Experiment Number : 04 **Date: 04-02-26**

Santhosh TMA

24BCS247 CSE-A1

ARRAYS

Aim:

To understand and implement array operations in Java.

PRE LAB EXERCISE

QUESTIONS

✓ What is an array?

An array is a collection of elements of the same data type stored in consecutive memory locations.

✓ Why are arrays used?

Arrays are used to store multiple values using a single name and to easily access and process large amounts of data.

✓ What is the difference between array and variable?

- A **variable** stores only one value at a time.
- An **array** stores multiple values of the same data type under one name.

IN LAB EXERCISE

Objective:

To perform array operations using simple programs.

PROGRAMS:

1. Program to Read and Print Array Elements

Code:

```
import java.util.Scanner; public class
ReadPrintArray {     public static void
main(String[] args) { Scanner sc = new
```

```
Scanner(System.in); int[] arr = new
int[5];
        System.out.println("Enter 5 elements:");
for(int i = 0; i < 5; i++)              arr[i] =
sc.nextInt();
        System.out.println("Array elements are:");
for(int i = 0; i < 5; i++)
            System.out.print(arr[i] + " ");
            System.out.print(arr[i] + " ");


    }
}
```

OUTPUT:

Input: 10 20
30 40 50

Output:

Array elements:

10 20 30 40 50

```
Enter 5 elements:
10

20
30
40
50
Array elements are:
10 20 30 40 50
=== Code Execution Successful ===
```

2. Program to Find Sum of Array Elements

Code:

```
import java.util.Scanner; public class
SumArray {     public static void
main(String[] args) { Scanner sc = new
Scanner(System.in); int[] arr = new
int[5];         int sum = 0;
        System.out.println("Enter 5 elements:");
for(int i = 0; i < 5; i++)              arr[i] =
sc.nextInt();         for(int i = 0; i < 5; i++)
sum += arr[i];
        System.out.println("Sum = " + sum);
    }
}
```

OUTPUT:

Input:

1,2,3,4,5

Output:

Sum = 15

```
Enter 5 elements:
1
2
3
4

5
Sum = 15

=== Code Execution Successful ===
```

3. Program to Find Largest Element in an Array

Code:

```
import java.util.Scanner; public class
LargestElement {     public static void
main(String[] args) {         Scanner sc =
new Scanner(System.in);         int[] arr =
new int[5]; System.out.println("Enter 5
elements:"); for(int i = 0; i < 5; i++)
arr[i] = sc.nextInt();         int max = arr[0];
for(int i = 1; i < 5; i++)             if(arr[i] >
max)                 max = arr[i];
        System.out.println("Largest element = " + max);
    }
}
```

OUTPUT:

Input: 8 12

34 56 13

Output:

Largest element = 56

```
Enter 5 elements:
8
12
34
56
13
Largest element = 56

=== Code Execution Successful ===
```

4. Program to Reverse an Array

Code:

```
import java.util.Scanner;


public class ReverseArray {     public
static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int[] arr = new int[5];
        System.out.println("Enter 5 elements:");
        for(int i = 0; i < 5; i++)
arr[i] = sc.nextInt();
        System.out.println("Reversed array:");
for(int i = 4; i >= 0; i--)
            System.out.print(arr[i] + " ");
    }
}
```

OUTPUT:

Input:

2 4 6 8
10

Output:

Reversed array:

10 8 6 4 2

```
Enter 5 elements:
2
4
6
8
10
Reversed array:
10 8 6 4 2
=== Code Execution Successful ===
```

5. Program to Count Even and Odd Numbers

Code:

```
import java.util.Scanner; public class
EvenOddCount {     public static void
main(String[] args) {         Scanner sc =
new Scanner(System.in);         int[] arr =
new int[5];

        int even = 0, odd = 0;
        System.out.println("Enter 5 elements:");
        for(int i = 0; i < 5; i++)
arr[i] = sc.nextInt();
for(int i = 0; i < 5; i++) {
if(arr[i] % 2 == 0)
even++;             else
odd++;
        }

        System.out.println("Even = " + even);
        System.out.println("Odd = " + odd);
    }
```

}

OUTPUT:

Input: 1
3 5 2 4

Output:

Even = 2

Odd = 3

```
Enter 5 elements:
1
3
5
2
4
Even = 2
Odd = 3

=== Code Execution Successful ===
```

6. Program to Sort Array in Ascending Order

Code:

```
import java.util.Scanner; public class
SortArray {     public static void
main(String[] args) {         Scanner sc =
new Scanner(System.in);         int[] arr =
new int[5];         int temp;
        System.out.println("Enter 5 elements:");
for(int i = 0; i < 5; i++)             arr[i] =
sc.nextInt();         for(int i = 0; i < 5; i++) {
for(int j = i + 1; j < 5; j++) {
```

```
if(arr[i] > arr[j]) {                    temp = arr[i];
arr[i] = arr[j];                    arr[j] = temp;
                }
            }
        }
        System.out.println("Sorted array:");
for(int i = 0; i < 5; i++)
            System.out.print(arr[i] + " ");
    }
}
```

OUTPUT:

Input: 12 67
34 43 12

Output:

Sorted array:

12 12 34 43 67

```
Enter 5 elements:
12
67
34
43
12
Sorted array:
12 12 34 43 67
=== Code Execution Successful ===
```

7. Program to Find Second Largest Element

Code:

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

```
public class SecondLargest {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int[] arr = new int[5];

        System.out.println("Enter 5 elements:");
        for (int i = 0; i < 5; i++) {
            arr[i] = sc.nextInt();
        }

        int largest = Integer.MIN_VALUE;
        int second = Integer.MIN_VALUE;

        for (int i = 0; i < 5; i++) {
            if (arr[i] > largest) {
                second = largest;
                largest = arr[i];
            } else if (arr[i] > second && arr[i] != largest) {
                second = arr[i];
            }
        }

        if (second == Integer.MIN_VALUE) {
            System.out.println("No second largest element exists");
        } else {
            System.out.println("Second largest = " + second);
        }
    }
```

}

OUTPUT:

Input: 12 56
34 13 52

Output:

Second largest = 52

```
Enter 5 elements:
12
56
34
13
52
Second largest = 52

=== Code Execution Successful ===
```

8. Program for Matrix Addition (2D Array)

Code:

```
import java.util.Scanner; public class
MatrixAddition {     public static void
main(String[] args) {         Scanner sc =
new Scanner(System.in);         int[][] a =
new int[2][2];         int[][] b = new
int[2][2];         int[][] sum = new int[2][2];
        System.out.println("Enter elements of matrix A:");
for(int i = 0; i < 2; i++)             for(int j = 0; j < 2; j++)
a[i][j] = sc.nextInt();
        System.out.println("Enter elements of matrix B:");
for(int i = 0; i < 2; i++)             for(int j = 0; j < 2; j++)
b[i][j] = sc.nextInt();
```

```
        for(int i = 0; i < 2; i++)
for(int j = 0; j < 2; j++)
sum[i][j] = a[i][j] + b[i][j];
        System.out.println("Sum matrix:");
for(int i = 0; i < 2; i++) {            for(int
j = 0; j < 2; j++)
                System.out.print(sum[i][j] + " ");
            System.out.println();
        }
    }
}
```

OUTPUT:

Matrix A:

1 2

3 4

Matrix B:

5 6

7 8

Sum matrix:

6 8

10 12

```
Enter elements of matrix A:
1 2
3 4
Enter elements of matrix B:
5 6
7 8
Sum matrix:
6 8
10 12
```

POST LAB EXERCISE

✓ Why is array indexing usually started from zero instead of one?

Because the index represents the offset from the starting memory address, and the first element has offset 0.

✓ What happens if we try to access an array element outside its declared size? It causes an error or undefined behavior and may crash the program or give garbage values.

✓ How does memory allocation differ for static arrays and dynamic arrays?

- **Static arrays**: Memory is allocated at compile time and size is fixed.
- **Dynamic arrays**: Memory is allocated at runtime and size can be changed.

✓ Why is searching faster in arrays compared to linked lists?

Because arrays allow direct access using index, while linked lists require sequential traversal.

✓ What is the difference between contiguous and non-contiguous memory allocation?

- **Contiguous memory**: Elements are stored in continuous memory locations (arrays).
- **Non-contiguous memory**: Elements are stored at different memory locations linked together (linked lists).

Result:

Thus the array operations were executed successfully.

ASSESSMENT

Description	Max Marks	Marks Awarded
Pre Lab Exercise	**5**	
In Lab Exercise	**10**	
Post Lab Exercise	**5**	
Viva	**10**	
Total	**30**	
	Faculty Signature	