

## Conditional Statements, Loops and Basic Array Manipulations

### Lab Objective

Familiarize students with conditional statements, looping and basic array manipulations in Java.

### Lab Outcomes

After completing this lab successfully, students will be able to:

1. **Apply** concepts of conditional statements, loops and arrays and **write** and **execute** programs using these concepts in Java.

### Psychomotor Learning Levels

This lab involves activities that encompass the following learning levels in psychomotor domain.

Level	Category	Meaning	Keywords
P1	Imitation	Copy action of another; observe and replicate.	Relate, Repeat, Choose, Copy, Follow, Show, Identify, Isolate.
P2	Manipulation	Reproduce activity from instruction or memory	Copy, response, trace, Show, Start, Perform, Execute, Recreate.

### Lab Activities

#### A. Reading Inputs and Writing Outputs using Dialog Box (JOptionPane)

- JOptionPane is a built-in Java class that makes it easy to pop up a standard dialog box that prompts users for a value or informs them of something.
- JOptionPane class has the following methods.

Method Name	Description
showConfirmDialog	Asks a confirming question, like yes/no/cancel
<b>showInputDialog</b>	Prompts for some input
<b>showMessageDialog</b>	Tell the user about something that has happened
showOptionDialog	The grand unification of the above three

- Write the following Java program.

```
import javax.swing.*;
class SampleDialogBox1 {
    public static void main (String[] args){
        JOptionPane.showMessageDialog (null, "Hello World",
            "First Dialog", JOptionPane.INFORMATION_MESSAGE);
    }
}
```

- Compile and execute the program. You will see the dialog box as shown in Figure 1.

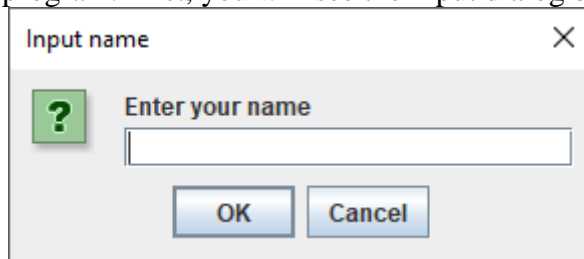


*Figure 1: A simple message dialog box*

- Next, we will see the process of reading inputs from user through a dialog box.
- Write the following Java program.

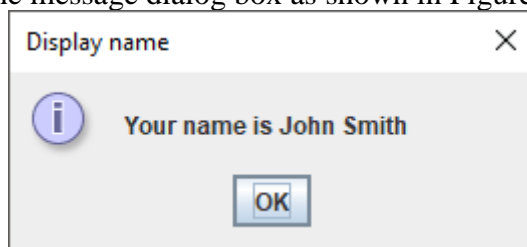
```
import javax.swing.*;
class SampleDialogBox2 {
    public static void main (String[] args){
        String name = JOptionPane.showInputDialog (null,
            "Enter your name", "Input name",
            JOptionPane.QUESTION_MESSAGE);
        JOptionPane.showMessageDialog(null, "Your name is" + name,
            "Display name", JOptionPane.INFORMATION_MESSAGE);
    }
}
```

- Compile and execute the program. First, you will see the input dialog box as shown in Figure 2.



*Figure 2: A simple input dialog box*

- After you enter the name, the message dialog box as shown in Figure 3 will be displayed.



*Figure 3: Dialog box shown after you enter name*

- Write, compile and execute the following Java program.

```
import javax.swing.*;
class SampleDialogBox3 {
    public static void main (String[] args){
        String first = JOptionPane.showInputDialog (null,
            "Enter first integer", "Input first integer",
            JOptionPane.QUESTION_MESSAGE);
        int firstNumber = Integer.parseInt(first);
        String second = JOptionPane.showInputDialog (null,
            "Enter second integer", "Input second integer",
            JOptionPane.QUESTION_MESSAGE);
        int secondNumber = Integer.parseInt(second);
    }
}
```

```

        int sum = firstNumber + secondNumber;
        JOptionPane.showMessageDialog(null, firstNumber + " + " +
            secondNumber + " = " + sum, "Result",
            JOptionPane.INFORMATION_MESSAGE);
    }
}

```

- More information on JOptionPane can be found in the following links.
  - <https://docs.oracle.com/javase/7/docs/api/javax/swing/JOptionPane.html>
  - <https://www.javatpoint.com/java-joptionpane>
- **Write a Java program that reads your weight in kilograms and your height in meters and then calculates the BMI (Body Mass Index) and displays the value.**

$$BMI = weight / (height)^2$$

Use appropriate JOptionPane dialog boxes to read inputs and write outputs.

## B. Arrays in Java

- Arrays are declared in the following manner in Java.
 

```
int[] numbers = new int[10];
```
- The above-mentioned line created an integer array numbers of size 10.
- The array elements can be accessed in the same way as in C.
- Examine the following program.

```

import java.io.*;
import java.lang.*;
import java.util.*;
class SampleArray{
    public static void main(String[] args){
        // initializing an integer array numbers with size 10
        int[] numbers = new int [10];
        // assigning values randomly
        for(int i=0; i<numbers.length; i++){
            numbers[i] = (int) (Math.random()*100);
        }
        // displaying the values
        for(int i=0; i<numbers.length; i++){
            System.out.print(numbers[i] + " ");
        }
    }
}

```

- **numbers.length** returns the size of the array.
- For Java String, you can use **charAt(i)** method to access the character at i<sup>th</sup> position of the string. Suppose,
 

```
String str = "Hello World";
System.out.print(str.charAt(0)); // returns the character 'H'
```
- **Write a Java program that reads the temperatures of last ten days and finds on which day the temperature was below the average of last ten days. Assume that days are numbered from 0 to 9.**

## C. Useful Mathematical Functions

- **Math.pow(a,b)** – returns  $a^b$   
Example: **Math.pow(2,3)** returns 8.
- **Math.random()** – returns a floating-point value in between 0 and 1 (exclusive).

*This lab sheet must be preserved till the end of the semester.*

Student ID		Solved during the LAB	
Student Name			
Section		Solved after the LAB	
Date			

## Lab 02: Lab Problems

**Lab02\_Problem01 (Conditional Statements):** The two roots of a quadratic equation  $ax^2 + bx + c = 0$  can be obtained using the following formula:

$$r_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{and} \quad r_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

$b^2 - 4ac$  is called the discriminant of the quadratic equation. If it is positive, the equation has two real roots. If it is zero, the equation has one root. If it is negative, the equation has no real roots. Write a Java program that prompts the user to enter values for  $a$ ,  $b$ , and  $c$  and displays the result based on the discriminant. If the discriminant is positive, display two roots. If the discriminant is **0**, display one root. Otherwise, display “The equation has no real roots”.

Note that you can use **Math.pow(x, 0.5)** to compute  $2x$ . Use appropriate *JOptionPane* dialog boxes to read inputs and write outputs.

**Lab02\_Problem02 (Looping):** Write a Java program that reads ten integers from console and determines how many positive and negative values have been read and computes the total and average of the input values (not counting zeros). Your program ends with the input **0**. Display the average as a floating-point number.

**Lab02\_Problem03 (Looping):** The greatest common divisor (GCD) of two integers **n1** and **n2** can be found in the following way: First find **d** to be the minimum of **n1** and **n2**, then check whether **d**, **d-1**, **d-2**, . . . , **2**, or **1** is a divisor for both **n1** and **n2** in this order. The first such common divisor is the greatest common divisor for **n1** and **n2**.

Write a Java program that prompts the user to enter two positive integers and displays the GCD. Use appropriate *JOptionPane* dialog boxes to read inputs and write outputs.

**Lab02\_Problem04 (Arrays):** Write a Java program that reads student scores, gets the best score, and then assigns grades based on the following scheme:

- Grade is A if score is  $\geq$  best - 10
- Grade is B if score is  $\geq$  best - 20;
- Grade is C if score is  $\geq$  best - 30;
- Grade is D if score is  $\geq$  best - 40;
- Grade is F otherwise.

The program prompts the user to enter the total number of students, then prompts the user to enter all of the scores and concludes by displaying the grades. Here is a sample run:

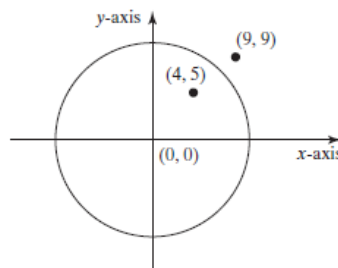
```
Enter the number of students: 4 Enter
Enter 4 scores: 40 55 70 58 Enter
Student 0 score is 40 and grade is C
Student 1 score is 55 and grade is B
Student 2 score is 70 and grade is A
Student 3 score is 58 and grade is B
```

**Lab02\_Problem05 (Arrays):** Write a Java program that reads the integers between 1 and 100 and counts the occurrences of each. Assume the input ends with 0. Here is a sample run of the program:

```
Enter the integers between 1 and 100: 2 5 6 5 4 3 23 43 2 0 Enter
2 occurs 2 times
3 occurs 1 time
4 occurs 1 time
5 occurs 2 times
6 occurs 1 time
23 occurs 1 time
43 occurs 1 time
```

**Lab02\_Problem06 (Conditional Statements):** Write a Java program that prompts the user to enter the exchange rate from currency in U.S. dollars (USD) to Bangladeshi Taka (BDT). Prompt the user to enter 0 to convert from USD to BDT and 1 to convert from BDT to USD. Then, prompt the user to enter the amount in USD or in BDT to convert it to BDT or USD, respectively. Use appropriate *JOptionPane* dialog boxes to read inputs and write outputs.

**Lab02\_Problem07:** Write a Java program that prompts the user to enter the center ( $p, q$ ) and the radius ( $r$ ) of a circle. Then, prompts user to enter a point ( $x, y$ ) and checks whether the point is within the circle centered at ( $p, q$ ) with radius  $r$ . For example, (4, 5) is inside the circle centered at (0,0) with radius 10 and (9, 9) is outside the circle, as shown in the following figure.



**Lab02\_Problem08:** A *Palindrome* is a number or a string that reads the same from either way (forward or backward). As an example, RADAR is a palindrome, but ROVER is not.

Write a Java program that prompts the user to enter a string and displays whether the string is a palindrome or not. You may not use any built-in Java methods to accomplish that.

**Lab02\_Problem09:** Write a Java program that randomly generates an integer array, *numbers*, of size 100. Then, find the value and index (position) of the highest and the smallest element.

**Lab02\_Problem10:** Write a Java program that reads in ten numbers and displays the number of distinct numbers and the distinct numbers separated by exactly one space (i.e., if a number appears multiple times, it is displayed only once). (*Hint:* Read a number and store it to an array if it is new. If the number is already in the array, ignore it.) After the input, the array contains the distinct numbers.