

Array in Java

Day 12 Assignment

1. What do you mean by an Array?
2. How to create an Array?
3. Can we change the size of an array at run time?
4. Can you declare an array without assigning the size of an array?
5. What is the default value of Array?
6. What is an ID array with an example?
7. Write a program on a 2D array?

1. What do you mean by an Array?

Ans: Array in java is a group of like-typed variables referred to by a common name. Arrays in Java work differently than they do in C/C++. Following are some important points about Java arrays.

- In Java, all arrays are dynamically allocated. (discussed below)
- Arrays are stored in contiguous memory [consecutive memory locations].
- Since arrays are objects in Java, we can find their length using the object property length. This is different from C/C++, where we find length using sizeof.
- A Java array variable can also be declared like other variables with [] after the data type.
- The variables in the array are ordered, and each has an index beginning with 0.
- Java array can also be used as a static field, a local variable, or a method parameter.
- The size of an array must be specified by int or short value and not long.
- The direct superclass of an array type is Object.
- Every array type implements the interfaces Cloneable and java.io.Serializable.
- This storage of arrays helps us randomly access the elements of an array [Support Random Access].
- The size of the array cannot be altered(once initialized). However, an array reference can be made to point to another array.

An array can contain primitives (int, char, etc.) and object (or non-primitive) references of a class depending on the definition of the array. In the case of primitive data types, the actual values are stored in contiguous memory locations. In the case of class objects, the actual objects are stored in a heap segment.

2. How to create an Array?

Ans: There are two ways you can declare and initialize an array in Java.

- `dataType [] nameOfArray = new dataType [size]`
 - The first is with the new keyword, where you have to initialize the values one by one.
- `dataType [] nameOfArray = {value1, value2, value3, value4}`
 - The second is by putting the values in curly braces.

3. Can we change the size of an array at run time?

Ans: No

4. Can you declare an array without assigning the size of an array?

Ans: Yes. We can declare an array without size but before using it needs to be initialized.

5. What is the default value of Array?

Ans:

in the case of an **int** array, it will be **0**.

in the case of a **boolean** array, it will be **false**.

in the case of a **String** array the default value is **null**.

In the case of a **char** array, the default value is **Unicode** (\u0000).

6. What is an 1D array with an example?

Ans: A one-dimensional array in Java is a collection of similar types of elements stored at contiguous memory locations.

Syntax of 1D array

data-type var-name[];

Or, data-type[] var-name;

Or, data-type []var-name;

7. Write a program on a 2D array?

Ans:

A two-dimensional array is an array of arrays i.e., it's a collection of arrays in which elements are arranged in rows and columns (tabular format). We can access the elements using both the row index and column index

// Declaring 2D array

```
DataType[][] ArrayName;
```

// Creating a 2D array

```
ArrayName = new DataType[r][c];
```

// Program of 2D array

```
import java.util.Arrays;
```

```
public class Main {
```

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

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

```
        // Marks Obtained By Student 1
```

```
        StudentMarks[0][0] = 90; // English
```

```
        StudentMarks[0][1] = 70; // Maths
```

```
        StudentMarks[0][2] = 84; // Science
```

```
        // Marks Obtained By Student 2
```

```
        StudentMarks[1][0] = 75; // English
```

```
        StudentMarks[1][1] = 77; // Maths
```

```
        StudentMarks[1][2] = 89; // Science
```

```
        // Displaying Marks of Students
```

```
        System.out.println("Student Marks Matrix");
```

```
        System.out.println(Arrays.deepToString(StudentMarks));
```

```
    }
```

```
}
```