

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 contiguous memory locations and accessed using a single name with an index.

- ✓ Why are arrays used?

Arrays are used to:

- Store multiple values of the same type in one structure
- Access data easily using index numbers
- Reduce the need for many separate variables
- Make programs more organized and efficient
- Simplify operations like sorting, searching, and iteration

- ✓ What is the difference between array and variable?

Variable

Stores only one value at a time

Array

Stores multiple values

Variable	Array
Has a single memory location	Uses multiple continuous memory locations
Accessed by name	Accessed by name with index (e.g., arr[0])
Example: int a = 10;	Example: int a[5] = {1,2,3,4,5};

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
```

```
J ReadPrintArray.java > ...
1 import java.util.Scanner;
2 public class ReadPrintArray {
3     Run | Debug
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         int[] arr = new int[5];
7         System.out.println("Enter 5 elements:");
8         for(int i = 0; i < 5; i++) {
9             arr[i] = sc.nextInt();
10            System.out.println("Array elements are:");
11            for(int i = 0; i < 5; i++)
12                System.out.print(arr[i] + " ");
13        }
14 }
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\sachita\OneDrive\Desktop\java> & 'C:\Program Files\+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\sachita\Applorange\39cd8d1a6238b4807aeb3f0b19e663cf\redhat.java\jdt_ws\java_0'
Enter 5 elements:
10
20
30
40
50
Array elements are:
10 20 30 40 50
PS C:\Users\sachita\OneDrive\Desktop\java>
```

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:

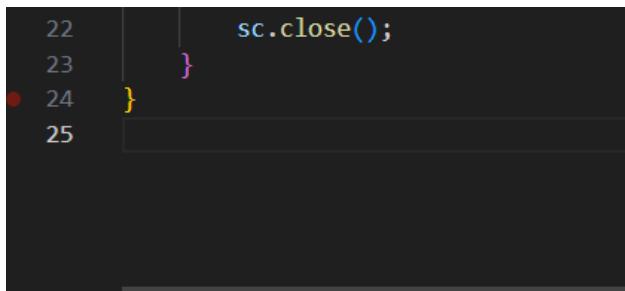
Input:

5 10 15 20 25

Output:

Sum = 75

```
1 import java.util.Scanner;
2
3 public class SumArray {
4     Run|Debug
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
8         int[] arr = new int[5];
9         int sum = 0;
10
11         System.out.println("Enter 5 elements:");
12
13         for (int i = 0; i < 5; i++) {
14             arr[i] = sc.nextInt();
15         }
16
17         for (int i = 0; i < 5; i++) {
18             sum = sum + arr[i];
19         }
20
21         System.out.println("Sum = " + sum);
22         sc.close();
23     }
24 }
25
```



The screenshot shows a Java code editor with the following code:

```
22     sc.close();
23 }
24 }
25
```

A red circular icon is positioned next to the closing brace of line 24, indicating a syntax error. The code editor has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL.

TERMINAL

```
PS C:\Users\sachita\OneDrive\Desktop\java>
PS C:\Users\sachita\OneDrive\Desktop\java> cd 'C:\Program Files\Java\jdk-21.0.10\bin\java.exe'
PS C:\Users\sachita\OneDrive\Desktop\java> cd 'C:\Users\sachita\AppData\Roaming\Code\User\workspaces\jdt_ws\java_6a99f975\bin'
PS C:\Users\sachita\AppData\Roaming\Code\User\workspaces\jdt_ws\java_6a99f975\bin> 'SumArray'
Enter 5 elements:
5
10
15
20
25
Sum = 75
```

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

```
1 import java.util.Scanner;
2 public class LargestElement {
3     Run | Debug
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         int[] arr = new int[5];
7         System.out.println("Enter 5 elements:");
8         for(int i = 0; i < 5; i++) {
9             arr[i] = sc.nextInt();
10            int max = arr[0];
11            for(int i = 1; i < 5; i++) {
12                if(arr[i] > max)
13                    max = arr[i];
14            }
15        }
16    }
17 }
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\sachita\OneDrive\Desktop\java> & 'C:\Program Files\Java:jdk-11.0.11+9-b161-7597953\bin\javac.exe' -cp 'C:\Users\sachita\AppData\Roaming\39cd8d1a6238b4807aeb3f0b19e663cf\redhat.java\jdt_ws\java_6a991\src' LargestElement.java
:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\sachita\AppData\Roaming\39cd8d1a6238b4807aeb3f0b19e663cf\redhat.java\jdt_ws\java_6a991\src'
Enter 5 elements:
12
45
23
9
30
Largest element = 45
```

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

```
1 import java.util.Scanner;
2
3 public class ReverseArray {
4     Run|Debug
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7         int[] arr = new int[5];
8         System.out.println("Enter 5 elements:");
9         for(int i = 0; i < 5; i++)
10            arr[i] = sc.nextInt();
11         System.out.println("Reversed array:");
12         for(int i = 4; i >= 0; i--)
13             System.out.print(arr[i] + " ");
14     }
15 }
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\sachita\OneDrive\Desktop\java> & 'C:\Program File
:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\sachita\
orange\39cd8d1a6238b4807aeb3f0b19e663cf\redhat.java\jdt_ws\java
Enter 5 elements:
1
2
3
4
5
Reversed array:
5 4 3 2 1
```

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
```

```
1 import java.util.Scanner;
2
3 public class EvenOddCount {
4     Run | Debug
5     public static void java.io.PrintStream.println(String x)
6         Prints a String and then terminates the line. This method behaves as though it invokes print(String) and then println().
7         • Parameters:
8             ◦ x The String to be printed.
9             Source: Java 21.0.10 (module: java.base)
10            System.out.println(x: "Enter 5 elements:");
11
12            for (int i = 0; i < 5; i++) {
13                arr[i] = sc.nextInt();
14            }
15
16            for (int i = 0; i < 5; i++) {
17                if (arr[i] % 2 == 0) {
18                    even++;
19                } else {
20                    odd++;
21                }
22            }
23
24        System.out.println("Even = " + even);
25        System.out.println("Odd = " + odd);
26
27        sc.close(); // closes the Scanner to
28    }
29
30
31
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\sachita\OneDrive\Desktop\java> & 'C:\Program
:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\users\sac
orange\39cd8d1a6238b4807aebe3f0b19e663cf\redhat.java\jdt_ws
Enter 5 elements:
2
7
4
9
10
Even = 3
Odd = 2
```

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

```
1 import java.util.Scanner;
2
3 public class SortArray {
4     Run | Debug
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
8         int[] arr = new int[5];
9         int temp;
10
11        System.out.println("Enter 5 elements:");
12
13        for (int i = 0; i < 5; i++) {
14            arr[i] = sc.nextInt();
15        }
16
17        for (int i = 0; i < 5; i++) {
18            for (int j = i + 1; j < 5; j++) {
19                if (arr[i] > arr[j]) {
20                    temp = arr[i];
21                    arr[i] = arr[j];
22                    arr[j] = temp;
23                }
24            }
25        }
26    }
27}
```

```
3  public class SortArray {
4      public static void main(String[] args) {
23         }
24     }
25
26     System.out.println("Sorted array:");
27     for (int i = 0; i < 5; i++) {
28         System.out.print(arr[i] + " ");
29     }
30
31     sc.close();
32 }
33
34
35
36
37
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\sachita\OneDrive\Desktop\java> & 'C:\Program Files\Java\jdk-17.0.2\bin\javac.exe' -cp 'C:\Users\sachita\OneDrive\Desktop\java\src' 'C:\Users\sachita\OneDrive\Desktop\java\src\SortArray.java'
Enter 5 elements:
45
12
78
23
9
Sorted array:
9 12 23 45 78
```

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

```
1 import java.util.Scanner;
2
3 public class SecondLargest {
4     Run | Debug
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
8         int[] arr = new int[5];
9
10        System.out.println("Enter 5 elements:");
11        for (int i = 0; i < 5; i++) {
12            arr[i] = sc.nextInt();
13        }
14
15        int largest = arr[0];
16        int second = arr[0];
17
18        for (int i = 1; i < 5; i++) {
19            if (arr[i] > largest) {
20                second = largest;
21                largest = arr[i];
22            }
23            else if (arr[i] > second && arr[i] != largest) {
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
23             second = arr[i];
24         }
25     }
26
27     System.out.println("Second largest = " + second);
28
29     sc.close();
30 }
31
32 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\sachita\OneDrive\Desktop\java> & 'C:\Program Files\Java
:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\sachita\AppData\Ro
mage\39cd8d1a6238b4807aeb3f0b19e663cf\redhat.java\jdt_ws\java_6a99f
Enter 5 elements:
10
45
23
89
67
Second largest = 67
```

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

```
1 import java.util.Scanner;  
2  
3 public class MatrixAddition {  
4     Run | Debug  
5     public static void main(String[] args) {  
6         Scanner sc = new Scanner(System.in);  
7  
8         int[][] a = new int[2][2];  
9         int[][] b = new int[2][2];  
10        int[][] sum = new int[2][2];  
11  
12        System.out.println("Enter elements of matrix A:");  
13        for (int i = 0; i < 2; i++)  
14            for (int j = 0; j < 2; j++)  
15                a[i][j] = sc.nextInt();  
16  
17        System.out.println("Enter elements of matrix B:");  
18        for (int i = 0; i < 2; i++)  
19            for (int j = 0; j < 2; j++)  
20                b[i][j] = sc.nextInt();  
21  
22        for (int i = 0; i < 2; i++)  
23            for (int j = 0; j < 2; j++)
```

```
22     for (int j = 0; j < 2; j++)  
23         sum[i][j] = a[i][j] + b[i][j];  
24  
25     System.out.println("Sum matrix:");  
26     for (int i = 0; i < 2; i++) {  
27         for (int j = 0; j < 2; j++)  
28             System.out.print(sum[i][j] + " ");  
29         System.out.println();  
30     }  
31  
32     sc.close();  
33 }  
34 }
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\sachita\OneDrive\Desktop\java> & 'C:\Program Files\Java\jdk-11.0.1\bin\javac.exe' '+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\sachita\OneDrive\Desktop\java\src' 'MatrixAddition.java'
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?
 - Arrays store elements in contiguous memory.
 - The index represents the offset from the starting address.
 - First element → offset 0 → base address + 0.
 - This makes address calculation simple and fast in programming languages.

- ✓ What happens if we try to access an array element outside its declared size?
 - It causes an error at runtime.

- In Java: `ArrayIndexOutOfBoundsException` occurs.

- In C/C++: may access garbage values or crash the program.

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

Static	Dynamic
Array	Array
Size fixed at compile time	Size decided at runtime
Memory allocated before execution	Memory allocated during execution

Cannot be
resized

Example: int
arr[5];

Can be
resized

Example: new int[n]; in
Java

- ✓ Why is searching faster in arrays compared to linked lists?
- ✓ What is the difference between contiguous and non-contiguous memory allocation?

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		