

A collection of related data pieces stored in contiguous memory locations is referred to as an array. It is the most basic data structure in which each data element may be retrieved simply by its index number alone. For example, if we want to keep a student's grades in five subjects, we don't need to establish distinct variables for each topic. We can instead create an array that stores the data items in contiguous memory locations.

This blog will serve as a resource for practically everything you need to know about arrays, including the differences between single and multidimensional arrays.

What is a Single and Multidimensional Array?

➤Single Array:

A container object that stores a fixed number of values of a single type is called an array.

➤Multidimensional Array:

In simple terms, we can describe multidimensional arrays as an array of arrays.

Single Array:

When it comes to Java arrays, an array is a collection of like-typed variables with a similar name. Arrays in Java are not the same as those in C/C++. The following are some key points to remember regarding Java arrays.

In Java, all arrays are dynamically allocated.

- o Because arrays are objects in Java, we may use the object attribute length to determine their length. This differs from C/C++, where we use sizeof to find length.
- o A Java array variable can also be declared like other variables with [] after the data type.
- o The variables in the array are ordered, and each has an index beginning from 0.
- o Java array can also be used as a static field, a local variable, or a method parameter.
- o The size of an array must be specified by int or short value and not long.

- o The direct superclass of an array type is Object.
- o Every array type implements the interfaces Cloneable and java.io.Serializable.

Depending on the array's definition, it can include both primitive (int, char, etc.) and object (or non-primitive) pointers to a class. The actual values for primitive data types are stored in contiguous memory regions. The real objects are kept in a heap segment for class objects.

UMA----

➤ **The general form of a one-dimensional array declaration is :**

type var-name[];

OR

type[] var-name;

➤ **Instantiating an Array in Java**

Only a reference to an array is produced when an array is declared. You make an array like this to create or provide memory to the array: When it comes to one-dimensional arrays, the general form of new is as follows:

var-name = new type [size];

int[] intArray = new int[20];

When an array is declared, all that is produced is a reference to it. To generate or contribute memory to the array, you create an array like this: When it comes to one-dimensional arrays, new takes the following form:

➤ **Accessing Java Array Elements using for Loop**

The index of each element in the array is used to access it. The index starts at 0 and goes all the way to (total array size)-1. Java for Loop allows you to access all of the elements in an array.

for (int i = 0; i < arr.length; i++)

System.out.println("Element at index " + i + " : "+ arr[i]);

BHAVIN-

Multidimensional Array:

The simplest version of a multidimensional array is a two-dimensional array. For ease of understanding, we can think of a two-dimensional array as an array of one-dimensional arrays.

➤ **Data in multidimensional arrays are stored in tabular form.**

Fig. Tabular Presentation Of 2-D Array

➤ **Declaration**

● **General form of declaring N-dimensional arrays:**

data_type array_name [size1][size2]....[sizeN];

- **data_type:** The data type that will be placed in the array.
- **array_name:** Name of the array.
- **size1, size2,... ,sizeN:** Dimensions and their sizes.

➤ **Examples:**

○ **Two dimensional array:**

int two_d[10][20];

○ **Three dimensional array:**

int three_d[10][20][30];

We can declare a two dimensional integer array say 'x' of size 10,20 as: `int x[10][20];`

Elements in two-dimensional arrays are commonly referred by `x[i][j]` where `i` is the row number and '`j`' is the column number.

A two – dimensional array can be seen as a table with '`x`' rows and '`y`' column

- where the row number ranges from 0 to (`x-1`)

- and column number ranges from 0 to (y-1).

HARSHAL-

➤ **Initializing Two – Dimensional Arrays:**

There are two ways in which a Two-Dimensional array can be initialized.

First Method:

- $\text{int } x[3][4] = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}.$
- The above array has 3 rows and 4 columns.
- The components in the braces are kept in the table in the same order as they appear in the braces.
 - The array will be filled in order, with the first four elements from the left in the first row, the following four elements in the second row, and so on.

Better Method:

- $\text{int } x[3][4] = \{\{0,1,2,3\}, \{4,5,6,7\}, \{8,9,10,11\}\};$
- This type of initialization make use of nested braces.
- Each set of inner braces represents one row.
- In the above example there are total three rows so there are three sets of inner braces.

➤ **Accessing Elements of Two-Dimensional Arrays:**

- Row and column indices are used to access elements in two-dimensional arrays.
- Example: $\text{int } x[2][1];$
- The element in the third row and second column is represented in the example above.
- Note: In arrays if size of array is N. Its index will be from 0 to N-1. Therefore, for row index 2 row number is $2+1 = 3$.
- We can use nested for loops to print all the items of a Two-Dimensional array. For loops, we'll need two. One traverses the rows, while the other traverses the columns.

➤ SHASHANK-

Three – dimensional Array (3D-Array)

A three-dimensional array is a more sophisticated version of a multidimensional array. For ease of understanding, a three-dimensional array can be thought of as an array of two-dimensional arrays.

Indirect Method of Declaration:

Declaration – Syntax:

```
data_type[][][] array_name = new data_type[x][y][z];
```

For example: int[][][] arr = new int[10][20][30];

Initialization – Syntax:

```
array_name[array_index][row_index][column_index] = value;
```

For example: arr[0][0][0] = 1;

Accessing Elements of Three-Dimensional Arrays:

Elements in three-dimensional arrays are commonly referred by `x[i][j][k]` where 'i' is the array number, 'j' is the row number and 'k' is the column number.

Representation of 3D array in Tabular Format: A three – dimensional array can be seen as a tables of arrays with 'x' rows and 'y' columns where the row number ranges from 0 to (x-1) and column number ranges from 0 to (y-1). A three – dimensional array with 3 array containing 3 rows and 3 columns is shown below:

Fig. Representation of 3-D array.

CHAITANYA-

Comparison Between Types Of Arrays in Java

An array is a data structure that allows you to store variables of similar data kinds in close proximity. The array's key benefit is random access and cache friendliness. The array can be divided into three types:

- One Dimensional (1D) Array

- Two Dimension (2D) Array

- Multidimensional Array

- ➤ **One Dimensional Array:**

- o It is a list of the variable of similar data types.

- o It allows random access and all the elements can be accessed with the help of their index.

- o The size of the array is fixed.

- o

- ➤ **Two Dimensional Array:**

- o It is a list of lists of the variable of the same data type.

- o It also allows random access and all the elements can be accessed with the help of their index.

- o It can also be seen as a collection of 1D arrays. It is also known as the Matrix.

- o Its dimension can be increased from 2 to 3 and 4 so on.

- o They all are referred to as a multi-dimension array.

- o The most common multidimensional array is a 2D array.

Differences On The Basis Of Different Criteria:

ALL ONE CRITERIA EACH-

Criteria	One Dimensional Array	Multidimensional Array
(MANSI)Definition	Store a single list of the element of a similar data type.	Store a single list of the element of a similar data type.
Representation (UMA)	Represent multiple data items as a list.	Represent multiple data items as a table consisting of rows and columns.
(BHAVIN)Dimension	One	Two
Size(bytes) (SHASHANK)	size of(datatype of the variable of the array) * size of the array	size of(datatype of the variable of the array)* the number of rows* the number of columns.
Address calculation. (HARSHAL)	Address of a[index] is equal to (base Address+ Size of each element of array * index).	Address of a[i][j] can be calculated in two ways row-major and column-major Column Major: Base Address + Size of each element (number of rows(j-lower bound of the column)+(i-lower bound of the rows)) Row Major: Base Address + Size of each element (number of columns(i-lower bound of the row)+(j-lower bound of the column))
Example (CHAITANYA)	int arr[5]; //an array with one row and five columns will be created.	int arr[2][5]; //an array with two rows and five columns will be created.

	{a , b , c , d , e }*	a b c d e
		f g h i j

(FIRST 3 POINTS HARSHAL NEXT 3 UMA LAST 2 BHAVIN)

Real Time Application Of Arrays and Multidimensional Arrays:

- Arrangement of the leader-board of a game can be done simply through arrays to store the score and arrange them in descending order to clearly make out the rank of each player in the game. (HARSHAL)
- A simple question Paper is an array of numbered questions with each of them assigned some marks. (HARSHAL)
- 2D arrays, commonly known as, matrices, are used in image processing. (HARSHAL)
- It is also used in speech processing, in which each speech signal is an array.(UMA)
- Your viewing screen is also a multidimensional array of pixels. (UMA)
- Book titles in a Library Management Systems. (UMA)
- Online ticket booking.(BHAVIN)
- Contacts on a cell phone.(BHAVIN)