

# **JAVA @17**

## Arrays

# Objective

After completing this session you will be able to understand,

- what is an Array?
- List out the types of Array.
- Declare and create an array.
- Iterate through an array.

# Metaphor for Arrays



A box of Apples. Box is the container for apples.

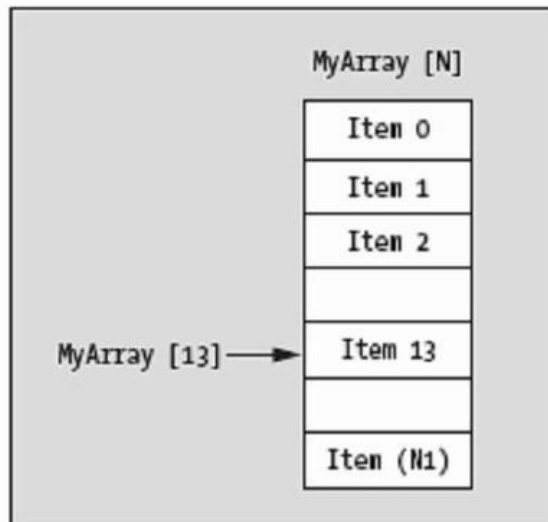
Similar to box Arrays are container for data of similar data type (like the apples).

# What is an Array?

## What is an Array?

An array is variable that can hold a group of values of *same type* and referred by a common name.

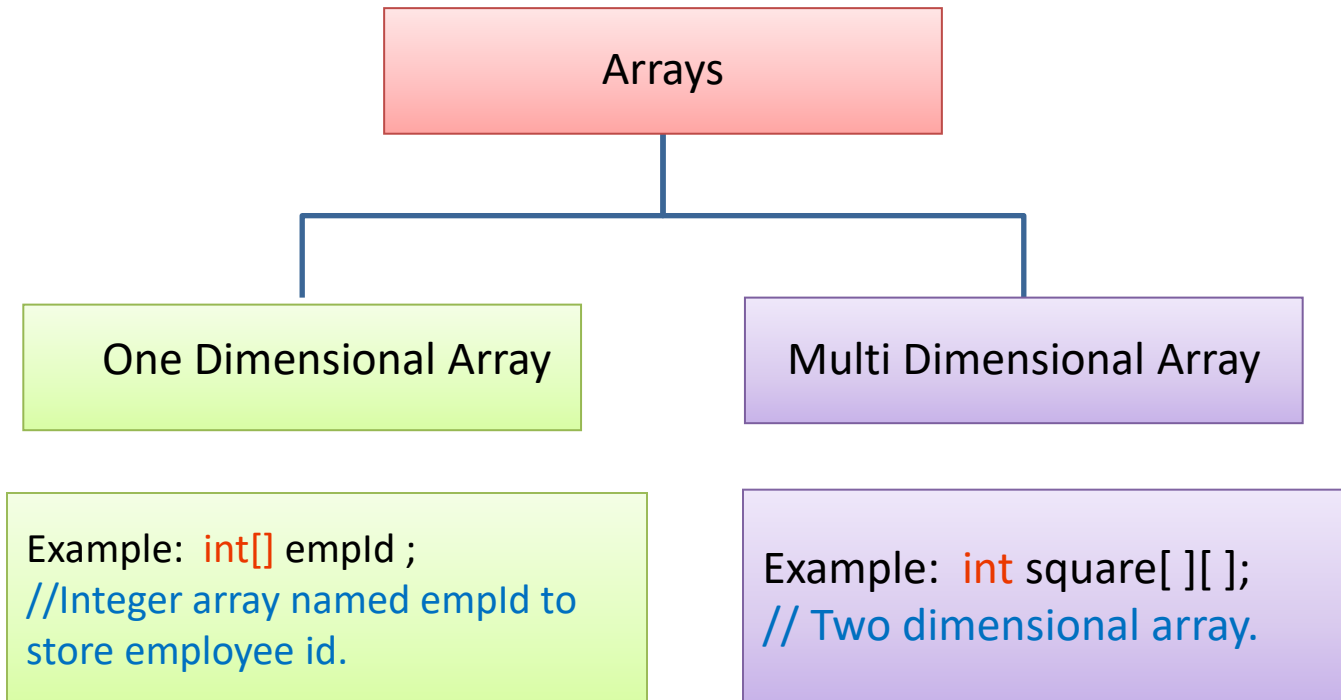
Arrays can hold primitives or objects of same types.



Individual values being stored inside an array named *MyArray*.

Each item can be accessed using the variable name *MyArray*.

# Types of Array



# Declaring and Creating an One Dimensional Array

## 1. Declaring an Array:

An array variable declared should has a data type and a valid identifier.

**Syntax** : `<type> [ ] <array-name>;`

**Example :**

`int [ ] empId;`



Declares an array of type int

`Employee [ ] emp;`



Declares an array of Employee Object

## 2. Initializing an Array

As arrays are also a type of object, they are created with the new keyword.

**Syntax:** `<array-name>= new <type>(<size>;`

**Example:**

`empId=new int[3];`



Creates an array that can hold 3 int values

`emp=new Employee[3];`

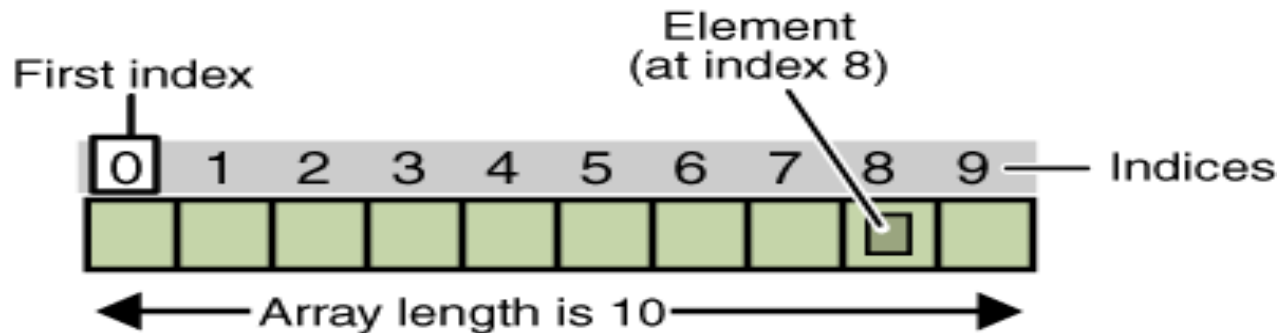


Creates an array that can hold 3 Employee objects.

# One Dimensional Arrays

## Some Facts about arrays:

- The **length** of an array is established when the array is created.
- The **length** of an array is **fixed** at the time of its creation.
- Each item in an array is called an **element**, and each element is accessed by its numerical **index**.



The above diagram depicts an array of ten elements

# Adding Elements to an One Dimensional Array

How to add values to a one dimensional array?

Based on the data type of the array the respective values can be stored inside the array.

Syntax:

```
<array-name>[<index-number>] = <value>;
```

Example:

```
empld[0]=76;  
empld[1]=13;  
empld[2]=56;
```

This adds the employee id's in the 'empld' array`

```
emp[0]=new Employee("Arun",12000);  
emp[1]=new Employee("Ram",11100);  
emp[2]=new Employee("Raj",8000);  
emp[3]=new Employee("Sam",10000);
```

This adds employee object

This sets the empName & empld objects in Employee Class



# Alternate way of adding elements to an One Dimensional Array

Let us look at an alternate way to create, initialize and assign values in one step,

```
int[] empld = { 76, 13, 56, 87};
```

This declares an array of type int with values 76, 13, 56 and 87

```
Employee[] emp={new Employee("Ram",22),new Employee("Arun",44)};
```

This declares an Employee object array, stores two employee objects

Here, the length of the array is determined by the number of values stored in the array.

# Accessing the Elements of an One Dimensional Array

## How to retrieve values from an array?

Array elements are accessed using the element's index.

The index position of the first element in the array is **0**, last element is position is **array length -1**.

## Syntax:

```
<type> <array name> = <array-name>[index-number];
```

## Example:

```
int id=empld[0];   Employee emp1=emp[3];
```

Retrieves the first item  
in empld array

Retrieves the fourth  
item in emp array

How to find the length of an array?

Use **empld.length**; to get the total number of items in the array.

# Lend a Hand–One Dimensional Array

Program to print even numbers between 0 & 100.

1. Create a program “ArrayDemo” add two methods,
  1. **storeNumbers** – Creates and stores a array with values from 0~100
  2. **printEvenNumber** – traverse through the array stored and print all the even numbers.

From the main method invoke both the mentioned above and print the even numbers.

# Lend a Hand-Solution

```
public class ArrayDemo {
```

```
    int[] numbers;
```

```
    public void storeNumbers() {  
        numbers = new int[101];  
        for (int i = 0; i <= 100; i++) {  
            numbers[i] = i;  
        }  
    }
```

This method stores 100 numbers in an array.

```
    public void printEvenNumber() {  
        System.out.println("The even numbers between 0 and 100 are ");  
        for (int i = 0; i < numbers.length; i++) {  
            if ((numbers[i] % 2) == 0) {  
                System.out.println(numbers[i]);  
            }  
        }  
    }
```

This method prints the even numbers between 0 and 100.

```
    public static void main(String args[]) {  
        ArrayDemo arrayDemo = new ArrayDemo();  
        arrayDemo.storeNumbers();  
        arrayDemo.printEvenNumber();  
    }
```

Invokes the methods.

```
}
```

# Multi Dimensional Arrays

What are multi dimensional arrays?

Multi Dimensional Arrays are *array of arrays*.

The two dimensional array can be termed as a physical table with rows and columns, each row labeled with an index of 0 to its maximum bound.

Syntax:

```
type array-name = new type[rows][cols];
```

Example:

```
int marks[ ][ ] = new int[2][3 ]; // 3 rows and 4 columns
```

A two dimensional array is similar to a matrix representation as depicted here

	0	1	2	3
0	41	38	28	31
1	32	34	36	43
2	23	31	12	18

# Advantages and Disadvantages of Arrays

## **Advantages of Java Array:**

- Arrays can store large number of elements by just specifying the index number and the array name.
- Arrays permit efficient random access
- Iteration in arrays is faster than iterating through its counterparts (such as a linked list of the same size)

## **Disadvantages of Java Array:**

- An array has fixed size.
- An array holds only one type of data.
- Insertion and deletion of elements is not efficient

Thank you

*You have successfully completed*  
**Arrays**