



# Arrays in JAVA

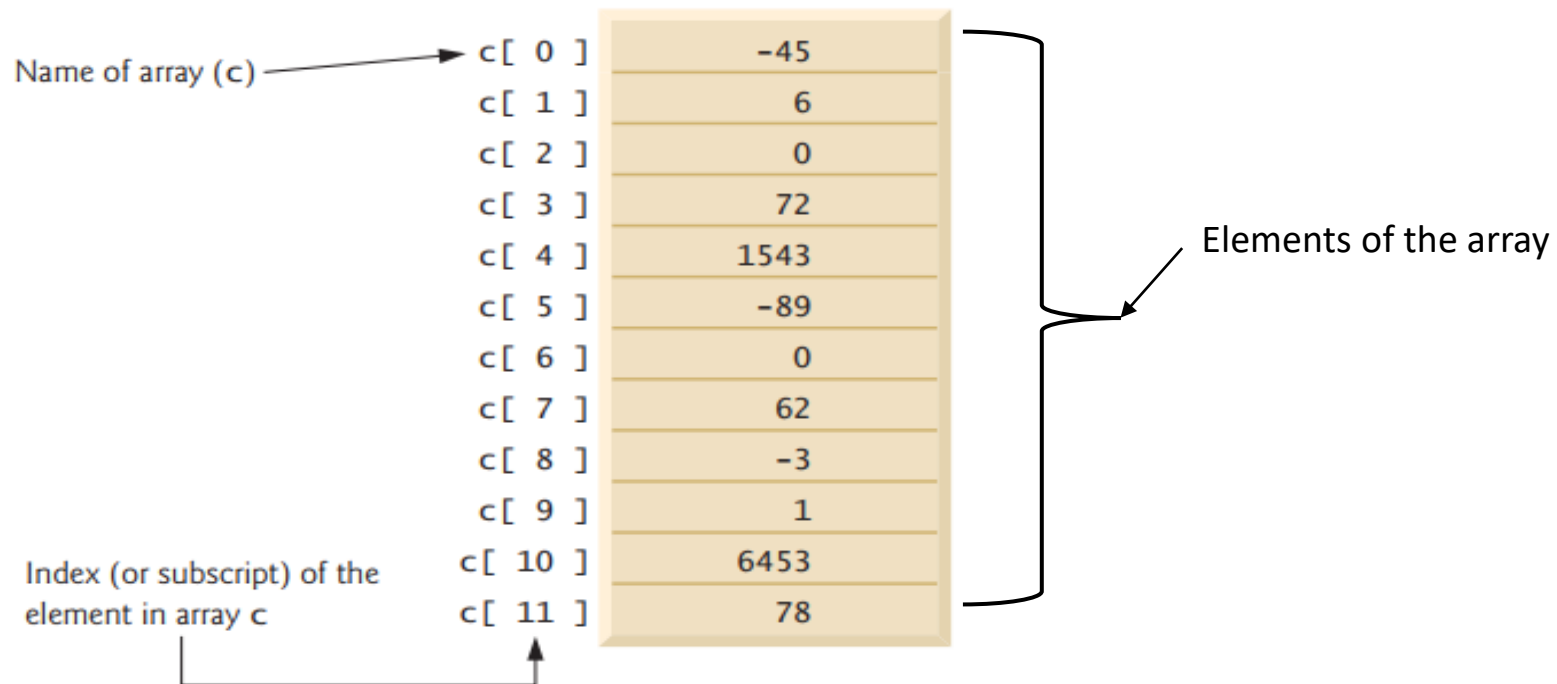
Computer Engineering

[Yusramohammed@tiu.edu.iq](mailto:Yusramohammed@tiu.edu.iq)

2022 - 2023

# What is an Array?

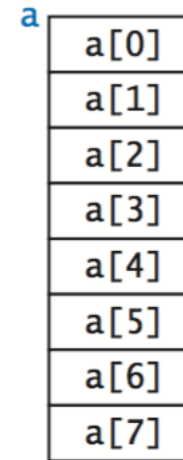
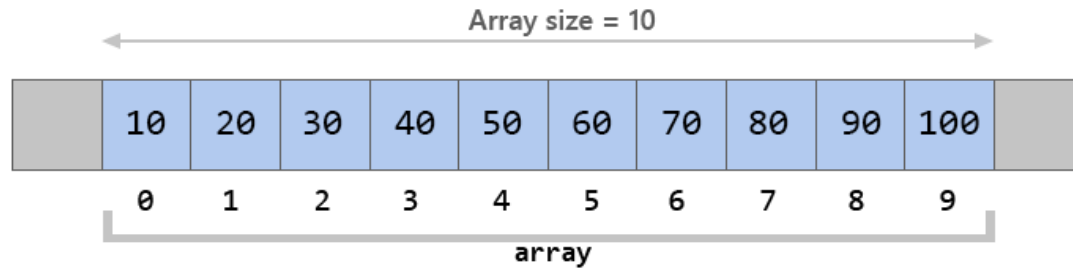
- An Array is a collection of similar type of elements that have a contiguous memory location.
- The array elements have same type.
- The position number of the element is called the element's **index** or **subscript**.
- Array names follow the same conventions as other variable names.



# Array Types?

- There are two types of array:

## 1. One dimensional Array:



## 2. Two Dimensional Array:

	column 1	column 2	column 3	column 4	column 5
row1	arr[0][0]	arr[0][1]	arr[0][2]	arr[0][3]	arr[0][4]
row2	arr[1][0]	arr[1][1]	arr[1][2]	arr[1][3]	arr[1][4]
row3	arr[2][0]	arr[2][1]	arr[2][2]	arr[2][3]	arr[2][4]

# One Dimensional Array: Declaration

- To use an array in a program, you must declare a variable to reference the array, and you must specify the type of array the variable can reference.
- **arrayType [] arrayName;** → Declaration of an array.
- **arrayName= new arrayType[numberOfIndex]** → specify the index of array

```
int[] c; // declare the array variable  
c = new int[ 12 ]; // create the array; assign to array variable
```

```
int []array1=new int[5];  
double []array2=new double[5];  
float []array3=new float[5];  
String []array4=new String[8];  
char [] array5=new char[4];
```

- When we want to output array values we use for repetition statement.

***for(int index=first index of array ; index<=arrayLength ; index++)***

# Array Declaration (Example)

- We create an array `array1` of type integer and has an index of 5 elements. It is not initialized. And we run the program to see the elements of the array.

```
public class Example1 {  
    public static void main(String[] args) {  
        int []array1=new int[5];  
        System.out.println("Array Index\tarray Values: ");  
        for(int index=0;index<array1.length;index++)  
        {  
            System.out.printf("%d\t\t%d\n",index,array1[index]);  
        }  
    }  
}
```

- **Output**

Array Index	array Values:
0	0
1	0
2	0
3	0
4	0

# Initializing Array

- You can create an array and initialize its elements with an **array initializer**—a comma-separated list of expressions (called an **initializer list**) enclosed in braces.
- When we initialize an array, the array length is determined by the number of elements in the initializer list.  
For Example:

```
int[] n = { 10, 20, 30, 40, 50 };  
           ↑   ↑   ↑   ↑   ↑  
         n[0] n[1] n[2] n[3] n[4]
```

- creates a five-element array with index values 0–4.
- Element `n[0]` is initialized to 10, `n[1]` is initialized to 20, and so on.
- If we don't initialize the array values it gives the **default value** to the array.

# Initializing Array (Example)

- We create an array **array1** of type **integer** and has an index of 5 elements and initialized to {23,45,67,89,87}. Write a program to create the array and print the program to see the elements of the array.

```
public class Example1 {  
    public static void main(String[] args) {  
        int [] array1={23,45,67,89,87};  
        System.out.println("Array Index\tarray Values: ");  
        for(int index=0;index<array1.length;index++)  
        {  
            System.out.printf("%d\t\t%d\n",index,array1[index]);  
        }  
    }  
}
```

- **Output**

Array Index	array Values:
0	23
1	45
2	67
3	89
4	87

# Output Specific index in Array

- If we want to output a specific index's value: We write

***arrayName[noOfindex]***

- And we don't need the ***for*** loop.

```
public class Example1 {  
    public static void main(String[] args) {  
        int [] array1={23,45,67,89,87};  
        System.out.println("Array Index\tarray Values: ");  
        System.out.printf("%d\t\t%d\n",2,array1[2]);  
    }  
}
```

- **Output**

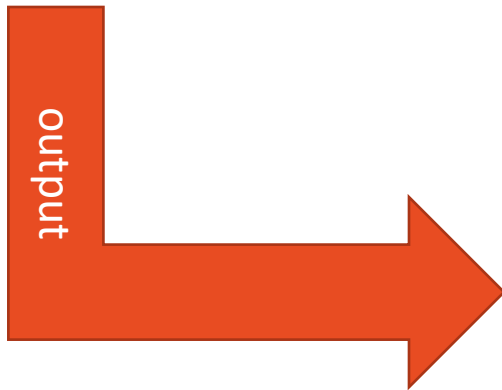
Array Index  
2

array Values:  
67



# Array of Type String (Example)

```
public static void main(String[] args) {  
    String array={"yusra", "pasha", "vinos", "yara"};  
    System.out.println("index\tvalue\n*****\t*****");  
    for(int i=0;i<array.length;i++){  
        System.out.println(i+"\t"+array[i]);  
    }  
}
```



```
run:  
index    value  
*****  *****  
0        yusra  
1        pasha  
2        vinos  
3        yara
```

# Sample Development

- Find the average for the elements in the array below

**{43, 65, 76, 98, 35}**

```
public class Average_Array {
    public static void main(String[] args) {
        int array[]={43, 65, 76, 98, 35};
        int sum=0;
        for(int i=0;i<array.length;i++){
            sum+= array[i];
        }
        System.out.println("Sum of array values: "+sum);
        double avg=sum/array.length;
        System.out.println("Average= "+avg);
    }
}
```

- Try it and see the output

# Another way to initializing array

- The user can give value to the array indexes. For this we have two for statement, one for inputting the values to the array and one for outputting the output.
- In the declaration of the array we should specify the index numbers.
- Example:
- ***int [] arrayName = new int [number of indexes].***

- ***Example***

```
public class Example2 {  
    public static void main(String[] args) {  
        Scanner input=new Scanner(System.in);  
  
        System.out.println("Enter the values to the array:");  
        //int x=input.nextInt();  
        int [] array=new int [5];  
        for (int index=0;index<array.length;index++)  
        {  
            array[index]=input.nextInt();  
        }  
        System.out.println("index\tvalues:");  
        for(int index=0;index<array.length;index++)  
        {  
            System.out.printf("%d\t%d\n", index, array[index]);  
        }  
    }  
}
```

## *Another way to initializing array (Example)*

```
Enter the values to the array:
```

```
54
```

```
32
```

```
54
```

```
43
```

```
54
```

```
index    values:
```

```
0        54
```

```
1        32
```

```
2        54
```

```
3        43
```

```
4        54
```

# Enhanced for Statement

- The **enhanced for statement** iterates through the elements of an array *without* using a counter
- It works just like for but the structure is different. In the enhanced for :

```
for ( parameter : arrayName )  
    statement
```

- where **parameter** has a type and an identifier (e.g., int number), and **arrayName** is the array through which to iterate.

```
public class EnhancedforTest {  
    public static void main(String[] args) {  
        int [] array={43,54,65,76,87,98,21};  
        int total=0;  
        for(int index:array)  
        {  
            total+=index;  
        }  
        System.out.println("total of array elements:"+total);  
    }  
}
```

- **Output**

total of array elements:444

# Sample Development



- Twenty students were asked to rate on a scale of 1 to 5 the quality of the food in the student cafeteria, with 1 being “awful” and 5 being “excellent.” Place the 20 responses in an integer array and determine the frequency of each rating.