Array Basics

An array is a collection of items that are of the data type stored in contiguous memory locations. It is also known as a subscript variable.

It can even store the collection of derived data types such as pointers, structures, etc.

An array can be of any dimension. The C++ Language places no limits on the number of dimensions in an array. This means we can create arrays of any number of dimensions. It could be a 2D array or a 3D array or more.

Advantages of Arrays?

- It is used to represent multiple data items of the same type by using only a single name.
- Accessing any random item at any random position in a given array is very fast in an array.
- There is no case of memory shortage or overflow in the case of arrays since the size is fixed and elements are stored in contiguous memory locations.

Array Operations

Defining an array

1. **Without specifying the size of the array:**

```
int arr[] = {1, 2, 3};
```

Here, we can leave the square brackets empty, although the array cannot be left empty in this case. It must have elements in it.

2. **With specifying the size of the array:**

```
int arr[3];arr[0] = 1, arr[1] = 2, arr[2] = 3;
```

Accessing an array element

An element in an array can easily be accessed through its index number. An index number is a special type of number which allows us to access variables of arrays. Index number provides a method to access each element of an array in a program. This must be remembered that the index number starts from 0 and not one.

Example:

```
#include <iostream>using namespace std; int main(){    int arr[] = {1, 2, 3};    cout << arr[1] << endl;}
```

Output:

```
2
```

Changing an array element

An element in an array can be overwritten using its index number.

Example:

```
#include <iostream>
using namespace std;

int main()
{
   int arr[] = {1, 2, 3};
   arr[2] = 8; // changing the element on index 2
   cout << arr[2] << endl;
}
```

Output:

```
8
```