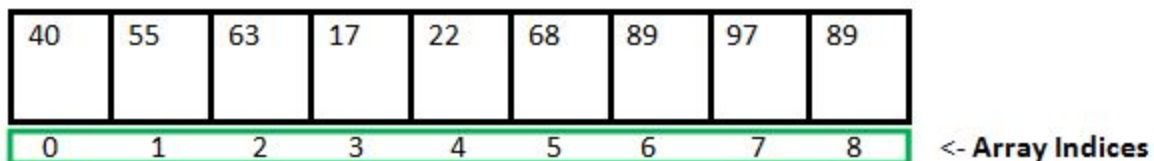


Introduction

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



Array Length = 9

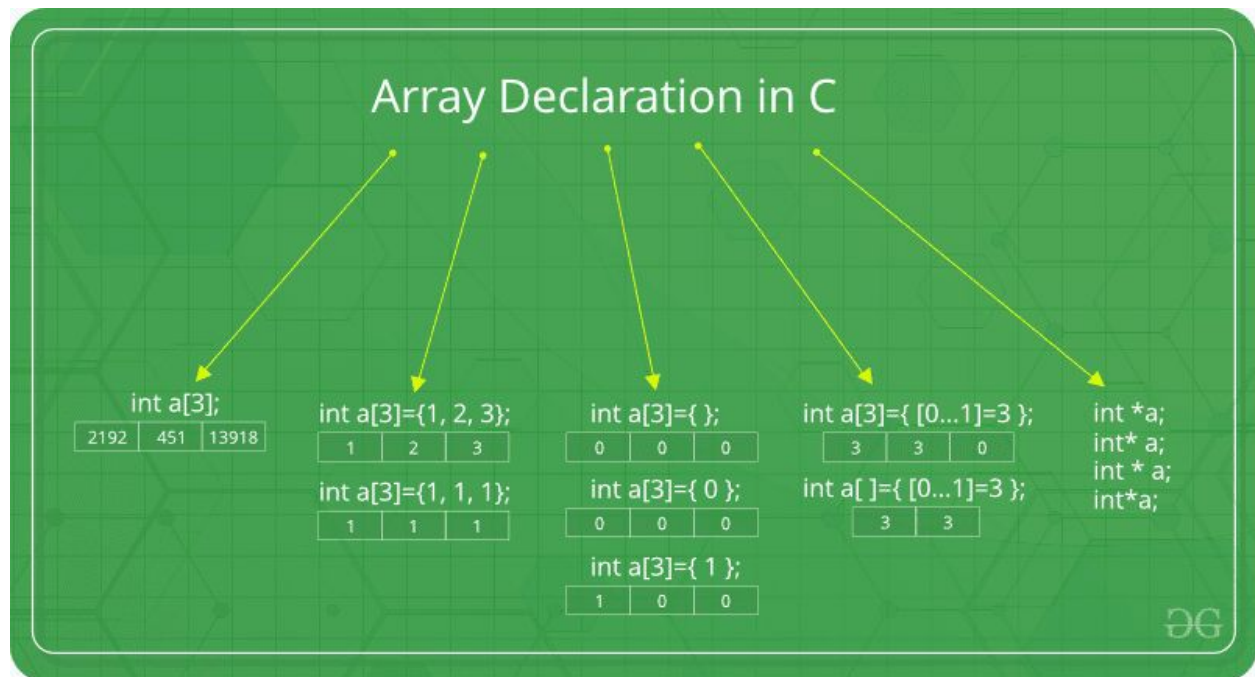
First Index = 0

Last Index = 8

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.

Array declaration in C/C++:



There are various ways in which we can declare an array. It can be done by specifying its type and size, by initializing it or both.

1 Array declaration by specifying size

```
int arr1[10];
```

2 . Array declaration by initializing elements

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

3 . Array declaration by specifying size and initializing elements

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

Advantages of an Array in C/C++:

1. Random access of elements using array index.
2. Use of less line of code as it creates a single array of multiple elements.
3. Easy access to all the elements.
4. Traversal through the array becomes easy using a single loop.
5. Sorting becomes easy as it can be accomplished by writing less line of code.

Disadvantages of an Array in C/C++:

1. 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.
2. Insertion and deletion of elements can be costly since the elements are needed to be managed in accordance with the new memory allocation.

Example 1:

```
#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

Example 2:

```
#include <stdio.h>
int main () {

    int n[ 10 ];
    int i;

    for ( i = 0; i < 10; i++ ) {
        n[ i ] = i + 100; /* set element at location i to i + 100 */
    }

    for (i= 0; i < 10; i++ ) {
        printf("Element[%d] = %d\n", i, n[i] );
    }

    return 0;
}
```

Output:

Element[0] = 100
Element[1] = 101
Element[2] = 102
Element[3] = 103
Element[4] = 104
Element[5] = 105
Element[6] = 106
Element[7] = 107
Element[8] = 108

Element[9] = 109