

## ARRAYS

### Aim:

To understand and implement array operations in Java.

### PRE LAB EXERCISE

#### QUESTIONS

##### 1. What is an array?

- An array is used to store multiple values of the same data type.
- All elements are stored under one variable name.
- Each element is accessed using an index value.

##### 2. Why are arrays used?

- Arrays help to store large amounts of data efficiently.
- They reduce code complexity by using a single variable.
- Data can be easily accessed and processed using indexes.

##### 3. What is the difference between array and variable?

#### ARRAY:

- An array is used to store multiple values in a single variable.
- All elements in array must be of the same data type.
- Array elements are accessed using an index value.

#### VARIABLE:

- A variable is used to store only one value.
- A variable occupies a single memory location.
- A variable is accessed directly by its 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] + " ");  
  
    }  
}
```

### OUTPUT:

#### Input:

10 20 30 40 50

#### Output:

Array elements are:

10 20 30 40 50

### OUTPUT:

```
Storage\Volume 490E\2A1190E2E405B0C9291258\ReadPrintArray.java\jul_1  
Enter 5 elements:  
1  
2  
3  
4  
5  
Array elements are:  
1 2 3 4 5  
PS C:\Users\Paramasivam\OneDrive\oops_lab\java\src>
```

## 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:

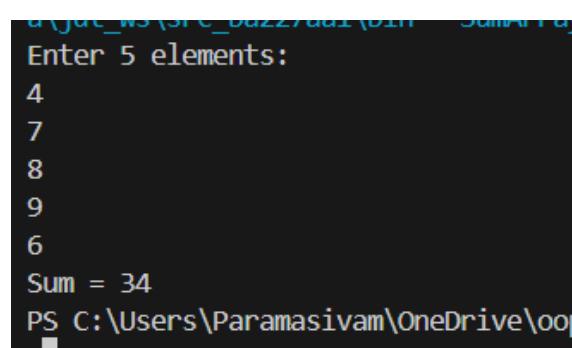
#### Input:

5 10 15 20 25

#### Output:

Sum = 75

### OUTPUT:



```
PS C:\Users\Paramasivam\OneDrive\o...  
Enter 5 elements:  
4  
7  
8  
9  
6  
Sum = 34  
PS C:\Users\Paramasivam\OneDrive\o...
```

### 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:

12 45 23 9 30

##### Output:

Largest element = 45

#### OUTPUT:

```
Storage (Volume C) 2017-03-26 09:30:59 1230 (V2)
a\jdt_ws\src_ba227aa1\bin' 'LargestElement'
Enter 5 elements:
30
45
89
90
100
Largest element = 100
```

#### 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:

1 2 3 4 5

##### Output:

Reversed array:

5 4 3 2 1

##### OUTPUT:

```
Enter 5 elements:
56
78
90
24
35
Reversed array:
35 24 90 78 56
PS C:\Users\Paramasivam\OneDrive\oops_lab
```

## 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:

2 7 4 9 10

#### Output:

Even = 3

Odd = 2

OUTPUT:

```
Enter 5 elements:
35
48
79
56
23
Even = 2
Odd = 3
PS C:\Users\Paramasivam\OneDrive\Documents\NetBeansProjects\JavaPrograms\src\java\com\javatpoint\array\EvenOdd\
```

## 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:**

45 12 78 23 9

**Output:**

Sorted array:

9 12 23 45 78

**OUTPUT:**

```
Enter 5 elements:  
45  
67  
89  
02  
12  
Sorted array:  
2 12 45 67 89  
PS C:\Users\Paramasivam\OneDr:
```

**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 = arr[0];
int second = arr[0];
for(int i = 0; i < 5; i++) {
    if(arr[i] > largest) {
        second = largest;
        largest = arr[i];
    }
}
System.out.println("Second largest = " + second);
}
```

**OUTPUT:**

**Input:**

10 45 23 89 67

**Output:**

Second largest = 67

**OUTPUT:**

```
Enter 5 elements:
20
903
344
234
90
Second largest = 344
```

**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

**OUTPUT:**

```
Enter elements of matrix A:  
5  
8  
9  
4  
Enter elements of matrix B:  
3  
6  
4  
0  
Sum matrix:  
8 14  
13 4  
PS C:\Users\Paramasivam\OneDrive\oops_lab\ja
```

## POST LAB EXERCISE

- 1. Why is array indexing usually started from zero instead of one?**
  - Array index represents the offset from the base memory address.
  - The first element has zero offset, so indexing starts at 0.
  - This makes address calculation faster and simpler for the system.
- 2. What happens if we try to access an array element outside its declared size?**
  - It may cause a runtime error or array index out of bounds exception.
  - The program may crash or behave unpredictably.
  - It can lead to data corruption or security issues.
- 3. How does memory allocation differ for static arrays and dynamic arrays?**

- Static arrays are allocated memory at compile time.
- Dynamic arrays are allocated memory at runtime.
- Static array size is fixed, while dynamic array size can be changed.

**4. Why is searching faster in arrays compared to linked lists?**

- Arrays allow direct access using index values.
- Elements are stored in contiguous memory locations.
- Linked lists require sequential traversal, which takes more time.

**5. What is the difference between contiguous and non-contiguous memory allocation?**

- Contiguous memory stores elements in adjacent memory locations.
- Non-contiguous memory stores elements in separate memory locations.
- Contiguous allocation allows faster access, while non-contiguous offers better flexibility.

**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	
<b>Total</b>	<b>30</b>	
<b>Faculty Signature</b>		