

# Arrays Programs

## Traversing :

- ✓ 1. WAP to print all the elements of 1-D, 2-D, 3-D & 4-D arrays.  
(Also 4 ways to traverse the elements) **(Todays Topic)**

## Sorting :

1. Sorting of an array in increasing or decreasing order using :
  - 1.1 Bubble Sort
  - 1.2 Selection Sort
  - 1.3 Insertion Sort
  - 1.4 Merge Sort
  - 1.5 Quick Sort
  - 1.6 Heap Sort
2. Write a Java program to sort a string array.

# Arrays Programs

## Searching :

1. WAP to test if an array contains a specific value using :
  - 1.1 Linear/Sequential Search.
  - 1.2 Binary Search
2. WAP to find the maximum and minimum value in an array.
3. WAP to find the second largest and smallest element in an array.
4. WAP to find kth smallest and largest element in an array.
5. WAP to find the duplicate values of an array of integer values.
6. WAP to find the duplicate values of an array of string values.

# Arrays Programs

## Searching :

7. WAP to find the first duplicate or repeating element in an integer array.
8. There is an array with every element repeated twice except one.  
Find that element?
9. WAP to find the number of even and odd integers in a given array of integers.
10. WAP to find the missing number in integer array of 1 to 10?
11. WAP to find the duplicate elements between two arrays (integer values).
12. WAP to find the duplicate elements between two arrays (string values).

# Arrays Programs

## Searching :

13. WAP to find duplicate elements from three sorted arrays.
14. WAP to find length of longest consecutive sequence in array of integers.
15. How to find sub array with maximum sum in an array of positive and negative number?

## Insertion :

1. WAP to insert an element (specific position) into an array.

# Arrays Programs

## Deletion :

1. WAP to remove a specific element from an array.
2. Write a Java program to remove duplicate elements from an array and then return the new length of the array.

For ex : Sample array: [20, 20, 30, 40, 50, 50, 50]

After removing the duplicate elements the program should return 4 as the new length of the array.

## Merging :

1. WAP to merge the elements of two arrays in one array.

# Arrays Programs

## Others :

1. WAP to sum values of an array.
2. WAP to find the index of an array element.
3. WAP to copy an array by iterating the array.
4. WAP to add two matrices of the same size.
5. WAP to check if an array of integers without 0 and -1.
6. WAP to check if two arrays are equal or not.
7. WAP to convert an array to ArrayList.
8. WAP to convert an ArrayList to an array.

# EPISODE - 3

## 1. Introduction To Arrays

- 1.1 What is Array
- 1.2 Features Of Arrays
- 1.3 Advantages Of Arrays
- 1.4 Disadvantages Of Arrays

## 2. Types Of Arrays :

- 2.1 Single Dimensional Arrays :-

- 1. 1-D Array

- 2.2 Multi Dimensional Arrays :-

- 2. 2-D Array
  - 3. 3-D Array

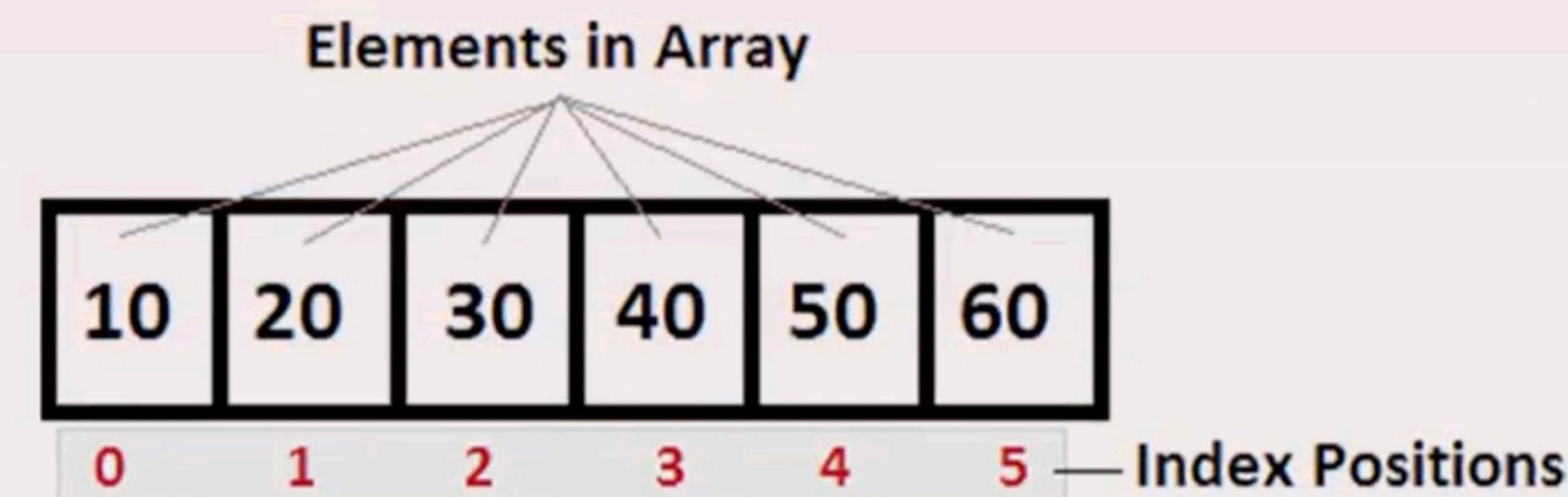
## 3. Array Declaration, Creation and Initialisation

## What is Array ?

- An array is an object that holds a fixed number of values of homogeneous or similar data-type.
- Or say An Array is a Data Structure where we store similar elements.
- The length of an array is assigned when the array is created and After creation, its length is fixed.

- For example : `int a[ ]=new int[6];`

It will create an array of length 6 and index value will always start from 0.



## Features Of An Array :

- A Java array variable can be declared like other variables with [ ] after the data type.
- The variables in the array are ordered and each have an index beginning from 0.
- In Java, Arrays are objects, and thus they occupy memory in ‘Heap Area’.
- The direct superclass of an array type is Object.
- They are always created at runtime.
- The length of an array can be find by using member ‘length’. This is different from C/C++ where we find length using sizeof.
- The elements of array are stored in consecutive memory locations.

## **Advantages Of An Array :**

- **Arrays are used to store multiple data items of same type by using only single name.**
- **We can access any element randomly by using indexes provided by arrays.**
- **Arrays can be used to implement other data structures like linked lists, stacks, queues, trees, graphs etc.**
- **Primitive type to wrapper classes object conversion will not happen so it is fast.**

## **Disadvantages Of An Array :**

- **Fixed Size** : We need to mention the size of the array, thus they have fixed size. When array is created, size cannot be changed.
- **Memory Wastage** : There is a lot of chance of memory wastage. Suppose we create an array of length 100 but only 10 elements are inserted, then 90 blocks are empty and thus memory wasted.
- **Strongly Typed** : Array stores only similar data type, thus strongly typed.
- **Reduce Performance** : The elements of array are stored in consecutive memory locations, thus to delete an element in an array we need to traverse through out the array so this will reduce performance.
- **No Methods** : Arrays does not have add or remove methods.

# Single Dimensional (1-D)

## Declaration Of An Array :

- Diferent Ways of Declaration of Arrays are :
  1. int[ ] a;
  2. int [ ]a;
  3. int[ ]a;
  4. int a[ ];
- Most Preferred Array Declaration is ‘int[ ] a;’, because here ‘a’ is one dimensional int array, thus name is clearly separated with type.
- We cannot provide size at the time of array declaration i.e. ‘int[3] a;’ or ‘int a[3];’ statement is incorrect.
- Note that, there is difference between below two statements :  
`int[ ] a,b; // here 'a' and 'b' both are arrays.`  
`int a[ ], b; // here 'a' is array and 'b' is simple int type variable, not an array.`

# Single Dimensional (1-D)

## Creation Of An Array :

- We can create an array after declaration as follows :

```
int[ ] a; //array declaration
```

```
a=new int[3]; // array creation
```

- It is compulsory to declare the size of an array at the time of creation.

- We can declare and create array within a single line as follows :

```
int[ ] a=new int[3];
```

- If we declare size of an array as ‘0’ i.e. ‘int[ ] a=new int[0];’, then program will successfully compile and execute.

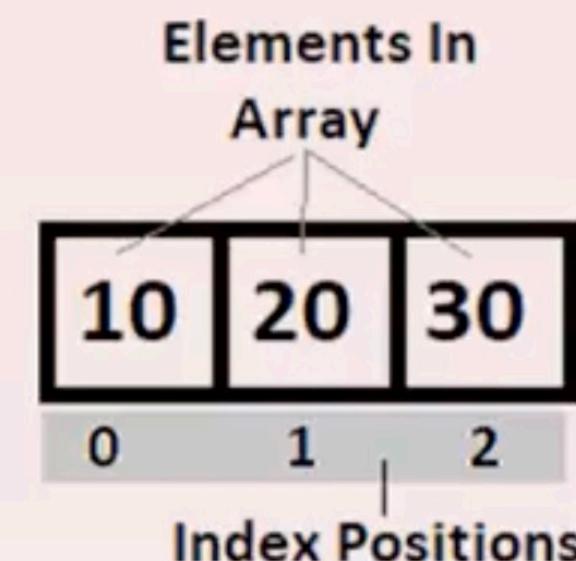
- If we declare size of an array as negative i.e. ‘int[ ] a=new int[-3];’, then the program will compile successfully but when we run the program, it will throw ‘NegativeArraySizeException’ exception.

# Single Dimensional (1-D)

## Initialization Of An Array :

- We can create an array after declaration as follows :

```
int[ ] a=new int[3]; //array declaration and creation  
a[0]=10; //array initialization at 0 index position  
a[1]=20;  
a[2]=30;
```



- If we initialize array at index position 3 or more i.e. 'a[3]=40;', it will throw 'ArrayIndexOutOfBoundsException' exception, so we cannot initialize an array more than its size.
- We can declare, create and initialize array within a single line as follows :
  1. `int[ ] a={10,20,30};`
  2. `int[ ] a=new int[ ]{10,20,30};`

# Single Dimensional (1-D)

## Retrieve Elements From An Array :

- An array given i.e. ‘int[ ] a={10,20,30};’, now we can print elements of an array by two ways, which are as follows :

### Way 1 : (using for loop)

```
for(int i=0;i<a.length;i++)  
{  
    System.out.println(a[i]);  
}
```

### Way 2 : (using for-each loop)

```
for(int i:a)  
{  
    System.out.println(i);  
}
```

# Multi-Dimensional (2-D)

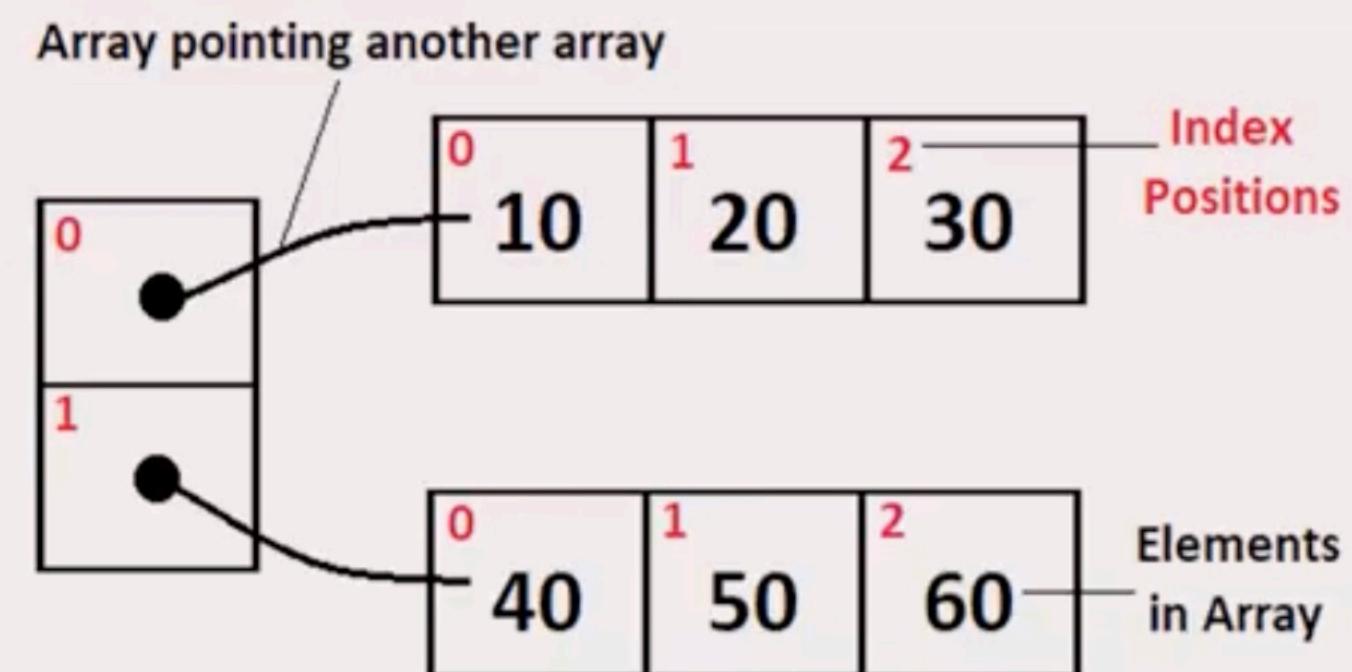
## What is Multi-Dimensional Array :

- An array having multiple rows or columns is known as multi-dimensional array.
- These are also known as array of arrays because array is present in another array.
- There are two types of multi-dimensional array :
  1. 2-D Array
  2. 3-D Array
- We can represent 2-D multi-dimensional array as follows :

Columns			Index Positions
0,0	0,1	0,2	
Rows	10	20	30
1,0	40	50	60
1,1			
1,2			

Elements in Array

OR



## Multi-Dimensional (2-D)

### Declaration Of 2-D Multi-Dimensional Array :

- Diferent Ways of Declaration of Arrays are :

1. int[ ][ ] a;	2. int [ ][ ]a;	3. int[ ][ ]a;
4. int a[ ][ ];	5. int[ ] a[ ];	

- Most Preferred Declaration is ‘int[ ][ ] a;’, because here ‘a’ is two dimensional int array, thus name is clearly separated with type.

- We cannot provide size at the time of array declaration i.e. ‘int[2][3] a;’ or ‘int a[2][3];’, this type of any statement is incorrect.

- Note that, there is difference between below statements :

int[ ][ ] a,b; // here ‘a’ and ‘b’ both are 2-D Arrays.

int[ ] a[ ], b; // here ‘a’ is 2-D and ‘b’ is 1-D Array.

int[ ] a[ ], b[ ]; // ‘a’ and ‘b’ both are 2-D Array.

int[ ][ ]a, [ ] b; // compile time error.

int[ ][ ]a, b[ ]; // ‘a’ is 2-D and ‘b’ is 3-D Array.

# Multi-Dimensional (2-D)

## Creation Of 2-D Multi-Dimensional Array :

- We can create an array after declaration as follows :

```
int[ ][ ] a; //array declaration  
a=new int[2][3]; // array creation
```

Matrix Array Creation

```
int[ ][ ] a; //array declaration  
a=new int[2][ ]; //array creation  
a[0]=new int[4];  
a[1]=new int[3];
```

Jagged Array Creation

- It is compulsory to declare the size of an array at the time of creation.
- We can declare and create 2-D array within a single line as follows :

```
int[ ][ ] a=new int[2][3];
```

Matrix Array Creation

```
int[ ][ ] a=new int[2][ ];  
a[0]=new int[4];  
a[1]=new int[3];
```

Jagged Array Creation