**C++ Arrays**

* Arrays are used to store multiple values in a single variable, instead of declaring separate variables for each value.
* An array is a collection of similar data items stored at contiguous memory locations.
* Elements can be accessed randomly using indices of an array.
* They can be used to store collection of primitive data types such as int, float, double, char, etc of any particular type.
* To declare an array, define the variable type, specify the name of the array followed by **square brackets** and specify the number of elements it should store:

**Syntax of array declaration**

Datatype name[size]

Example:

Int a[5];

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

a[0] a[1] a[2] a[3] a[4]

**length of array = 5**

**first index = 0**

**last index = 4**

**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++:**

**Array declaration by specifying size**

**// 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];**

**Array declaration by initializing elements**

**// Array declaration by initializing elements**

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

**// Compiler creates an array of size 4.**

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

**Array declaration by specifying size and initializing elements**

**// 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}"**

**Advantages of an Array in C/C++:**

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

**Facts about Array in C/C++:**

**Accessing Array Elements:**

1. Array elements are accessed by using an integer index. Array index starts with 0 and goes till size of array minus 1.
2. Name of the array is also a pointer to the first element of array.

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**int arr[5];**

**arr[0] = 5;**

**arr[2] = -10;**

**// this is same as arr[1] = 2**

**arr[3 / 2] = 2;**

**arr[3] = arr[0];**

**cout << arr[0] << " " << arr[1] << " " << arr[2] << " "**

**<< arr[3];**

**return 0;**

**}**

**Output**

5 2 -10 5

**// This C++ program compiles fine**

**// as index out of bound**

**// is not checked in C.**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**int arr[2];**

**cout << arr[3] << " ";**

**cout << arr[-2] << " ";**

**return 0;**

**}**

**Output**

-449684907 4195777

 the following program compiles fine but may produce unexpected output when run.

**The elements are stored at contiguous memory locations**

**Example:**

**// C++ program to demonstrate that array elements**

**// are stored contiguous locations**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**// an array of 10 integers.**

**// If arr[0] is stored at**

**// address x, then arr[1] is**

**// stored at x + sizeof(int)**

**// arr[2] is stored at x +**

**// sizeof(int) + sizeof(int)**

**// and so on.**

**int arr[5], i;**

**cout << "Size of integer in this compiler is "**

**<< sizeof(int) << "\n";**

**for (i = 0; i < 5; i++)**

**// The use of '&' before a variable name, yields**

**// address of variable.**

**cout << "Address arr[" << i << "] is " << &arr[i]**

**<< "\n";**

**return 0;**

**}**

Another way to traverse the array

**#include<bits/stdc++.h>**

**using namespace std;**

**int main()**

**{**

**int arr[6]={11,12,13,14,15,16};**

**// Way -1**

**for(int i=0;i<6;i++)**

**cout<<arr[i]<<" ";**

**cout<<endl;**

**// Way 2**

**cout<<"By Other Method:"<<endl;**

**for(int i=0;i<6;i++)**

**cout<<i[arr]<<" ";**

**cout<<endl;**

**return 0;**

**}**

**Output**

11 12 13 14 15 16

By Other Method:

11 12 13 14 15 16