

---

## Experiment 6

Write a Program in Java to add two matrices. The objective of this assignment is to learn Arrays in Java

```
1  import java.util.Scanner;
2
3  public class Matrix {
4
5      public static void main(String[] args) {
6          Scanner sc = new Scanner(System.in);
7          int choice;
8
9          // Loop to allow the user to perform multiple operations
10         do {
11             // Menu for the user to select an operation
12             System.out.println("\n--- Matrix Operations Menu ---");
13             System.out.println("1. Addition");
14             System.out.println("2. Subtraction");
15             System.out.println("3. Multiplication");
16             System.out.println("4. Transpose");
17             System.out.println("5. Exit");
18             System.out.print("Enter your choice (1-5): ");
19             choice = sc.nextInt();
20
21             // Switch statement to call the appropriate method based on user's choice
22             switch (choice) {
23                 case 1:
24                     addMatrices(sc);
25                     break;
26                 case 2:
27                     subtractMatrices(sc);
28                     break;
29                 case 3:
30                     multiplyMatrices(sc);
31                     break;
32                 case 4:
33                     transposeMatrix(sc);
34                     break;
35                 case 5:
36                     System.out.println("Exiting program. Goodbye!");
37                     break;
38                 default:
39                     System.out.println("Invalid choice. Please enter a number between 1 and 5.");
40             }
41         } while (choice != 5); // Continue until the user chooses to exit
42
43         sc.close(); // Close the scanner after the loop is exited
44     }
45
46     /**
47      * Adds two matrices of the same dimensions[cite: 2, 64].
48      */
49     public static void addMatrices(Scanner sc) {
50         System.out.print("Enter number of rows: ");
51         int rows = sc.nextInt();
52         System.out.print("Enter number of columns: ");
53         int cols = sc.nextInt();
54
55         int[][] matrix1 = new int[rows][cols];
56         int[][] matrix2 = new int[rows][cols];
57         int[][] sum = new int[rows][cols];
58
59         System.out.println("Enter elements of Matrix 1:");
60         for (int i = 0; i < rows; i++) {
61             for (int j = 0; j < cols; j++) {
62                 matrix1[i][j] = sc.nextInt();
63             }
64         }
65
66         System.out.println("Enter elements of Matrix 2:");
67         for (int i = 0; i < rows; i++) {
68             for (int j = 0; j < cols; j++) {
69                 matrix2[i][j] = sc.nextInt();
70             }
71         }
72         // Performing addition
73         for (int i = 0; i < rows; i++) {
74             for (int j = 0; j < cols; j++) {
75                 sum[i][j] = matrix1[i][j] + matrix2[i][j];
```

```

    }
}

System.out.println("Resultant Matrix (Sum):");
printMatrix(sum);
}

/**
 * Subtracts the second matrix from the first.
 * Both matrices must have the same dimensions.
 */
public static void subtractMatrices(Scanner sc) {
    System.out.print("Enter number of rows: ");
    int rows = sc.nextInt();
    System.out.print("Enter number of columns: ");
    int cols = sc.nextInt();

    int[][] matrix1 = new int[rows][cols];
    int[][] matrix2 = new int[rows][cols];
    int[][] difference = new int[rows][cols];

    System.out.println("Enter elements of Matrix 1 (Minuend):");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            matrix1[i][j] = sc.nextInt();
        }
    }

    System.out.println("Enter elements of Matrix 2 (Subtrahend):");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            matrix2[i][j] = sc.nextInt();
        }
    }

    // Performing subtraction
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            difference[i][j] = matrix1[i][j] - matrix2[i][j];
        }
    }

    System.out.println("Resultant Matrix (Difference):");
    printMatrix(difference);
}

/**
 * Multiplies two matrices.
 * The number of columns in the first matrix must equal the number of rows in the second.
 */
public static void multiplyMatrices(Scanner sc) {
    System.out.print("Enter number of rows for Matrix 1: ");
    int rows1 = sc.nextInt();
    System.out.print("Enter number of columns for Matrix 1: ");
    int cols1 = sc.nextInt();

    System.out.print("Enter number of rows for Matrix 2: ");
    int rows2 = sc.nextInt();
    System.out.print("Enter number of columns for Matrix 2: ");
    int cols2 = sc.nextInt();

    // Condition for multiplication
    if (cols1 != rows2) {
        System.out.println("Matrix multiplication is not possible.");
        System.out.println("Columns of Matrix 1 must be equal to rows of Matrix 2.");
        return;
    }

    int[][] matrix1 = new int[rows1][cols1];
    int[][] matrix2 = new int[rows2][cols2];
    int[][] product = new int[rows1][cols2];

    System.out.println("Enter elements of Matrix 1:");
    for (int i = 0; i < rows1; i++) {
        for (int j = 0; j < cols1; j++) {
            matrix1[i][j] = sc.nextInt();
        }
    }

    System.out.println("Enter elements of Matrix 2:");
    for (int i = 0; i < rows2; i++) {

```

```

        for (int j = 0; j < cols2; j++) {
            matrix2[i][j] = sc.nextInt();
        }
    }

    // Performing multiplication using three nested loops
    for (int i = 0; i < rows1; i++) {
        for (int j = 0; j < cols2; j++) {
            for (int k = 0; k < cols1; k++) {
                product[i][j] += matrix1[i][k] * matrix2[k][j];
            }
        }
    }

    System.out.println("Resultant Matrix (Product):");
    printMatrix(product);
}

/**
 * Finds the transpose of a single matrix.
 * The rows of the original matrix become the columns of the new matrix.
 */
public static void transposeMatrix(Scanner sc) {
    System.out.print("Enter number of rows: ");
    int rows = sc.nextInt();
    System.out.print("Enter number of columns: ");
    int cols = sc.nextInt();

    int[][] matrix = new int[rows][cols];
    int[][] transpose = new int[cols][rows]; // Dimensions are swapped

    System.out.println("Enter the elements of the Matrix:");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            matrix[i][j] = sc.nextInt();
        }
    }

    // Performing transpose
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            transpose[j][i] = matrix[i][j];
        }
    }

    System.out.println("Resultant Matrix (Transpose):");
    printMatrix(transpose);
}

/**
 * A helper method to print any 2D integer array in matrix format.
 */
public static void printMatrix(int[][] matrix) {
    for (int[] row : matrix) {
        for (int element : row) {
            System.out.print(element + "\t"); // Use tab for better spacing
        }
        System.out.println();
    }
}
}

```

--- Matrix Operations Menu ---

1. Addition
2. Subtraction
3. Multiplication
4. Transpose
5. Exit

Enter your choice (1-5): 1

Enter number of rows: 2

Enter number of columns: 2

Enter elements of Matrix 1:

1 3

4 6

Enter elements of Matrix 2:

2 6

7 0

Resultant Matrix (Sum):

3 9

11 6

--- Matrix Operations Menu ---

1. Addition
2. Subtraction
3. Multiplication
4. Transpose
5. Exit

Enter your choice (1-5): 2

Enter number of rows: 2

Enter number of columns: 2

Enter elements of Matrix 1 (Minuend):

1 5

6 7

Enter elements of Matrix 2 (Subtrahend):

2 5

9 0

Resultant Matrix (Difference):

-1 0

-3 7

--- Matrix Operations Menu ---

1. Addition
2. Subtraction
3. Multiplication
4. Transpose
5. Exit

Enter your choice (1-5): 3

Enter number of rows for Matrix 1: 2

Enter number of columns for Matrix 1: 1

Enter number of rows for Matrix 2: 2

Enter number of columns for Matrix 2: 2

Matrix multiplication is not possible.

Columns of Matrix 1 must be equal to rows of Matrix 2.

--- Matrix Operations Menu ---

1. Addition
2. Subtraction
3. Multiplication
4. Transpose
5. Exit

Enter your choice (1-5): 4

Enter number of rows: 2

Enter number of columns: 2

Enter the elements of the Matrix:

1 4

6 7

Resultant Matrix (Transpose):

1     6

4     7

--- Matrix Operations Menu ---

1. Addition

2. Subtraction

3. Multiplication

4. Transpose

5. Exit

Enter your choice (1-5): 5

Exiting program. Goodbye!