

**S12BLH31**  
**PROGRAMMING IN JAVA**

**LAB MANUAL**

**UNIT –I PROGRAMS**

**1 a. Implementation of Matrix Operations using Arrays.**

**Ex-1.a1: Matrix Addition**

**Aim:**

To write a Java program to implement Matrix Addition using Arrays.

**Algorithm:**

Step 1: Start

Step 2: Create a Scanner object to read input from the user.

Step 3: Prompt the user to enter the number of rows (r) and columns (c).

Step 4: Declare three 2D arrays of size r x c:

- A[][] → to store the first matrix
- B[][] → to store the second matrix
- sum[][] → to store the result of matrix addition

Step 5: Prompt the user to enter elements for the first matrix

Step 6: Prompt the user to enter elements for the second matrix

Step 7: Perform matrix addition

Step 8: Display the result matrix (sum[][])

Step 9: Stop

**Program:**

```
import java.util.Scanner;

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

        System.out.print("Enter the number of rows: ");
        int r = s.nextInt();

        System.out.print("Enter the number of columns: ");
        int c = s.nextInt();

        int A[][] = new int[r][c];
        int B[][] = new int[r][c];
        int sum[][] = new int[r][c];

        System.out.println("Enter elements for the first matrix:");
        for (int i = 0; i < r; i++)
        {
            for (int j = 0; j < c; j++)
            {
                A[i][j] = s.nextInt();
            }
        }

        System.out.println("Enter elements for the second matrix:");
```

```
for (int i = 0; i < r; i++)  
{  
    for (int j = 0; j < c; j++)  
    {  
        B[i][j] = s.nextInt();  
    }  
}
```

```
for (int i = 0; i < r; i++)  
{  
    for (int j = 0; j < c; j++)  
    {  
        sum[i][j] = A[i][j] + B[i][j];  
    }  
}
```

```
System.out.println("Sum of the matrices:");
```

```
for (int i = 0; i < r; i++)  
{  
    for (int j = 0; j < c; j++)  
    {  
        System.out.print(sum[i][j] + " ");  
    }  
    System.out.println();  
}  
}
```

### **Output:**

#### **Compilation:**

C:\javanew>javac MatrixAddition.java

#### **Run:**

C:\javanew>java MatrixAddition

Enter the number of rows: 3

Enter the number of columns: 3

Enter elements for the first matrix:

2

2

2

2

2

2

2

2

2

Enter elements for the second matrix:

3

1

4

5

2

3

4

5

1

Sum of the matrices:

5 3 6

7 4 5

6 7 3

### **Result:**

Thus the Addition of Matrix program was compiled and executed successfully.

## **Ex- 1.a2: Matrix Multiplication**

### **Aim:**

To write a Java program to implement Matrix Multiplication using Arrays.

### **Algorithm:**

Step 1: Start the program

Step 2: Create a Scanner object to read input from the user.

Step 3: Read the number of rows and columns for both matrices:

Step 4: Check the compatibility condition for matrix multiplication:

- If  $col1 \neq row2$ , print "Matrix multiplication is not possible" and exit the program.

Step 5: Declare three matrices A,B and C

Step 6: Input elements of matrix A

Step 7: Input elements of matrix B

Step 8: Perform matrix multiplication

Step 9: Display the result matrix C

Step 10: Stop the program

**Program:**

```
import java.util.Scanner;

public class matrixmultiplication
{

    public static void main(String args[])
    {
        int row1, col1, row2, col2;
        Scanner s = new Scanner(System.in);

        System.out.print("Enter number of rows in first matrix: ");
        row1 = s.nextInt();
        System.out.print("Enter number of columns in first matrix: ");
        col1 = s.nextInt();
        System.out.print("Enter number of rows in second matrix: ");
        row2 = s.nextInt();
        System.out.print("Enter number of columns in second matrix: ");
        col2 = s.nextInt();
        if (col1 != row2)
        {
            System.out.println("Matrix multiplication is not possible");
            return;
        }

        int A[][] = new int[row1][col1];
        int B[][] = new int[row2][col2];
        int C[][] = new int[row1][col2];
```

```
System.out.println("\nEnter values for matrix A : ");  
for (int i = 0; i < row1; i++)
```

```
{  
    for (int j = 0; j < col1; j++)  
    {  
        A[i][j] = s.nextInt();  
    }  
}
```

```
System.out.println("\nEnter values for matrix B : ");  
for (int i = 0; i < row2; i++)
```

```
{  
    for (int j = 0; j < col2; j++)  
    {  
        B[i][j] = s.nextInt();  
    }  
}
```

```
System.out.println("\nMatrix multiplication is : ");  
for (int i = 0; i < row1; i++)
```

```
{  
    for (int j = 0; j < col2; j++)  
    {  
        C[i][j] = 0;  
        for (int k = 0; k < col1; k++)  
        {  
            C[i][j] += A[i][k] * B[k][j];  
        }  
        System.out.print(C[i][j] + " ");  
    }  
}
```

```
        System.out.println();
    }

}

}
```

### **Output:**

#### **Compilation:**

C:\javanew>javac matrixmultiplication.java

#### **Run:**

C:\javanew>java matrixmultiplication

Enter number of rows in first

matrix: 2 Enter number of

columns in first matrix: 3 Enter

number of rows in second

matrix: 3

Enter number of columns in second matrix: 2

Enter values for matrix A :

2

4

6

8

10

12

Enter values for matrix B :

2

4



3

5

6

7

Matrix multiplication is :

52    70

118   166

**Result:**

Thus the Multiplication of Matrix program was compiled and executed successfully.

### **Ex- 1.a3: Matrix Transpose**

#### **Aim:**

To write a Java Program implementing Matrix Transpose using Arrays.

#### **Algorithm:**

Step 1: Start

Step 2: Create a Scanner object to read input from the user.

Step 3: Read the number of rows (r) and columns (c) of the matrix.

Step 4: Declare a 2D array A of size [r][c] to store the original matrix.

Step 5: Input the elements of the matrix A[i][j]

Step 6: Display the original matrix

Step 7: Display the transpose of the matrix A[j][i]

Step 8: End

#### **Program:**

```
import java.util.Scanner;

public class Transpose
{
    public static void main(String[] args)
    {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter the number of rows: ");
        int r = s.nextInt();
        System.out.print("Enter the number of columns: ");
        int c = s.nextInt();
```

```
int A[][] = new int[r][c];
```

```
System.out.println("Enter elements for the first matrix:");
```

```
for (int i = 0; i < r; i++)
```

```
{
```

```
    for (int j = 0; j < c; j++)
```

```
    {
```

```
        A[i][j] = s.nextInt();
```

```
    }
```

```
}
```

```
System.out.println("Original Matrix:");
```

```
for (int i = 0; i < r; i++)
```

```
{
```

```
    for (int j = 0; j < c; j++)
```

```
    {
```

```
        System.out.print(A[i][j] + "  ");
```

```
    }
```

```
    System.out.println();
```

```
}
```

```
System.out.println("Transpose of Matrix:");
```

```
for (int i = 0; i < r; i++)
```

```
{
```

```
    for (int j = 0; j < c; j++)
```

```
    {
```

```
        System.out.print(A[j][i] + "  ");
```

```
    }
```

```
    System.out.println();
```

```
}
```

```
}  
}
```

### **Output:**

#### **Compilation:**

C:\javanew>javac Transpose.java

#### **Run:**

C:\javanew>java Transpose

Enter the number of rows: 2

Enter the number of columns: 2

Enter elements for the first matrix:

1

2

3

4

Original Matrix:

1   2

3   4

Transpose of Matrix:

1   3

2   4

#### **Result:**

Thus the Transpose of Matrix program was compiled and executed successfully.

## **1 b. Program to perform String Operations.**

### **Ex-1.b: String Operations**

#### **Aim:**

To write a Java program to perform String Operations.

#### **Algorithm:**

Step 1: Start

Step 2: Declare and initialize string variables

Step 3: Perform and display the length of the string

Step 4: Trim leading and trailing spaces from the string

Step 5: Convert string to uppercase and lowercase

Step 6: Access a character at a specific index

Step 7: Extract a substring from the string

Step 8: Compare two strings for equality

Step 9: Check if the string contains a specific sequence

Step 10: Replace a character in the string

Step 11: Split the string into words using a delimiter (like space)

Step 12: End

**Program:**

```
public class StringOperations {  
    public static void main(String[] args) {  
        String s1 = " Hello Java ";  
        String s2 = "hello";  
        System.out.println("String Operations");  
        System.out.println("Length: " + s1.length());  
        System.out.println("Trimmed: " + s1.trim());  
        System.out.println("Uppercase: " + s1.toUpperCase());  
        System.out.println("Lowercase: " + s1.toLowerCase());  
        System.out.println("Char at 1: " + s1.charAt(1));  
        System.out.println("Substring (1 to 5): " + s1.substring(1, 5));  
        System.out.println("Equals: " + s1.trim().equals(s2));  
        System.out.println("Contains 'Java': " + s1.contains("Java"));  
        System.out.println("Replace 'a' with 'x': " + s1.replace('a', 'x'));  
        System.out.println("Split by space:");  
        for (String word : s1.trim().split(" ")) {  
            System.out.println(word);  
        }  
    }  
}
```

## **Output:**

### **Compilation:**

```
C:\javanew>javac StringOperations.java
```

### **Run:**

```
C:\javanew>java StringOperations
```

String Operations

Length: 12

Trimmed: Hello Java

Uppercase: HELLO JAVA

Lowercase: hello java

Char at 1: H

Substring (1 to 5): Hell

Equals: false

Contains 'Java': true

Replace 'a' with 'x': Hello Jxvx

Split by space:

Hello

Java

## **1.c Program to implement Looping Control Statements**

### **Ex-1.c: Looping Control Statements**

#### **Aim:**

To write a Java program to implement Looping control statements.

#### **Algorithm:**

Step 1: Start

Step 2: Print numbers from 1 to 5 using for loop

Step 3: Prompt the user: "Enter a number to find sum using while loop: "

Step 4: Password Check using do-while loop

Step 5: Print Array Elements using Enhanced for loop

Step 6: End

#### **Program:**

```
import java.util.Scanner;

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

        // 1. FOR LOOP: Print numbers 1 to 5
        System.out.println("Using for loop:");
        for (int i = 1; i <= 5; i++) {
            System.out.print(i + " ");
        }
        System.out.println("\n");
```



## **// 2. WHILE LOOP: Sum of first n numbers**

```
System.out.print("Enter a number to find sum using while loop: ");
int n = s.nextInt(), sum = 0, i = 1;
while (i <= n) {
    sum += i;
    i++;
}
System.out.println("Sum = " + sum + "\n");
```

## **// 3. DO-WHILE LOOP: Keep asking until password is correct**

```
String password;
do {
    System.out.print("Enter password (hint: java): ");
    password = s.next();
} while (!password.equals("java"));
System.out.println("Access Granted!\n");
```

## **// 4. ENHANCED FOR LOOP: Print array elements**

```
int[] marks = {90, 85, 78, 92, 88};
System.out.println("Student Marks:");
for (int mark : marks) {
    System.out.println(mark);
}
}
```

## **Output:**

### **Compilation:**

```
C:\javanew>javac LoopExamples.java
```

### **Run:**

```
C:\javanew>java LoopExamples
```

Using for loop:

1 2 3 4 5

Enter a number to find sum using while loop: 1000

Sum = 500500

Enter password (hint: java): hello

Enter password (hint: java): java

Access Granted!

Student Marks:

90

85

78

92

88

## **1.d Program to implement Conditional Control Statements**

**Ex-1.d:** Conditional Control Statements

### **Aim:**

To write a Java program to implement Conditional control statements.

### **Algorithm:**

Step 1: Start

Step 2: Input the student's marks

Step 3: Validate marks using if statement

Step 4: Check Pass or Fail using if-else statement

Step 5: Grade Evaluation using if-else-if ladder

Step 6: Check Distinction Category using nested if

Step 7: Print Remarks Based on Grade

Step 8: End

### **Program:**

```
import java.util.Scanner;
```

```
public class StudentEvaluation {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        // Input student's marks  
        System.out.print("Enter student marks (0–100): ");  
        int marks = scanner.nextInt();  
        // 1. IF Statement – Basic check
```

```
if (marks >= 0 && marks <= 100) {  
    System.out.println("Valid marks entered.");  
}
```

## **// 2. IF-ELSE – Pass or Fail**

```
if (marks >= 35) {  
    System.out.println("Result: Pass");  
} else {  
    System.out.println("Result: Fail");  
}
```

## **// 3. IF-ELSE-IF – Grade Evaluation**

```
String grade;  
if (marks >= 90) {  
    grade = "A+";  
} else if (marks >= 80) {  
    grade = "A";  
} else if (marks >= 70) {  
    grade = "B";  
} else if (marks >= 60) {  
    grade = "C";  
} else if (marks >= 50) {  
    grade = "D";  
} else if (marks >= 35) {  
    grade = "E";  
} else {  
    grade = "F";  
}  
System.out.println("Grade: " + grade);
```

**// 4. NESTED IF – Check for distinction**

```
if (marks >= 75) {  
    if (marks >= 90) {  
        System.out.println("Category: Distinction with Honors");  
    } else {  
        System.out.println("Category: Distinction");  
    }  
}
```

**// 5. SWITCH – Remarks based on grade**

```
switch (grade) {  
    case "A+":  
        System.out.println("Remarks: Excellent Performance!");  
        break;  
    case "A":  
        System.out.println("Remarks: Very Good!");  
        break;  
    case "B":  
        System.out.println("Remarks: Good!");  
        break;  
    case "C":  
        System.out.println("Remarks: Fair.");  
        break;  
    case "D":  
        System.out.println("Remarks: Satisfactory.");  
        break;  
    case "E":  
        System.out.println("Remarks: Needs Improvement.");
```

```
        break;
    case "F":
        System.out.println("Remarks: Failed. Better luck next time.");
        break;
    default:
        System.out.println("Remarks: Invalid grade.");
    }
    scanner.close();
}
}
```

### **Output:**

#### **Compilation:**

C:\javanew>javac StudentEvaluation.java

#### **Run:**

C:\javanew>java StudentEvaluation

Enter student marks (0?100): 75

Valid marks entered.

Result: Pass

Grade: B

Category: Distinction

Remarks: Good!