Day 6

Arrays

What is an Array?

An array is a **collection of elements** that share the **same data type**.

Imagine you want to store multiple values of the same kind, like a list of numbers. Instead of creating separate variables for each value, you can use an array to store all the values in one place. This makes your code more organized and easier to manage.

Types of Arrays

In Java, there are mainly two types of arrays:

1. Single/One-Dimensional Array
2. Double/Two-Dimensional Array

1. Single/One-Dimensional Array

A single-dimensional array is like a simple list of elements, where each element is accessed using a single index.

Approach 1 : Declaration and Initialization Separately

- **Declaration of Array**:

```
int a[] = new int[5]; // An array of 5 integers
```

or

```
int []a = new int[5];
```

- **Initialization of Array**:

```
a[0] = 10;
a[1] = 20;
a[2] = 30;
a[3] = 40;
a[4] = 50;
```

Approach 2: Declaration and Initialization Together

```
int a[] = {10, 20, 30, 40, 50};
```

2. Double/Two-Dimensional Array

A two-dimensional array is like a table or a matrix, where you have rows and columns. Each element is accessed using two indexes: one for the row and one for the column.

Approach 1 : Declaration and Initialization Separately

- **Declaration of Array**:

```
int[][] a = new int[2][2];       // A 2x2 matrix (2D array)

or

int a[][] = new int[2][2];

or

int []a[] = new int[2][2];
```

- **Initialization of Array**:

```
a[0][0] = 10;

a[0][1] = 20;

a[1][0] = 30;

a[1][1] = 40;
```

Approach 2: Declaration and Initialization Together

```
int a[][] = {
            {10, 20},
            {30, 40},
            {50, 60}
          };
```

- Use **Approach 1** when you need to assign values dynamically or incrementally.
- Use **Approach 2** when you have a fixed set of known values and want a concise way to declare and initialize the array in one step.

Summary

- **Single-Dimensional Array**: A simple list of elements, accessed using a single index.
- **Two-Dimensional Array**: A matrix or table, accessed using two indexes (row and column).

Object type:

An **Object** in Java is the root class of the Java class hierarchy, an Object type can hold any type of data(primitives, objects etc..)

Object type variable:

```
Object x;

x=10;   //valid

x=10.5;  //valid

x=”welcome”;   //valid

x=true;  //valid

x=‘A’; //valid
```

Object type array:

```
Object  x[]=new Object[5];

x[0]=10;   //valid

x[1]=10.5;  //valid

x[2]=”welcome”;   //valid

x[3]=true;  //valid
```

Lab Assignments

Single dimensional array:

1. Write a program to find the largest element in a given array.

Sample Input:

int[] numbers = {4, 2, 7, 1, 9};

Expected Output:

Largest Element: 9

2. Write a program to find the smallest element in a given array.

Sample Input:

int[] numbers = {4, 2, 7, 1, 9};

Expected Output:

Smallest Element: 1

3. Write a program to calculate the sum of all elements in a given array.

Sample Input:

int[] numbers = {4, 2, 7, 1, 9};

Expected Output:

Sum of Elements: 23

4. Write a program to calculate the average of all elements in a given array.

Sample Input:

int[] numbers = {4, 2, 7, 1, 9};

Expected Output:

Average of Elements: 4.6

5. Write a program to reverse the elements of a given array.

Sample Input:

int[] numbers = {4, 2, 7, 1, 9};

Expected Output:

Reversed Array: 9 1 7 2 4

6. Write a program to check if a given array contains a specific element (Search an element in array).

Sample Input:

int[] numbers = {4, 2, 7, 1, 9};

int target = 7;

Expected Output:

Element 7 found: true

7. Write a program to count how many times a specific element appears in a given array.

Sample Input:

int[] numbers = {4, 2, 7, 2, 9, 2};

int target = 2;

Expected Output:

Element 2 occurs 3 times.

8. Write a program to remove duplicate elements from a given array.

Sample Input:

int[] numbers = {4, 2, 7, 2, 9, 4};

Expected Output:

Array without duplicates: 2 4 7 9

9. Write a program to find the index of a specific element in a given array.

Sample Input:

int[] numbers = {4, 2, 7, 1, 9};

int target = 7;

Expected Output:

Index of 7: 2

10. Write a program to print all elements of a given array in a single line.

Sample Input:

int[] numbers = {4, 2, 7, 1, 9};

Expected Output:

Array elements: 4 2 7 1 9

11. Write a program to copy the elements of one array into another.

Sample Input:

int[] original = {4, 2, 7, 1, 9};

Expected Output:

Copied Array: 4 2 7 1 9

12. Write a program to sort the elements of a given array in ascending order (Sorting an array).

Sample Input:

int[] numbers = {4, 2, 7, 1, 9};

Expected Output:

Sorted Array: 1 2 4 7 9

13. Write a program to find the sum of all even numbers in a given array.

Sample Input:

int[] numbers = {4, 2, 7, 1, 9};

Expected Output:

Sum of Even Numbers: 6

Two-dimensional array:

1. Write a program to print a two-dimensional array in a grid format.

Sample Input:

int[][] array = {

{1, 2, 3},

{4, 5, 6},

{7, 8, 9}

};

Expected Output:

Two-Dimensional Array:

1 2 3

4 5 6

7 8 9

Additional Notes: Ensure that the output is displayed in a tabular format.

2. Write a program to find and print the sum of all elements in a two-dimensional array.

Sample Input:

int[][] array = {

{1, 2, 3},

{4, 5, 6},

{7, 8, 9}

};

Expected Output:

Sum of All Elements: 45

3. Write a program to find and print the maximum element in a two-dimensional array.

Sample Input:

int[][] array = {

{1, 2, 3},

{4, 5, 6},

{7, 8, 9}

};

Expected Output:

Maximum Element: 9