**Interger Arrays in C/C++ -**

It is a group of variables of similar data types referred to by a single element.

Its elements are stored in a contiguous memory location.

The size of the array should be mentioned while declaring it.

Array elements are always counted from zero (0) onward.

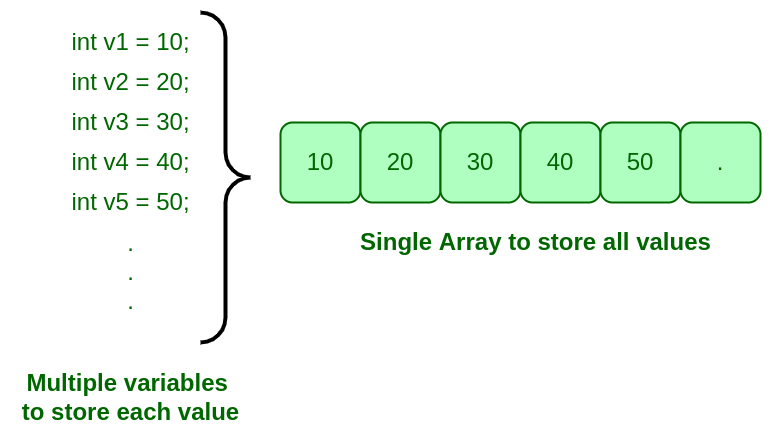
Array elements can be accessed using the position of the element in the array.

The array can have one or more dimensions.

An array in C/C++ or be it in any programming language is a collection of similar data items stored at contiguous memory locations and elements can be accessed randomly using indices of an array.  They can be used to store the collection of primitive data types such as int, float, double, char, etc of any particular type. To add to it, an array in C/C++ can store derived data types such as structures, pointers etc. Given below is the picture representation of an array.



**Why do we need arrays?**   
We can use normal variables (v1, v2, v3, ..) when we have a small number of objects, but if we want to store a large number of instances, it becomes difficult to manage them with normal variables. The idea of an array is to represent many instances in one variable.



# Advantages:-

# Code Optimization:  we can retrieve or sort the data efficiently.

# Random access: We can get any data located at an index position.

# Disadvantages:-

# Size Limit: We can store only the fixed size of elements in the array. It doesn’t grow its size at runtime.

# 

# In the above image int a[3]={[0…1]=3}; this kind of declaration has been obsolete since GCC 2.5

Array declaration by specifying the size

#include <iostream>

using namespace std;

int main()

{

// array declaration by specifying size

int arr1[10];

// With recent C/C++ versions, we can also

// declare an array of user specified size

int n = 10;

int arr2[n];

return 0;

}

Array declaration by initializing elements

#include <iostream>

using namespace std;

int main()

{

int arr[] = { 10, 20, 30, 40};

return 0;

// Compiler creates an array of size 4.

// above is same as "int arr[4] = {10, 20, 30, 40}"

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#include <iostream>

using namespace std;

int main()

{

// Array declaration by specifying size and initializing

// elements

int arr[6] = { 10, 20, 30, 40 };

// Compiler creates an array of size 6, initializes first

// 4 elements as specified by user and rest two elements as

// 0. above is same as "int arr[] = {10, 20, 30, 40, 0, 0}"

return 0;

}

Advantages of an Array in C/C++:

Random access of elements using the array index.

Use of fewer lines of code as it creates a single array of multiple elements.

Easy access to all the elements.

Traversal through the array becomes easy using a single loop.

Sorting becomes easy as it can be accomplished by writing fewer lines of code.

Disadvantages of an Array in C/C++:

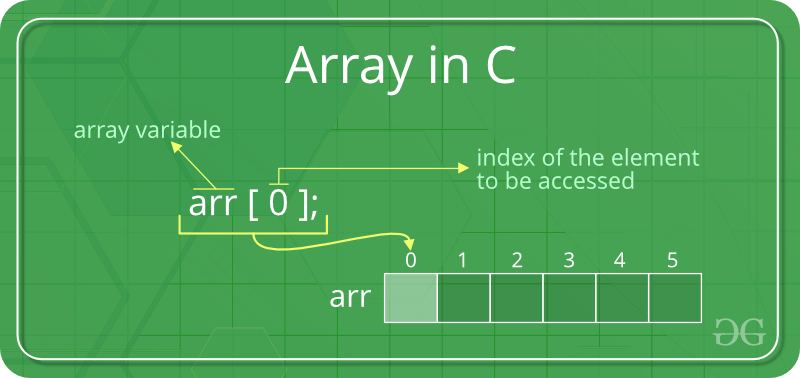
Allows a fixed number of elements to be entered which is decided at the time of declaration. Unlike a linked list, an array in C is not dynamic. Insertion and deletion of elements can be costly since the elements are needed to be managed in accordance with the new memory allocation.

Facts about Array in C/C++:

Accessing Array Elements:   
Array elements are accessed by using an integer index. Array index starts with 0 and goes till the size of the array minus 1.

The name of the array is also a pointer to the first element of the array*.*

**Array declaration by specifying the size**



Example:-

#include <stdio.h>

int main()

{

int arr[5];

arr[0] = 5;

arr[2] = -10;

arr[3 / 2] = 2; // this is same as arr[1] = 2

arr[3] = arr[0];

printf("%d %d %d %d", arr[0],

arr[1], arr[2], arr[3]);

return 0;}

Output-

5 2 -10 5 5