



Lecture 7: Arrays and Matrices

In an array, multiple values of the same data type can be stored with one variable name. In a computer, array elements are stored in a sequence of adjacent memory locations. Arrays are of two types:

1. One-dimensional array.
2. multi-dimensional array.

✚ One-Dimensional Arrays

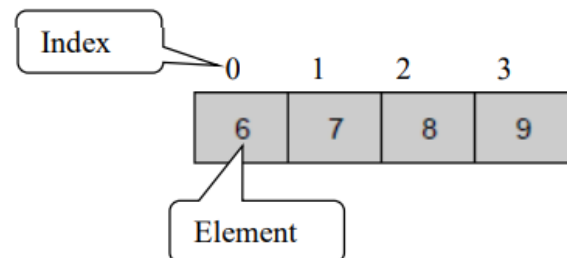
The position of an element in the array is called the array index or subscript. In the case of an array of five elements $A[4]=\{6, 7, 8, 9,\}$, their index or subscript values are 0, 1, 2, and 3. Note that the count for array elements or subscripts starts from 0 as shown below.

$A[0] = 6$

$A[1] = 7$

$A[2] = 8$

$A[3] = 9$



INPUT/OUTPUT of one-dimensional array

The input/output of an array is carried out element by element **either** a for loop or while loop may be used. For example, an array Bill[5] having n elements is to be read as follows Examples:

Ex1

```
Int Bill [5]
for (int i = 0; i < 5; i++) {
```

```

    cin>> Bill[i];
    cout << Bill[i] << "\n";
}

```

Or an array can be read in another way called "static initialization" as shown:

```
int Bill[5]={10, 20, 30,40, 50};
```

and the output (printing) is as follows:

```
for (int i = 0; i<5; i++) ;
```

```
Cout <<Bill[i]<< "\n";
```

OR for (int i = 0; i<5; i++);

```
Cout <<Bill[i]<<endl;
```

Ex2

```
int Bill[5] = {10, 20, 30, 40, 50};
for (int i = 0; i < 5; i++) {
    cout << Bill[i] << "\n";
}

```

Ex: This example outputs the index of each element together with its value.

```

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

int main() {
    string cars[5] = {"Volvo", "BMW", "Ford", "Mazda", "Tesla"};
    for (int i = 0; i < 5; i++) {
        cout << i << " = " << cars[i] << "\n";
    }
    return 0;
}

```

✚ TWO-DIMENSIONAL ARRAYS (MATRIX)

The two-dimensional array is represented by i rows and j columns. The figure below shows an array of two rows and five columns.

$A[0][0] = 5$

$A[0][1] = 2$

$A[1][0] = 6$

$A[1][3] = 9$

	0	1	2	3	4
0	5	2	3	2	4
1	6	7	8	9	8

❖ A two-dimensional array can be declared as below.

type name [number of rows] [number of columns];

For example:

```
int A[2][5];
```

```
float B[10][20];
```

INPUT/OUTPUT OF TWO-DIMENSIONAL ARRAY

The two-dimensional array $A[m][n]$ can be read as follow:

```
for(i=0; i<m; i++)
```

```
    for(j=0; j<n; j++)
```

```
        cin>>A[i][j];
```

We can use the **static initialization** with the two-dimensionally as follow:

```
float M[2][5]= {5.1, 2.2, 3.8, 2.5, 4.7, 6.1, 7.2, 8.8, 9.0, 8.4};
```

```
float M[2][5]= {{5.1, 2.2},{ 3.8, 2.5}, {4.7, 6.1}, {7.2, 8.8}, {9.0, 8.4}};
```

To print a two-dimensional array, we can use the following form:

```
for(i=0; i<m; i++)
{
    for(j=0; j<n; j++)
        cout<<A[i][j]<<" ";
    cout<<endl;
}
```

Example:

This program initializes 8 elements in a two-dimensional array of size four rows and two columns, then prints the array on output:

```
#include<iostream>
using namespace std;
int main()
{
    int arr[4][2] = {{1, 2}, {3, 4}, {5, 6}, {7, 8}};
    int i, j;
    cout<<"The Two-dimensional Array is:\n";
    for(i=0; i<4; i++)
    {
        for(j=0; j<2; j++)
            cout<<arr[i][j]<<" ";
        cout<<endl;
    }
    cout<<endl;
    return 0;
}
```

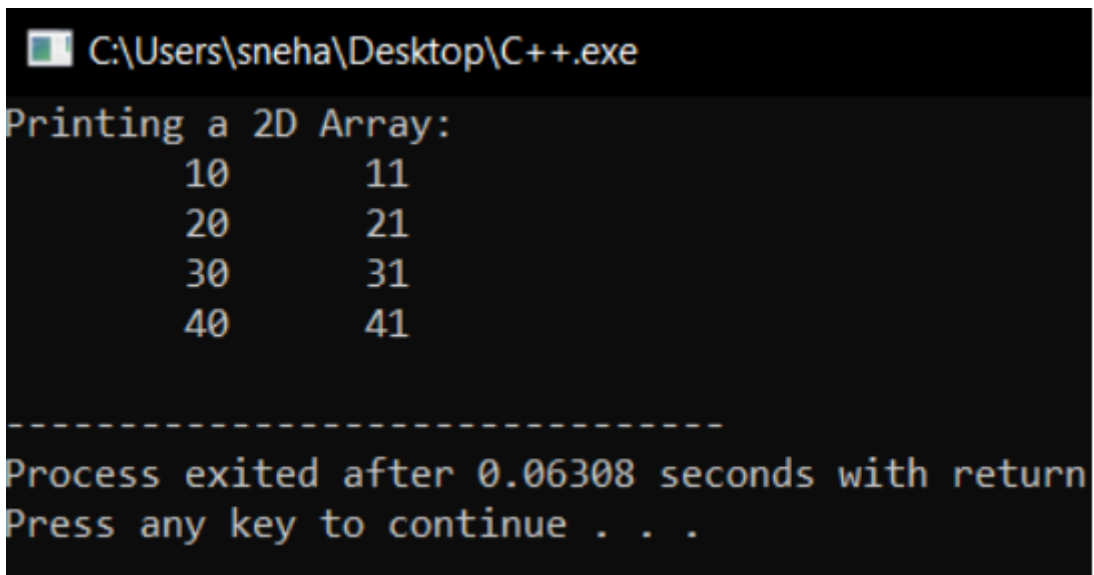


```
C:\Users\DEV\...
The Two-dimensional Array is:
1 2
3 4
5 6
7 8
```

```
#include<iostream>
using namespace std;
main( )
{
    int arr[4][2] = {
        { 10, 11 },
        { 20, 21 },
        { 30, 31 },
        { 40, 41 }
    };

    int i,j;

    cout<<"Printing a 2D Array:\n";
    for(i=0;i<4;i++)
    {
        for(j=0;j<2;j++)
        {
            cout<<"\t"<<arr[i][j];
        }
        cout<<endl;
    }
}
```



C:\Users\sneha\Desktop\C++.exe

```
Printing a 2D Array:
    10    11
    20    21
    30    31
    40    41
-----
Process exited after 0.06308 seconds with return
Press any key to continue . . .
```