

## Question 2

Correct

Marked out of 5.00

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.

**Input:**

5  
10 0 20 30 40

**Output:**

**java.lang.ArithmeticException: / by zero**  
**I am always executed**

Input:

3  
10 20 30

**Output**

java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3  
I am always executed

**For example:**

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

**Answer:** (penalty regime: 0 %)

```

1 import java.util.Scanner;
2 public class ExceptionHandlingExample {
3     public static void main(String[] args) {
4         Scanner scanner = new Scanner(System.in);
5         try {
6             int size = scanner.nextInt();
7             int[] numbers = new int[size];
8             for (int i = 0; i < size; i++) {
9                 numbers[i] = scanner.nextInt();
10            }
11            int result = numbers[0] / numbers[1];
12            int outOfBoundsAccess = numbers[size]; // This will throw ArrayIndexOutOfBoundsException
13
14        } catch (ArithmeticException e) {
15            System.out.println(e);
16        } catch (ArrayIndexOutOfBoundsException e) {
17            System.out.println(e);
18        } finally {
19            System.out.println("I am always executed");
20            scanner.close();
21        }
22    }
23 }
24

```

	Test	Input	Expected	Got	
✓	1	6 1 0 4 1 2 8	java.lang.ArithmeticException: / by zero I am always executed	java.lang.ArithmeticException: / by zero I am always executed	✓
✓	2	3 10 20 30	java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3 I am always executed	java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3 I am always executed	✓

Passed all tests! ✓

## Question 3

Correct

Marked out of 5.00

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

**Sample input and Output:**

82 is even.  
Error: 37 is odd.

Fill the preloaded answer to get the expected output.

**For example:****Result**

82 is even.  
Error: 37 is odd.

**Answer:** (penalty regime: 0 %)

Reset answer

```

1 public class EvenOddChecker {
2     static class OddNumberException extends Exception {
3         public OddNumberException(String message) {
4             super(message);
5         }
6     }
7     public static void checkEven(int number) throws OddNumberException {
8         if (number % 2 != 0) {
9             throw new OddNumberException("Error: " + number + " is odd.");
10        } else {
11            System.out.println(number + " is even.");
12        }
13    }
14    public static void main(String[] args) {
15        int[] numbers = {82, 37};
16        for (int number : numbers) {
17            try {
18                checkEven(number);
19            } catch (OddNumberException e) {
20                System.out.println(e.getMessage());
21            }
22        }
23    }
24 }
25

```

	Expected	Got	
✓	82 is even. Error: 37 is odd.	82 is even. Error: 37 is odd.	✓

Passed all tests! ✓

◀ Lab-09-MCQ

Jump to...

[The "Nambiar Number" Generator ►](#)