

Module no : 6

## Arrays

15/10/2023

### Agenda :

- ① Definition and Syntax
- ② Input / Output
- ③ for - each loop

## Arrays :-

### Definition :-

Java array is a data structure that stores data of the same type in a sequential manner. An array takes a contiguous section of the memory.

### Declaring an array :-

```
class Main {
```

```
    public static void main (String [] args) {
```

```
        int [] arr1 = {4, 8, 9, 1, 0}; // Array with initial values
```

(Size determined automatically).

```
        int [] arr2 = new int [5]; // Empty array with size = 5.
```

// Initially all values are 0.

```
        int [] arr3 = new int [5]; // Alternate declarations.
```

```
        int arr4 [] = new int [5];
```

```
        int [] arr5 = new int [5];
```

```
}
```

```
}
```

## Case Study : Class Room Average :-

Given the number marks of students in a classroom, calculate the average marks. Marks = [50, 45, 82, 76, 92, 83];

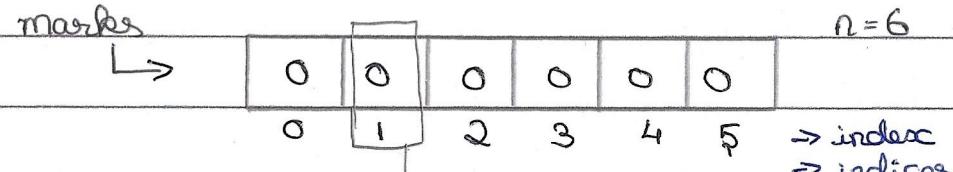
### Syntax :-

```
int [] marks = new int [6];
```

①      ②      ③      ④

- ① what data type that the array holds [int, String, double, float]
- ② name of the array
- ③ allocates new memory for array
- ④ size of memory which is need to be allocated.  $\text{marks}[6] \Rightarrow \boxed{\phantom{0}}_0 \boxed{\phantom{0}}_1 \boxed{\phantom{0}}_2 \boxed{\phantom{0}}_3 \boxed{\phantom{0}}_4 \boxed{\phantom{0}}_5$

### Memory allocation :-



- contains 'n' buckets
- Each bucket is capable for holding integer only. [int[ ]marks]

→ To display the value of marks in 1<sup>st</sup> index  
⇒ `System.out.println(marks[1]);`

To update the value in specific index say at index 4 :-

$\Rightarrow \text{marks}[4] = 20$ ; //updated array  $\Rightarrow$ 

0	0	0	0	20	0
0	1	2	3	4	5

similarly

$\Rightarrow \text{marks}[0] = 35$ ; //updated array  $\Rightarrow$ 

35	0	0	0	20	0
0	1	2	3	4	5

\* arrays are contiguous (i.e.,) 6 blocks are contiguously inside our memory

\* if we use "System.out.println(marks);" it will result in displaying memory location instead of array.

\* "marks.length" results in size of array.

\* "System.out.println(~~new~~ Main.~~to~~.toString(marks));" used to display whole array.

## Arrays in Java:

### Program:

```
package com.company;

import java.util.*;

public class Main {
    public static void main(String[] args) {
        int[] marks_1 = new int[6];
        marks_1[0] = 50;
        marks_1[1] = 45;
        marks_1[2] = 82;
        marks_1[3] = 76;
        marks_1[4] = 92;
        String [] subjects = {"tamil","english","maths","science","social science"};
        //also we can able to use different data types too
        System.out.println(marks_1[3]);
        System.out.println(marks_1.length);
        System.out.println(subjects.length);
        System.out.println(subjects[4]);
    }
}
```

### Output:

```
"C:\Program Files\Java\jdk-21\bin\java.exe"
76
6
5
social science

Process finished with exit code 0
```

---

## Array - Input Output:

Program :-

```
import java.util.*;  
class Main {  
    public static void main (String args[]) {  
        Scanner sc = new Scanner (System.in);  
        int [ ] arr = new int [5];  
        for (int i=0 ; i < arr.length ; i++) {  
            arr [i] = sc.nextInt();  
        }  
        for (int i=0 ; i < arr.length ; i++) {  
            System.out.println ("At index " + i + " => " + arr[i]);  
        }  
    }  
}
```

## Arrays in Java:

### Program:

```
package com.company;
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the length of the array: ");
        int n = sc.nextInt();
        int[] arr = new int[n];
        System.out.println("Enter the elements: ");
        for (int i = 0; i < arr.length; i++) {
            arr[i] = sc.nextInt();
        }
        for (int i = 0; i < arr.length; i++) {
            System.out.println("At the index "+i+" ==> "+arr[i]);
        }
    }
}
```

### Output:

```
"C:\Program Files\Java\jdk-21\bin\java.exe"
Enter the length of the array:
5
Enter the elements:
64
4986
456
799
346
At the index 0 ==> 64
At the index 1 ==> 4986
At the index 2 ==> 456
At the index 3 ==> 799
At the index 4 ==> 346

Process finished with exit code 0
```

## Linear Search Algorithm :



Task : Given an array and a key value as input, find the index of the key in the array. Return(-1) if it is not present.

Sample Input :-

$$n = 5$$

$$\text{arr} = 10 \quad 20 \quad 30 \quad 40 \quad 50$$

$$x = 40$$

Sample Output :-

$$\text{index} = 3$$

## Arrays in Java – Linear Search Algorithm:

### Program:

```
package com.company;

import java.util.*;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the length of the array (n): ");
        int n = sc.nextInt();
        int[] arr = new int[n];
        System.out.println("Enter the elements of the array (arr): ");
        for (int i = 0; i < arr.length; i++) {
            arr[i] = sc.nextInt();
        }
        System.out.println("Enter the number which is needed to search (keyValue): ");
        int keyValue = sc.nextInt();
        int answer = -1;
        for (int i = 0; i < arr.length; i++) {
            answer = (arr[i] == keyValue) ? i : answer;
        }
        System.out.println(keyValue+" is present in the "+answer+"th index of the array");
    }
}
```

Output:

```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:
```

```
Enter the length of the array (n):
```

```
5
```

```
Enter the elements of the array (arr):
```

```
10
```

```
20
```

```
30
```

```
40
```

```
50
```

```
Enter the number which is needed to search (keyValue):
```

```
40
```

```
40 is present in the 3th index of the array
```

---

```
Process finished with exit code 0
```

## for-each loop :-

Definition :- The for each loop in java, also called as 'enhanced for loop', was introduced in java 5. It is of the alternative approaches that are used for traversing the iterables.

As the ~~name~~ name suggests, it is mainly used to iterate over each of the iterable elements one by one.

## Sample Program :-

```
class Main {  
    public static void main (String [] args) {  
        int [] arr = {5, 4, 3, 2, 1};  
        for (int num : arr) {  
            System.out.println (num);  
        }  
    }  
}
```

## Arrays in Java – For-Each Loop:

### Program:

```
package com.company;

public class Main {
    public static void main(String[] args) {
        String[] names = {"hritick", "aswin", "karthi"};
        for (String name:names){
            System.out.println(name);
        }
    }
}
```

### Output:

```
"C:\Program Files\Java\jdk-21\bin\java.exe"
hritick
aswin
karthi
```

```
Process finished with exit code 0
```

---