
C++ Programming Language
7th part Array in C++
By: Muqaddar Niazi

Array

An array is defined as the collection of similar type of data items stored at contiguous memory locations. Arrays are the derived data type in C programming language which can store the primitive type of data such as int, char, double, float, etc. It also has the capability to store the collection of derived data types, such as pointers, structure, etc. The array is the simplest data structure where each data element can be randomly accessed by using its index number.

C array is beneficial if you have to store similar elements. For example, if we want to store the marks of a student in 6 subjects, then we don't need to define different variables for the marks in the different subject. Instead of that, we can define an array which can store the marks in each subject at the contiguous memory locations.

Continue…

- By using the array, we can access the elements easily. Only a few lines of code are required to access the elements of the array.

Advantages of Array

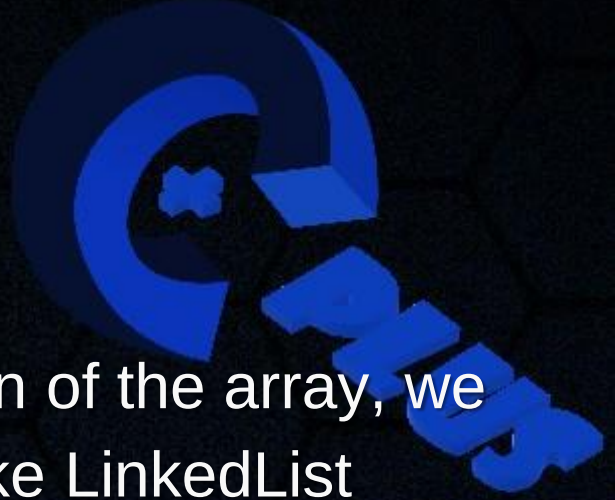
1) Code Optimization: Less code to the access the data.

2) Ease of traversing: By using the for loop, we can retrieve the elements of an array easily.

3) Ease of sorting: To sort the elements of the array, we need a few lines of code only.

4) Random Access: We can access any element randomly using the array.

Disadvantages of Array



1) Fixed Size: Whatever size, we define at the time of declaration of the array, we can't exceed the limit. So, it doesn't grow the size dynamically like LinkedList which we will learn later.

Declaration of C Array

We can declare an array in the c language in the following way.

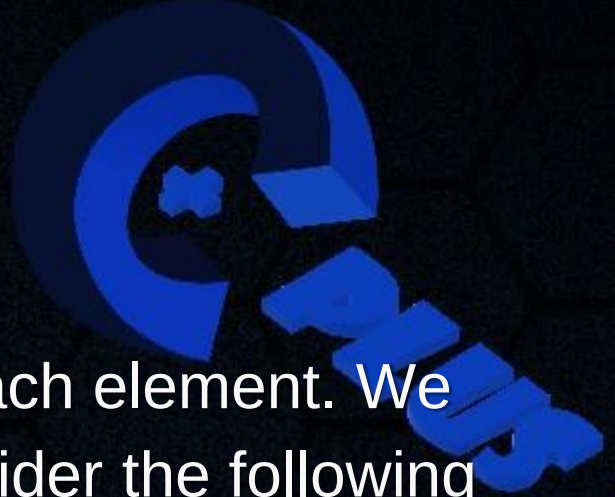
data_type array_name[array_size];

Example:

int marks[5];

Here, int is the *data_type*, marks are the *array_name*, and 5 is the *array_size*.

Initialization of C Array



The simplest way to initialize an array is by using the index of each element. We can initialize each element of the array by using the index. Consider the following example.

```
marks[0]=80;//initialization of array
marks[1]=60;
marks[2]=70;
marks[3]=85;
marks[4]=75;
```

Continue…

marks[0] marks[1] marks[2] marks[3] marks[4]

Initialization of Array

Example:

```
//program array initialization
#include<iostream>
using namespace std;
main(){
```

- **int** i=0;
- **int** marks[5];//declaration of array
- marks[0]=80;//initialization of array
- marks[1]=60;
- marks[2]=70;
- marks[3]=85;
- marks[4]=75;
- //traversal of array
- **for**(i=0;i<5;i++){
- cout<<marks[i])<<endl;
- }//end of for loop
- }

C Array: Declaration with Initialization

We can initialize the c array at the time of declaration. Let's see the code.

```
int marks[5]={20,30,40,50,60};
```

In such case, there is no requirement to define the size. So it may also be written as the following code.

```
int marks[]={20,30,40,50,60};
```

Example:

```
#include<iostream>
using namespace std;
main(){
int i=0;
int marks[5]={20,30,40,50,60};//declaration and initialization of array
 //traversal of array
for(i=0;i<5;i++){
cout<<marks[i]<<endl;
}
}
```

Accessing Array Elements

An element is accessed by indexing the array name. This is done by placing the index of the element within square brackets after the name of the array. For example:

cout<<marks[1];

Practice

Write a program for declaring, initialization and accessing of array.

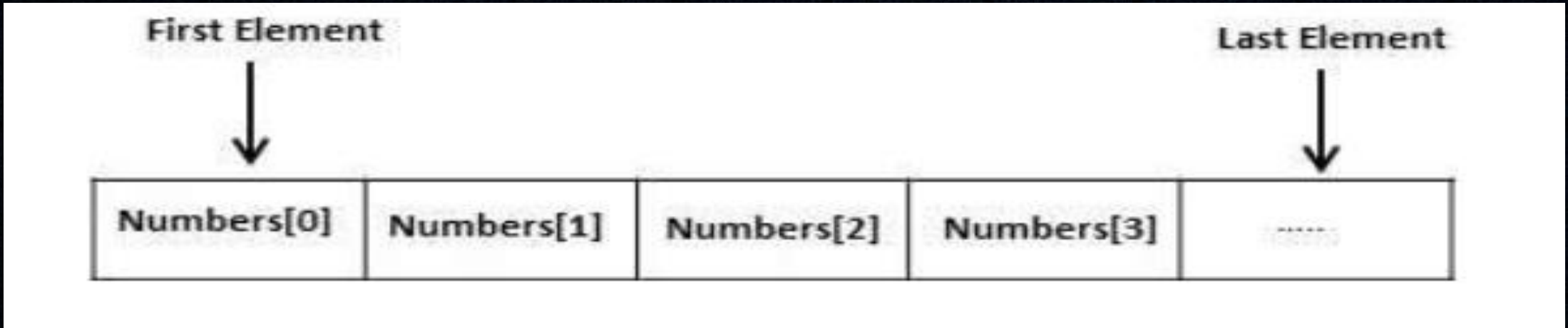
Types of Array

Basically we have two kinds of array:

1. One dimensional Array
2. Multidimensional Array

One Dimensional Array

A type of array which store elements of array in linear.



As far we have studied that was one dimensional array.

Practice

- Write a program for an array to find highest number.

Practice

- Write a program for searching an element in array

Multidimensional Array

C programming language allows multidimensional arrays. Here is the general form of a multidimensional array declaration:

type name[size1][size2]...[sizeN];

For example, the following declaration creates a three-dimensional integer array:

int threedim [5][10][4];

Two Dimensional Array

The two-dimensional array can be defined as an array of arrays. The 2D array is organized as matrices which can be represented as the collection of rows and columns. However, 2D arrays are created to implement a relational database lookalike data structure. It provides ease of holding the bulk of data at once which can be passed to any number of functions wherever required.

Declaration of two dimensional Array in C

The syntax to declare the 2D array is given below:

data_type array_name[rows][columns];

Consider the following example:

float x[3][4];

Here, 3 is the number of rows, and 4 is the number of columns.

	Column 1	Column 2	Column 3	Column 4
Row 1	x[0][0]	x[0][1]	x[0][2]	x[0][3]
Row 2	x[1][0]	x[1][1]	x[1][2]	x[1][3]
Row 3	x[2][0]	x[2][1]	x[2][2]	x[2][3]

Initialization of 2Di Array in C

In the 1D array, we don't need to specify the size of the array if the declaration and initialization are being done simultaneously. However, this will not work with 2D arrays. We will have to define at least the second dimension of the array. The two-dimensional array can be declared and defined in the following way.

```
int arr[4][3]={{1,2,3},{2,3,4},{3,4,5},{4,5,6}};
```

Accessing 2d Array Elements in C

For Accessing of 2d array we should mention the array_name and also the both indexes row and column.

array_name[i][j];

Example:

```
#include<iostream>
using namespace std;
int main(){
int i=0,j=0;
int arr[4][3]={{1,2,3},{2,3,4},{3,4,5},{4,5,6}};
//traversing 2D array
for(i=0;i<4;i++){
 for(j=0;j<3;j++){
   cout<<arr[i][j])<<endl;
 }//end of j
}//end of i
}
```

Practice

Write ✍ a program which takes values for a 3x4 matrix and print out the matrix

Practice

- Write a program to add two matrixes.

Three-Dimensional Array

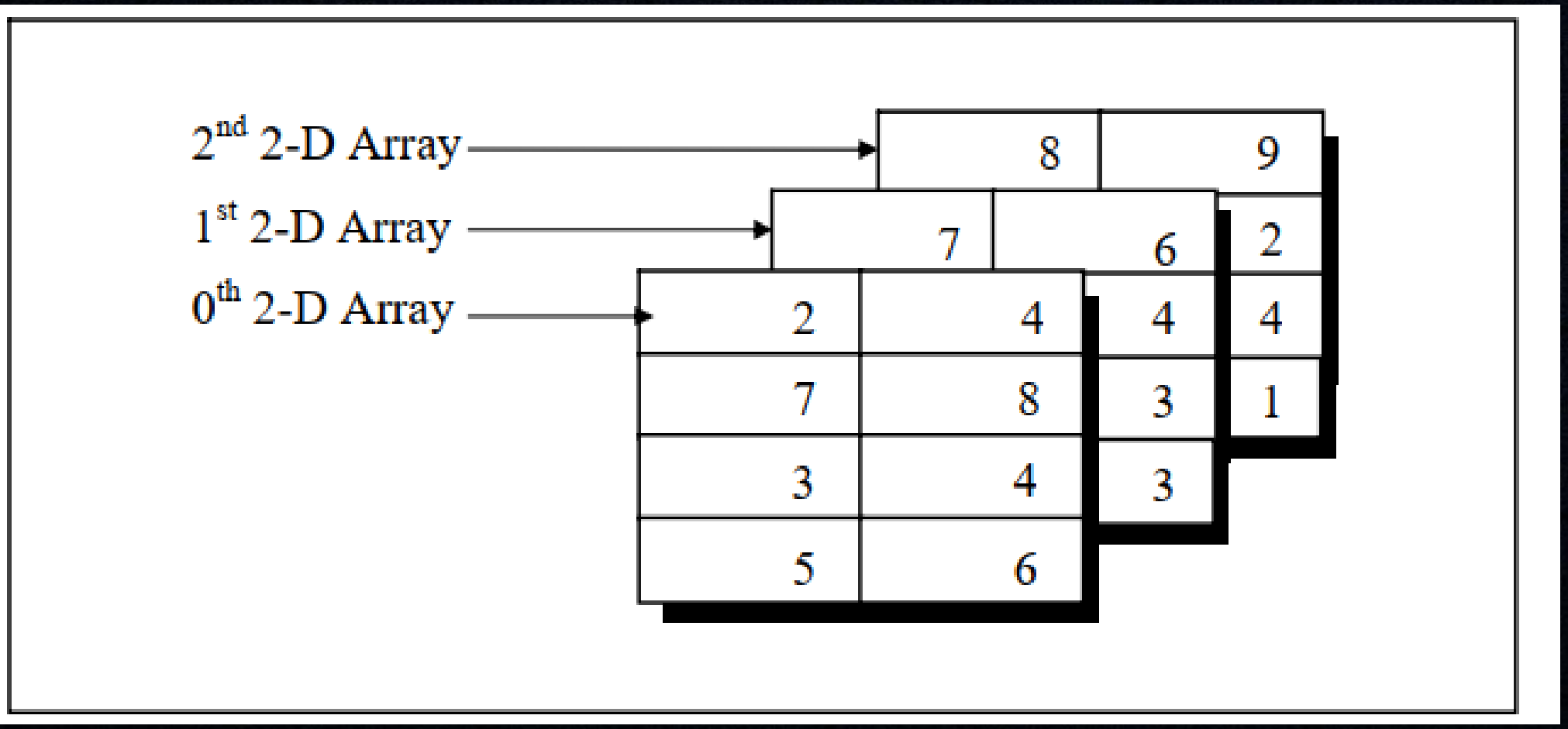
We aren’t going to show a programming example that uses a threedimensional array. This is because, in practice, one rarely uses this array. However, an example of initializing a three-dimensional array will consolidate your understanding of subscripts:

Initialization and Declaration of 3d Array

```
int arr[3][4][2] = {
{   { 2, 4 },
    { 7, 8 },
    { 3, 4 },
    { 5, 6 }
},
{   { 7, 6 },
    { 3, 4 },
    { 5, 3 },
    { 2, 3 }
},
{   { 8, 9 },
    { 7, 2 },
    { 3, 4 },
    { 5, 1 }, }
} ;
```

Continue…

A three-dimensional array can be thought of as an array of arrays of arrays.



C++ Passing One Dim Array to Function

In C++, to reuse the array logic, we can create function. To pass array to function in C++, we need to provide only array name.

functionname(arrayname); //passing array to function

C++ Passing Array to Function Example: print array elements

```
#include <iostream>
using namespace std;
void printArray(int arr[5]);
int main()
{
        int arr1[5] = { 10, 20, 30, 40, 50 };
        int arr2[5] = { 5, 15, 25, 35, 45 };
        printArray(arr1); //passing array to function
        printArray(arr2);
}
void printArray(int arr[5])
{
    cout << "Printing array elements:"<< endl;
    for (int i = 0; i < 5; i++)
    {
                cout<<arr[i]<<"\n";
    }
}
```

Passing 2 Dim Array to Function

For Passing of 2dim array you need to specify the second index in function declaration or definition, and in calling part we just pass the array name.

```
void fun(int num[][3]);
```

E.g. of Passing two dim array to function.

```
#include<iostream>
using namespace std;
void array(int num[][2]){
        for(int i = 0 ; i<=1 ; i++){
                for(int j = 0 ; j<=1 ; j++)
                cout<<num[i][j]<<endl;
        }
}
main(){
        int num[2][2]={{12,3},{3,3}};
        array(num);
}
```

Good Bye

End of 7th part, array in c language

References

Let us c: yashwant

www.javatpoint.com

www.tutorialpoint.com