Al-Mustaqbal University Collage

Biomedical Engineering Department

Class: First

Subject: Computer Skills & Programming

Lecture 7: ARRAYS AND MATRICES

BY

IT. Zahraa Abdzaid AbdelAbbas

Supervised by: Assist.Lect**.** **Halah Fadhil**

. .

ARRAYS AND MATRICES

An array contains multiple objects of identical types stored sequentially in memory. The individual objects in an array, referred to as array elements, can be addressed using a number, the so-called index or subscript. An array is also referred to as a vector.

For example:

Int a[20];

Char name[30];

In an array, multiple values of the same data type can be stored with one variable name. In 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.

1. ONE DIMENSOINAL ARRAY

A method of defining a one-dimensional array and giving it the values later.

To define an array of any type, whether int, float, double, String, etc, this way data type name [number of elements in the array].This way the computer understands that we want to define a one-dimensional array that has a specific type.

for example:

int A[10]; // Array ”A” has 10 elements of type integer.

float B[20]; // Array “B” has 20 elements of type float.

double D[15]; // Array “D” has 15 elements of type double.

char name[20]; // Array “name” has 20 elements of type char.

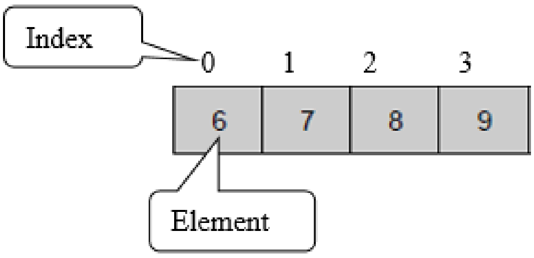
The position of an element in array is called array index or subscript. In the case of an array of four elements A[4]={6 , 7, 8, 9,}, their index or subscript values are 0, 1, 2, and 3 . Note that count for array elements or subscripts starts from 0 as shown below.

A[0] = 6

A[1] = 7

A[2] = 8

A[3] = 9



.   .

INPUT/OUTPUT OF ONE DIMENSOINAL ARRAY

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

```
for (int i = 0; i<5; i++)
cin>> Bill[i] ;
```

An array can be read by another way called "static initialization" as shown:
int Bill[5]={10, 20, 30,40, 50};

and the output (printing) is as follows:

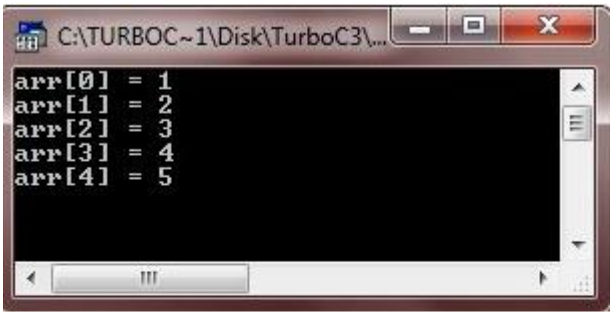
```
    for (int i = 0; i<5; i++)
       cout<< Bill[i]<<"  " ;
```

```
OR    for (int i = 0; i<5; i++)
       cout<< Bill[i]<<endl ;
```

Example1:

```
#include<iostream.h>
#include<conio.h>

int main()
{

        int arr[5] = {1, 2, 3, 4, 5};
        int i;
        for(i=0; i<5; i++)
        {
                cout<<"arr["<<i<<"] = "<<arr[i]<<"\n";
        }
        getch();
}
```

Output of Example 1:



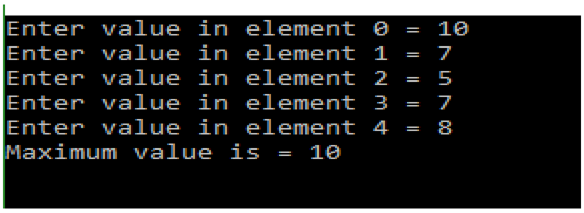
```
arr[0] = 1
arr[1] = 2
arr[2] = 3
arr[3] = 4
arr[4] = 5
```

Example2:

Example write a program which find the maximum number in one dimensional C++ array :

```
#include <iostream.h>
#include <conio.h>
 main()
{
float abc [5], max;
 int i;

for (i=0; i<=4; i++)
{
cout<<"Enter value in element "<<i<<" = ";
cin>>abc[i];
}
max = abc [0];
 for (i=1; i<=4; i++)
{
 if (max < abc[i])
max = abc [i];
}
cout << "Maximum value is = " << max;
getch();
}
```

Run screen of example 2



```
Enter value in element 0 = 10
Enter value in element 1 = 7
Enter value in element 2 = 5
Enter value in element 3 = 7
Enter value in element 4 = 8
Maximum value is = 10
```

2. 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 DIMENSOINAL 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 dimensional array 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:

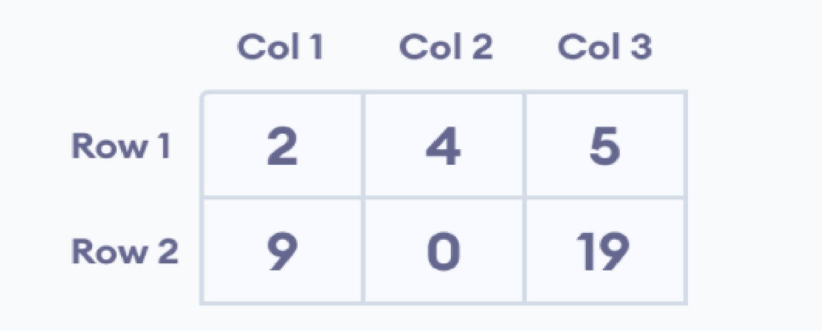
```
for(i=0; i<m; i++)
{
for(j=0; j<n; j++)
cout<<A[i][j]<<"  ";
cout<<endl;
 }
```

. .

For example

```
int  test[2][3] = { {2, 4, 5}, {9, 0, 19}};
```

This array has 2 rows and 3 columns, which is why we have two rows of elements with 3 elements each.

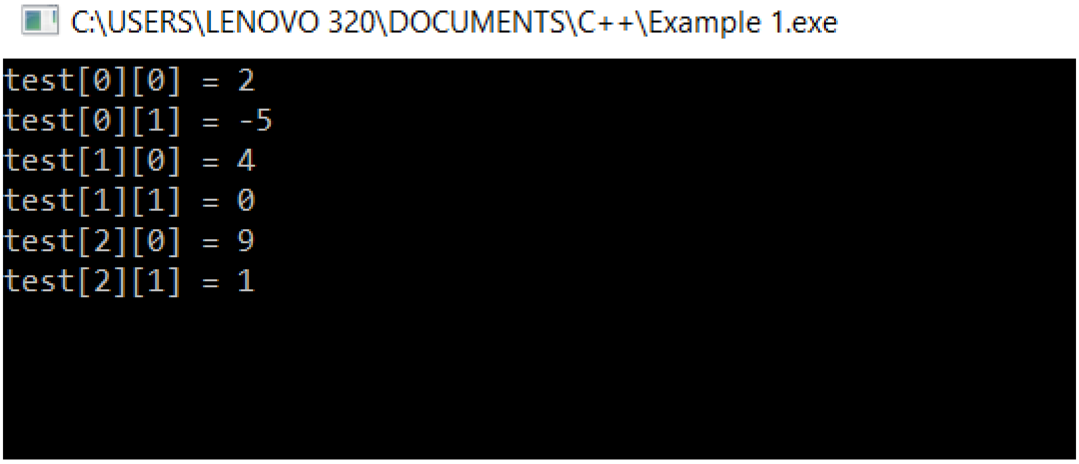


	Col 1	Col 2	Col 3
Row 1	2	4	5
Row 2	9	0	19

Example 3:

```
#include<iostream.h>
#include<conio.h>

int main()
{

        int test[3][2] = {{2, -5},
                   {4, 0},
                   {9, 1}};

    // use of nested for loop
    // access rows of the array
    for (int i = 0; i < 3; ++i)
    {

        // access columns of the array
        for (int j = 0; j < 2; ++j)
        {
            cout << "test[" << i << "][" << j << "] = " << test[i][j] << endl;
        }
    }
        getch();
}
```

Output of example 3:



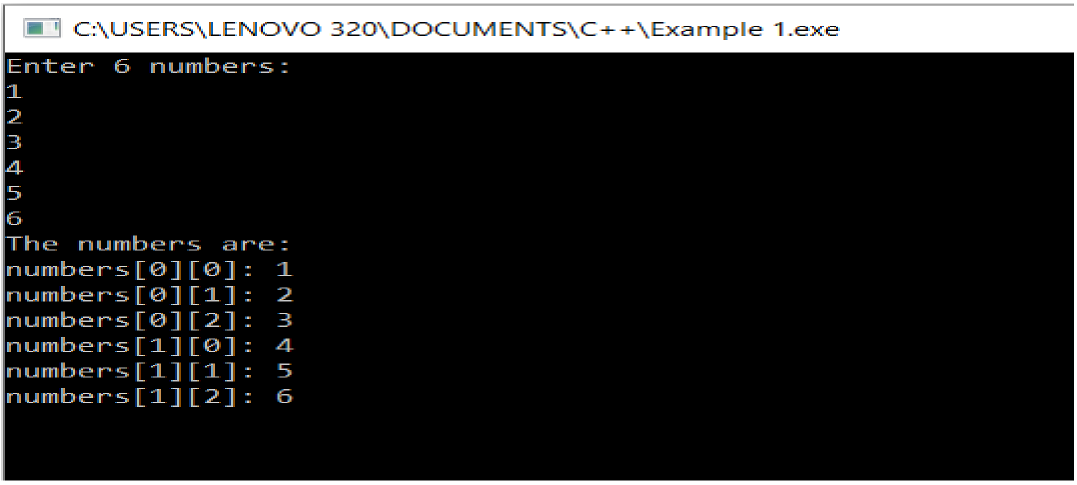
C:\USERS\LENOVO 320\DOCUMENTS\C++\Example 1.exe

```
test[0][0] = 2
test[0][1] = -5
test[1][0] = 4
test[1][1] = 0
test[2][0] = 9
test[2][1] = 1
```

Example 4:

```
#include<iostream.h>
#include<conio.h>

int main()
{

  int numbers[2][3];

    cout << "Enter 6 numbers: " << endl;

    // Storing user input in the array
    for (int i = 0; i < 2; ++i) {
        for (int j = 0; j < 3; ++j) {
            cin >> numbers[i][j];
        }
    }

    cout << "The numbers are: " << endl;

    //  Printing array elements
    for (int i = 0; i < 2; ++i) {
        for (int j = 0; j < 3; ++j) {
            cout << "numbers[" << i << "][" << j << "]: " << numbers[i][j] << endl;
        }
    }
         getch();
}
```

Output of example 4:



```
Enter 6 numbers:
1
2
3
4
5
6
The numbers are:
numbers[0][0]: 1
numbers[0][1]: 2
numbers[0][2]: 3
numbers[1][0]: 4
numbers[1][1]: 5
numbers[1][2]: 6
```

Thank you