# OODP Workshop 5

1. Consider the following program specification:

Write a program that will prompt the user for 10 integer values and stores them in an array. The values entered by the user only be between 0 and 100 and the program should validate these appropriately. The program should then:

* Display the average of the values
* Display the highest value in the array
* Display the lowest value in the array
  1. Discuss how the values should be validated. What control structures could be used?

**Solution:**

The java code for this:

*//Id: k240175*

*//Sabbir Hasan Rabbi*

import **java.util.Scanner**;

**public** **class** Array {

**public** **static** **void** main(**String**[] args) {

**Scanner** scanner = new Scanner(System.in);

**int**[] values = new **int**[10];

**int** sum = 0;

**int** highest = Integer.MIN\_VALUE;

**int** lowest = Integer.MAX\_VALUE;

        for (**int** i = 0; i < values.length; i++) {

            while (true) {

                System.out.print("Enter value " + (i + 1) + " (between 0 and 100): ");

**int** input = scanner.nextInt();

                if (input >= 0 && input <= 100) {

                    values[i] = input;

                    break;

                } else {

                    System.out.println("Invalid input. Please enter a number between 0 and 100.");

                }

            }

            sum += values[i];

            if (values[i] > highest) {

                highest = values[i];

            }

            if (values[i] < lowest) {

                lowest = values[i];

            }

        }

**double** average = (**double**) sum / values.length;

        System.out.println("Average value: " + average);

        System.out.println("Highest value: " + highest);

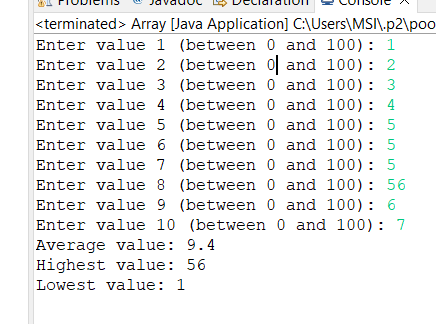
        System.out.println("Lowest value: " + lowest);

        scanner.close();

    }

}

The output:



### **Validation Discussion:**

**Validation** is crucial to ensure that the user inputs are within the acceptable range (0 to 100). I used this methos for my program.

1. **Prompting User Input**: Use a loop to repeatedly prompt the user until valid input is provided. In the example above, a while (true) loop ensures that the user input is repeatedly requested until it meets the specified criteria.
2. **Input Validation**: Inside the loop, use an if statement to check whether the input is within the valid range (0 to 100). If the input is valid, it is stored in the array, and the loop breaks. If invalid, the user is informed, and the loop continues to prompt for input.
3. **Updating Statistics**: After receiving valid input, the program updates:
   * The sum for calculating the average later.
   * The highest and lowest values by comparing the current input against the existing highest and lowest values.
4. **Control Structures**:
   * **if Statements**: Used to validate whether the input is within the allowed range and to determine if the current value is the highest or lowest.
   * **while (true) Loop**: Used for repeatedly prompting the user until valid input is obtained.

By using these control structures and validation techniques, I ensured that the program handles invalid input gracefully and provides accurate statistical information based on the given data.

1. **You have been asked to store the IDs and scores of competitors in three rounds of a game using a 2D Array and display the scores on the screen from the same 2D array. Following is the data:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Score 1 | Score 2 | Score 3 |
| 123 | 23 | 12 | 24 |
| 234 | 34 | 10 | 32 |
| 345 | 12 | 34 | 21 |
| 456 | 10 | 23 | 21 |
| 567 | 10 | 19 | 23 |

**Solution:**

The java code :

package testing;

**public** **class** Array {

**public** **static** **void** main(String[] args) {

**int**[][] data = {

            {123, 23, 12, 24},

            {234, 34, 10, 32},

            {345, 12, 34, 21},

            {456, 10, 23, 21},

            {567, 10, 19, 23}

        };

        System.out.print( "ID    " + "Score1    "+ "Score2    "+"Score3    ");

        System.out.println();

        for (**int** i = 0; i < data.length; i++) {

            for(**int** j=0;j<4;j++)

            {

                System.out.print(data[i][j]+"\t  ");

            }

            System.out.println();

        }

    }

}

The output:

