List of Practical (With Solution)

Course: BCA

Subject Name and Code: Programming in Java Lab (BCAC BCAC0819)

Year/Semester: II/III

Objective: The objective of this course is that the students will understand fundamentals of programming such as variables, conditional and iterative execution, methods, etc. It will also provide the foundation of good programming skills by discussing key issues to the design of object oriented programming.

Experiment 1: (If Else)

Compulsory:

Program 1: WAP to print a message “Hello JAVA”.

```
public class HelloJavaProgram1 {
    public static void main(String[] args) {
        // Print the message "Hello JAVA"
        System.out.println("Hello JAVA");
    }
}
run:
Hello JAVA
BUILD SUCCESSFUL (total time: 0 seconds)
```

Program 2: Write a program to display whether a number is even or odd.

```
import java.util.Scanner;

public class EvenOddChecker {
    public static void main(String[] args) {
        // Create a Scanner object to read input from the user
        Scanner scanner = new Scanner(System.in);

        // Prompt the user to enter a number
        System.out.print("Enter an integer: ");
        int number = scanner.nextInt();

        // Check if the number is even or odd
        if (number % 2 == 0) {
            System.out.println(number + " is even.");
        } else {
            System.out.println(number + " is odd.");
        }
        // Close the scanner
        scanner.close();
    }
}
```

Program 3 Write the following program using if else if ladder. Accept an hour from the user and output the following as indicated below. Include the last condition in the else section.

i) Hour greater than or equal to 0 and less than 12	“Good Morning”
ii) Hour greater than or equal to 12 and less than 18	“Good Afternoon”
iii) Hour greater than or equal to 18 and less than 24	“Good Evening”
iv) Any other input	“Time is out of range”

Solution

```
import java.util.Scanner;

public class TimeGreeting {
    public static void main(String[] args) {
        // Create a Scanner object to read input from the user
        Scanner scanner = new Scanner(System.in);

        // Prompt the user to enter an hour
        System.out.print("Enter the hour (0-23): ");
        int hour = scanner.nextInt();

        // Determine the appropriate greeting based on the hour
        if (hour >= 0 && hour < 12) {
            System.out.println("Good Morning");
        } else if (hour >= 12 && hour < 18) {
            System.out.println("Good Afternoon");
        } else if (hour >= 18 && hour < 24) {
            System.out.println("Good Evening");
        } else {
            System.out.println("Time is out of range");
        }

        // Close the scanner
        scanner.close();
    }
}
```

Additional Programs:

Program 4 : A student receives marks in three subjects. The program will calculate the total marks, average marks, and determine the grade based on the average:

Average >= 90: Grade A
Average >= 80 and < 90: Grade B
Average >= 70 and < 80: Grade C
Average >= 60 and < 70: Grade D
Average < 60: Grade

```
import java.util.Scanner;

public class StudentGradeCalculator {
    public static void main(String[] args) {
        // Create a Scanner object to read input from the user
        Scanner scanner = new Scanner(System.in);

        // Prompt the user to enter marks for three subjects
        System.out.print("Enter marks for Subject 1: ");
        double marks1 = scanner.nextDouble();

        System.out.print("Enter marks for Subject 2: ");
        double marks2 = scanner.nextDouble();

        System.out.print("Enter marks for Subject 3: ");
        double marks3 = scanner.nextDouble();

        // Calculate total marks and average
        double totalMarks = marks1 + marks2 + marks3;
        double averageMarks = totalMarks / 3;

        // Determine the grade based on the average marks
        char grade;
        if (averageMarks >= 90) {
            grade = 'A';
        } else if (averageMarks >= 80) {
            grade = 'B';
        } else if (averageMarks >= 70) {
            grade = 'C';
        } else if (averageMarks >= 60) {
            grade = 'D';
        } else {
            grade = 'F';
        }
        // Display the results
        System.out.println("Total Marks: " + totalMarks);
        System.out.println("Average Marks: " + averageMarks);
        System.out.println("Grade: " + grade);

        // Close the scanner
        scanner.close();
    }
}
```

Experiment 2: Nested If Else

Compulsory:

Program 1: : WAP to find maximum of three numbers

```
import java.util.Scanner;

public class MaxOfThreeNumbers {
    public static void main(String[] args) {
        // Create a Scanner object to read input from the user
        Scanner scanner = new Scanner(System.in);

        // Prompt the user to enter three numbers
        System.out.print("Enter the first number: ");
        double num1 = scanner.nextDouble();

        System.out.print("Enter the second number: ");
        double num2 = scanner.nextDouble();

        System.out.print("Enter the third number: ");
        double num3 = scanner.nextDouble();

        // Find the maximum of the three numbers
        double max;
        if (num1 >= num2 && num1 >= num3) {
            max = num1;
        } else if (num2 >= num1 && num2 >= num3) {
            max = num2;
        } else {
            max = num3;
        }

        // Display the result
        System.out.println("The maximum of the three numbers is: " + max);

        // Close the scanner
        scanner.close();
    }
}
```

```
// program that finds the maximum of three numbers using nested if-else statements:
import java.util.Scanner;

public class MaxOfThreeNumbersNestedIfElse {
    public static void main(String[] args) {
        // Create a Scanner object to read input from the user
        Scanner scanner = new Scanner(System.in);

        // Prompt the user to enter three numbers
        System.out.print("Enter the first number: ");
        double num1 = scanner.nextDouble();

        System.out.print("Enter the second number: ");
        double num2 = scanner.nextDouble();

        System.out.print("Enter the third number: ");
        double num3 = scanner.nextDouble();

        // Find the maximum of the three numbers using nested if-else
        double max;
        if (num1 >= num2) {
            // num1 is greater than or equal to num2
            if (num1 >= num3) {
                // num1 is also greater than or equal to num3
                max = num1;
            } else {
                // num1 is greater than or equal to num2 but less than num3
                max = num3;
            }
        } else {
            // num1 is less than num2
            if (num2 >= num3) {
                // num2 is greater than or equal to num3
                max = num2;
            } else {
                // num2 is less than num3
                max = num3;
            }
        }

        // Display the result
        System.out.println("The maximum of the three numbers is: " + max);

        // Close the scanner
        scanner.close();
    }
}
```

Program 2: A movie theatre has the following ticket pricing rules:

Children under 12 years old pay $5.

Seniors 65 years and older pay $7.

Regular adults (12-64 years old) pay $10.

Members get a $2 discount on all ticket prices.

We'll prompt the user to input their age and whether they have a membership card. Based on this input, the program will determine and print the ticket price.

Solution

```
import java.util.Scanner;

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

        // Get the age of the customer
        System.out.print("Enter your age: ");
        int age = scanner.nextInt();

        // Get the membership status of the customer
        System.out.print("Do you have a membership card? (yes/no): ");
        String membership = scanner.next();

        // Initialize the ticket price
        double ticketPrice = 0.0;

        // Determine the ticket price based on age
        if (age < 12) {
            ticketPrice = 5.0;
        } else if (age >= 65) {
            ticketPrice = 7.0;
        } else {
            ticketPrice = 10.0;
        }

        // Apply discount if the customer has a membership card
        if (membership.equalsIgnoreCase("yes")) {
            ticketPrice -= 2.0;
        }

        // Display the final ticket price
        System.out.println("Your ticket price is: $" + ticketPrice);

        // Close the scanner
        scanner.close();
    }
}
```

```
run:
Enter your age: 34
Do you have a membership card? (yes/no): yes
Your ticket price is: $8.0
BUILD SUCCESSFUL (total time: 5 seconds)
```

Program3: Create a program using switch case statement to identify the day of the week.

```
import java.util.Scanner;

public class DayOfWeek {
    public static void main(String[] args) {
        // Create a Scanner object to read input from the user
        Scanner scanner = new Scanner(System.in);

        // Prompt the user to enter a number (1-7)
        System.out.print("Enter a number (1-7) to find the day of the week: ");
        int dayNumber = scanner.nextInt();

        // Identify the day of the week using a switch statement
        String dayName;
        switch (dayNumber) {
            case 1:
                dayName = "Monday";
                break;
            case 2:
                dayName = "Tuesday";
                break;
            case 3:
                dayName = "Wednesday";
                break;
            case 4:
                dayName = "Thursday";
                break;
            case 5:
                dayName = "Friday";
                break;
            case 6:
                dayName = "Saturday";
                break;
            case 7:
                dayName = "Sunday";
                break;
            default:
                dayName = "Invalid number. Please enter a number between 1 and 7.";
                break;
        }

        // Display the result
        System.out.println("Day of the week: " + dayName);

        // Close the scanner
        scanner.close();
    }
}
```

Additional Programs:

Program3: WAP to implement that performs basic arithmetic operations (addition, subtraction, multiplication, division) based on user input. use switch to select the arithmetic operation

```
import java.util.Scanner;

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

        System.out.println("Enter first number: ");
        double num1 = scanner.nextDouble();
        System.out.println("Enter second number: ");
        double num2 = scanner.nextDouble();

        System.out.println("Choose an operation: +, -, *, /");
        char operator = scanner.next().charAt(0);

        double result;

        switch (operator) {
            case '+':
                result = num1 + num2;
                break;
            case '-':
                result = num1 - num2;
                break;
            case '*':
                result = num1 + num2;
                break;
            case '/':
                if (num2 != 0) {
                    result = num1 / num2;
                } else {
                    System.out.println("Division by zero is not allowed.");
                    return;
                }
                break;
            default:
                System.out.println("Invalid operator");
                return;
        }

        System.out.println("The result is: " + result);
    }
}
```

Experiment 3: Simple and Nested Loop

Compulsory:

Program1 : Write a Program in Java to Calculate the Factorial of an Integer using a for loop

```
import java.util.Scanner;

public class FactorialCalculator3 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Prompt user to enter a number
        System.out.print("Enter a number to calculate its factorial: ");
        int number = scanner.nextInt();

        // Calculate the factorial
        long factorial = 1;
        for (int i = 1; i <= number; i++) {
            factorial *= i;
        }

        // Print the factorial
        System.out.println("The factorial of " + number + " is " + factorial);

        // Close the scanner
        scanner.close();
    }
}
```

```
run:
Enter a number to calculate its factorial: 5
The factorial of 5 is 120
BUILD SUCCESSFUL (total time: 6 seconds)
```

Program 2: You are tasked with developing a program that simulates a login system. The user is given three attempts to enter the correct password (In integer format eg. 4545). If the user enters the correct password within three attempts, they are granted access. If the user fails to enter the correct password in three attempts, they are locked out.

```
import java.util.Scanner;

public class LoginSystem {
    public static void main(String[] args) {
        // Define the correct password
        final int CORRECT_PASSWORD = 4545;
        // Create a Scanner object to read input from the user
        Scanner scanner = new Scanner(System.in);

        // Variable to track the number of attempts
        int attempts = 0;
        boolean accessGranted = false;

        // Allow the user up to 3 attempts to enter the correct password
        while (attempts < 3) {
            // Prompt the user to enter the password
            System.out.print("Enter password: ");
            int enteredPassword = scanner.nextInt();

            // Check if the entered password is correct
            if (enteredPassword == CORRECT_PASSWORD) {
                accessGranted = true;
                break;
            } else {
                // Increment the number of attempts
                attempts++;
                System.out.println("Incorrect password. You have " + (3 - attempts) + " attempts
left.");
            }
        }

        // Display the result based on whether access was granted
        if (accessGranted) {
            System.out.println("Access granted.");
        } else {
            System.out.println("Access denied. You have been locked out.");
        }

        // Close the scanner
        scanner.close();
    }
}
```

Additional Programs:

Program 3: WAP to display table of a number in a format n x i=m

```
import java.util.Scanner;

public class MultiplicationTable {
    public static void main(String[] args) {
        // Create a Scanner object to read input from the user
        Scanner scanner = new Scanner(System.in);

        // Prompt the user to enter a number
        System.out.print("Enter a number to display its multiplication table: ");
        int number = scanner.nextInt();

        // Display the multiplication table
        System.out.println("Multiplication table for " + number + ":");
        for (int i = 1; i <= 10; i++) {
            int result = number * i;
            System.out.println(number + " x " + i + " = " + result);
        }

        // Close the scanner
        scanner.close();
    }
}
```

Program 4: WAP to find sum of digits of a number

```
import java.util.Scanner;

public class SumOfDigits {
    public static void main(String[] args) {
        // Create a Scanner object to read input from the user
        Scanner scanner = new Scanner(System.in);

        // Prompt the user to enter a number
        System.out.print("Enter an integer: ");
        int number = scanner.nextInt();

        // Initialize sum to 0
        int sum = 0;

        // Make the number positive if it is negative
        number = Math.abs(number);

        // Calculate the sum of the digits
        while (number > 0) {
            // Extract the last digit of the number
            int digit = number % 10;
            // Add the digit to the sum
            sum += digit;
            // Remove the last digit from the number
            number /= 10;
        }

        // Display the result
        System.out.println("Sum of the digits: " + sum);

        // Close the scanner
        scanner.close();
    }
}
```

Program 5: WAP to implement type conversion in java.

```
public class TypeConversionDemo {
    public static void main(String[] args) {
        // Implicit Type Conversion (Widening Conversion)
        int intValue = 100;
        double doubleValue = intValue;  // int is automatically converted to double
        System.out.println("Implicit Conversion:");
        System.out.println("Integer value: " + intValue);
        System.out.println("Double value after implicit conversion: " + doubleValue);

        // Explicit Type Conversion (Narrowing Conversion)
        doubleValue = 123.456;
        intValue = (int) doubleValue;  // double is explicitly converted to int

        System.out.println("\nExplicit Conversion:");
        System.out.println("Double value: " + doubleValue);
        System.out.println("Integer value after explicit conversion: " + intValue);

        // Converting String to Numeric Types
        String numberString = "2024";
        int number = Integer.parseInt(numberString);  // String to int

        System.out.println("\nString to Numeric Conversion:");
        System.out.println("String value: " + numberString);
        System.out.println("Integer value after conversion: " + number);

        // Converting Numeric Types to String
        int anotherNumber = 1234;
        String anotherNumberString = Integer.toString(anotherNumber);  // int to String

        System.out.println("\nNumeric to String Conversion:");
        System.out.println("Integer value: " + anotherNumber);
        System.out.println("String value after conversion: " + anotherNumberString);
    }
}
```

Experiment 4: Patterns

Compulsory:

Program 1: Using For…..Loop display the following pattern :

```
*
**
***
****
*****
```

```
public class PatternDisplay {
    public static void main(String[] args) {
        // Number of rows for the pattern
        int rows = 5;

        // Loop to iterate through each row
        for (int i = 1; i <= rows; i++) {
            // Loop to print stars in each row
            for (int j = 1; j <= i; j++) {
                System.out.print("*");
            }
            // Move to the next line after printing stars in a row
            System.out.println();
        }
    }
}
```

Program 2: WAP to print the pattern

```
11111
12111
11311
11141
11115
```

```
public class PatternPrint {
    public static void main(String[] args) {
        // Number of rows and columns for the pattern
        int size = 5;

        // Loop to iterate through each row
        for (int i = 0; i < size; i++) {
            // Loop to print each column in the current row
            for (int j = 0; j < size; j++) {
                // Print '1' except for the diagonal and last column
                if (i == j) {
                    System.out.print((i + 1));  // Print 1-based index of the row
                } else {
                    System.out.print("1");
                }
            }
            // Move to the next line after printing all columns in a row
            System.out.println();
        }
    }
```

Additional Programs:

Program3: WAP to print the pattern

```
11111
12111
12311
12341
12345
```

```
public class PatternPrint {
    public static void main(String[] args) {
        // Number of rows and columns for the pattern
        int size = 5;

        // Loop to iterate through each row
        for (int i = 1; i <= size; i++) {
            // Loop to print each column in the current row
            for (int j = 1; j <= size; j++) {
                // Print numbers up to the current row index or 1
                if (j <= i) {
                    System.out.print(j);
                } else {
                    System.out.print("1");
                }
            }
            // Move to the next line after printing all columns in a row
            System.out.println();
        }
    }
}
```

Program 4: WAP to print the pattern

```
#####
$####
$$###
$$$##
$$$$#
```

```
public class PatternPrint {
    public static void main(String[] args) {
        // Number of rows for the pattern
        int rows = 4;

        // Loop to iterate through each row
        for (int i = 1; i <= rows; i++) {
            // Print '$' characters for each row
            for (int j = 1; j < i; j++) {
                System.out.print("$");
            }
            // Print '#' characters for each row
            for (int k = i; k < rows; k++) {
                System.out.print("#");
            }
            // Move to the next line after printing all characters in the row
            System.out.println();
        }
    }
}
```

Experiment 5: Array

Compulsory:

Program 1: WAP to read and print an array. Also find greatest and smallest element in array.

```
import java.util.Scanner;

public class ArrayOperations {
    public static void main(String[] args) {
        // Create a Scanner object to read input from the user
        Scanner scanner = new Scanner(System.in);

        // Prompt the user to enter the number of elements in the array
        System.out.print("Enter the number of elements in the array: ");
        int size = scanner.nextInt();

        // Initialize the array with the specified size
        int[] array = new int[size];

        // Read the elements of the array from the user
        System.out.println("Enter " + size + " elements:");
        for (int i = 0; i < size; i++) {
            array[i] = scanner.nextInt();
        }

        // Print the elements of the array
        System.out.println("Array elements:");
        for (int i = 0; i < size; i++) {
            System.out.print(array[i] + " ");
        }
        System.out.println();

        // Initialize variables to find the smallest and largest elements
        int smallest = array[0];
        int largest = array[0];

        // Find the smallest and largest elements in the array
        for (int i = 1; i < size; i++) {
            if (array[i] < smallest) {
                smallest = array[i];
            }
            if (array[i] > largest) {
                largest = array[i];
            }
        }

        // Display the smallest and largest elements
        System.out.println("Smallest element in the array: " + smallest);
        System.out.println("Largest element in the array: " + largest);

        // Close the scanner
        scanner.close();
    }
}
```

Program 2 You're developing a simple scoring system for a cricket match. The match involves a single over (6 balls) and you're required to record the runs scored on each ball. The system should also calculate the total runs scored in the over and determine if the over included any dot balls (a ball where no runs are scored).

Task:
Create a 1-dimensional array to store the runs scored on each of the 6 balls in the over.
Calculate the total runs scored in the over.
Count the number of dot balls (balls where zero runs were scored).
Determine the highest run scored on a single ball.

Runs scored in the over (ball by ball):

Ball 1: 1 run
Ball 2: 4 runs
Ball 3: 0 runs (dot ball)
Ball 4: 6 runs
Ball 5: 2 runs
Ball 6: 0 runs (dot ball)

```
public class CricketScoringSystem {
    public static void main(String[] args) {
        // Step 1: Initialize the array with runs scored on each ball
        int[] runsPerBall = {1, 4, 0, 6, 2, 0};

        // Step 2: Calculate the total runs scored in the over
        int totalRuns = 0;
        for (int runs : runsPerBall) {
            totalRuns += runs;
        }

        // Step 3: Count the number of dot balls
        int dotBalls = 0;
        for (int runs : runsPerBall) {
            if (runs == 0) {
                dotBalls++;
            }
        }

        // Step 4: Determine the highest run scored on a single ball
        int highestRun = runsPerBall[0];
        for (int runs : runsPerBall) {
            if (runs > highestRun) {
                highestRun = runs;
            }
        }

        // Output the results
        System.out.println("Total runs scored in the over: " + totalRuns);
        System.out.println("Number of dot balls: " + dotBalls);
        System.out.println("Highest run scored on a single ball: " + highestRun);
    }
}
```

Additional Programs:

Program 3: Write a program to search an element in an array.

```
import java.util.Scanner;

public class ArraySearch {
    public static void main(String[] args) {
        // Create a Scanner object to read input from the user
        Scanner scanner = new Scanner(System.in);

        // Prompt the user to enter the number of elements in the array
        System.out.print("Enter the number of elements in the array: ");
        int size = scanner.nextInt();

        // Initialize the array with the specified size
        int[] array = new int[size];
        // Read the elements of the array from the user
        System.out.println("Enter " + size + " elements:");
        for (int i = 0; i < size; i++) {
            array[i] = scanner.nextInt();
        }

        // Prompt the user to enter the element to search for
        System.out.print("Enter the element to search for: ");
        int searchElement = scanner.nextInt();

        // Search for the element in the array
        boolean found = false;
        int position = -1;
        for (int i = 0; i < size; i++) {
            if (array[i] == searchElement) {
                found = true;
                position = i;
                break;
            }
        }

        // Display the result
        if (found) {
            System.out.println("Element " + searchElement + " is present at index " + position +
".");
        } else {
            System.out.println("Element " + searchElement + " is not present in the array.");
        }

        // Close the scanner
        scanner.close();
    }
}
```

Program 4: Write a program to sort an array using bubble sort.

```
import java.util.Scanner;

public class BubbleSortExample {
    public static void main(String[] args) {
        // Create a Scanner object to read input from the user
        Scanner scanner = new Scanner(System.in);

        // Prompt the user to enter the number of elements in the array
        System.out.print("Enter the number of elements in the array: ");
        int size = scanner.nextInt();

        // Initialize the array with the specified size
        int[] array = new int[size];

        // Read the elements of the array from the user
        System.out.println("Enter " + size + " elements:");
        for (int i = 0; i < size; i++) {
            array[i] = scanner.nextInt();
        }

        // Perform bubble sort
        bubbleSort(array);

        // Print the sorted array
        System.out.println("Sorted array:");
        for (int i = 0; i < size; i++) {
            System.out.print(array[i] + " ");
        }
        System.out.println();

        // Close the scanner
        scanner.close();
    }

    // Method to perform bubble sort
    public static void bubbleSort(int[] array) {
        int n = array.length;
        boolean swapped;
        // Loop through all array elements
        for (int i = 0; i < n - 1; i++) {
            swapped = false;
            // Compare each pair of adjacent elements
            for (int j = 0; j < n - i - 1; j++) {
                if (array[j] > array[j + 1]) {
                    // Swap if elements are in the wrong order
                    int temp = array[j];
                    array[j] = array[j + 1];
                    array[j + 1] = temp;
                    swapped = true;
                }
            }
            // If no elements were swapped, the array is sorted
            if (!swapped) {
                break;
            }
        }
    }
}
```

Program 5: WAP to read and print a matrix

```
import java.util.Scanner;

public class MatrixOperations {
    public static void main(String[] args) {
        // Create a Scanner object to read input from the user
        Scanner scanner = new Scanner(System.in);

        // Prompt the user to enter the number of rows and columns
        System.out.print("Enter the number of rows: ");
        int rows = scanner.nextInt();
        System.out.print("Enter the number of columns: ");
        int columns = scanner.nextInt();

        // Initialize the matrix with the specified rows and columns
        int[][] matrix = new int[rows][columns];

        // Read the elements of the matrix from the user
        System.out.println("Enter the elements of the matrix:");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < columns; j++) {
                System.out.print("Element [" + i + "][" + j + "]: ");
                matrix[i][j] = scanner.nextInt();
            }
        }

        // Print the matrix
        System.out.println("Matrix:");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < columns; j++) {
                System.out.print(matrix[i][j] + " ");
            }
            System.out.println(); // Move to the next line after printing a row
        }

        // Close the scanner
        scanner.close();
    }
}
```

Program 6: WAP to find sum of two matrices

```
import java.util.Scanner;

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

        // Prompt the user to enter the number of rows and columns
        System.out.print("Enter the number of rows: ");
        int rows = scanner.nextInt();
        System.out.print("Enter the number of columns: ");
        int columns = scanner.nextInt();

        // Initialize matrices with the specified rows and columns
        int[][] matrix1 = new int[rows][columns];
        int[][] matrix2 = new int[rows][columns];
        int[][] sumMatrix = new int[rows][columns];

        // Read elements of the first matrix
        System.out.println("Enter the elements of the first matrix:");
        readMatrix(scanner, matrix1, rows, columns);

        // Read elements of the second matrix
        System.out.println("Enter the elements of the second matrix:");
        readMatrix(scanner, matrix2, rows, columns);

        // Compute the sum of the two matrices
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < columns; j++) {
                sumMatrix[i][j] = matrix1[i][j] + matrix2[i][j];
            }
        }

        // Display the resulting matrix
        System.out.println("Sum of the two matrices:");
        printMatrix(sumMatrix, rows, columns);

        // Close the scanner
        scanner.close();
    }

    // Method to read matrix elements from the user
    public static void readMatrix(Scanner scanner, int[][] matrix, int rows, int columns) {
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < columns; j++) {
                System.out.print("Element [" + i + "][" + j + "]: ");
                matrix[i][j] = scanner.nextInt();
            }
        }
    }

    // Method to print matrix
    public static void printMatrix(int[][] matrix, int rows, int columns) {
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < columns; j++) {
                System.out.print(matrix[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

Experiment 6: Strings

Compulsory:

Program1: Write a Java program that will accept command-line arguments and display the same.

```
public class CommandLineArgs {
    public static void main(String[] args) {
        // Check if there are any command-line arguments
        if (args.length == 0) {
            System.out.println("No command-line arguments provided.");
```