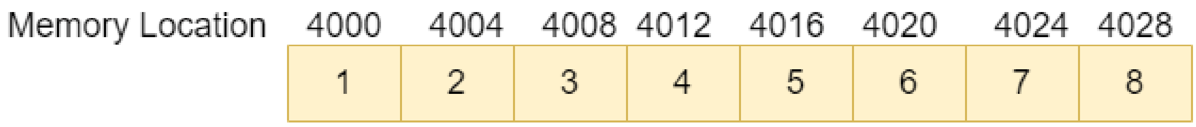
What is Array

The array is a data structure where values or items are placed in a linear order, which means the memory assigned to each item is contiguous. The data type of an array is the same for all the elements present in it.

With the contiguous memory allocation, it becomes easier to find out the memory location of any element in the array by just knowing the first memory location and adding the offset.

For Example:



Memory Location	4000	4004	4008	4012	4016	4020	4024	4028
	1	2	3	4	5	6	7	8

In the above example, we have an integer array, and the memory locations are contiguous, where each integer takes 4 bytes, and the starting address is 4000.

We cannot increase or decrease the size of an array dynamically once it is declared. The memory allocation is static in most of the languages, so we have to declare the array size at the time of array declaration.

Indexing in the array

We can identify and access each value with the help of an index number in an array.

For example, arr[1] will be used to access the second element of the array in zero-based indexing.

The array indexing can be of the following types:

0-based indexing

If the first value of the array gets the index 0, it is called 0-based indexing. Most of the programming languages follow 0-based indexing.

1-based indexing

If the array index starts from 1, it is called 1-based indexing. It means the first value of the array will be assigned index 1.

N-based indexing

When the index of the first element can be anything, even negative values, it is called N-based indexing.

Initialization Methods in an Array:

There are various methods by which we can declare and initialize the array in Java.

1. Using the 'new' keyword

With the help of a new keyword, we can allocate the specific size of memory to the array. In java, when we use the new keyword, it allocates the memory in a heap, so it allocates the memory as static.

For Example: