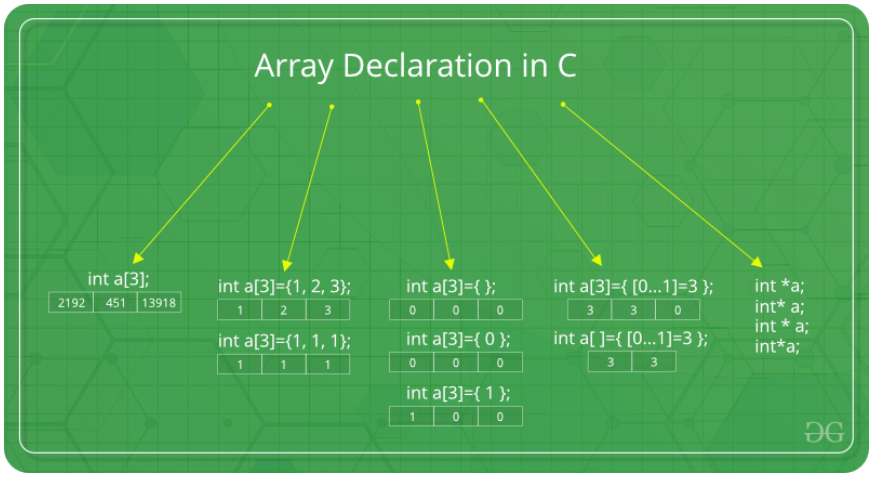
Array declaration in C++

Array declaration in C++ is same as C.



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

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

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

```
// 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}"
}
```

Array declaration by specifying the size and initializing elements

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

    return 0;
}
```

Advantages of an Array in C/C++:

1. Random access of elements using the array index.
2. Use of fewer lines 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 fewer lines 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.