جامعة
القاهرة الجديدة
التكنولوجية

NEW CAIRO
TECHNOLOGICAL
UNIVERSITY

# TUE – The Technological Universities in Egypt

# NCTU – New Cairo Technological University

## Faculty of Industry and Energy Technology

## Information Technology Department

## Second-Year

## Course: Programming Essentials in C++

# Lecture 8

Presented by

**Dr. Ghada Maher**

# Contents:


- Multi-dimensional array
- C++ Passing Array to Function

# Multi-Dimensional Array


- Used when data is provided in a table form.
- For Example , to store 4 Marks for 6 students.

| | M 1 | M2 | M3 | M4 |
|---|---|---|---|---|
| Student 1 | | | | |
| Student 2 | | | | |
| Student 3 | | | | |
| Student 4 | | | | |
| Student 5 | | | | |
| Student 6 | | | | |

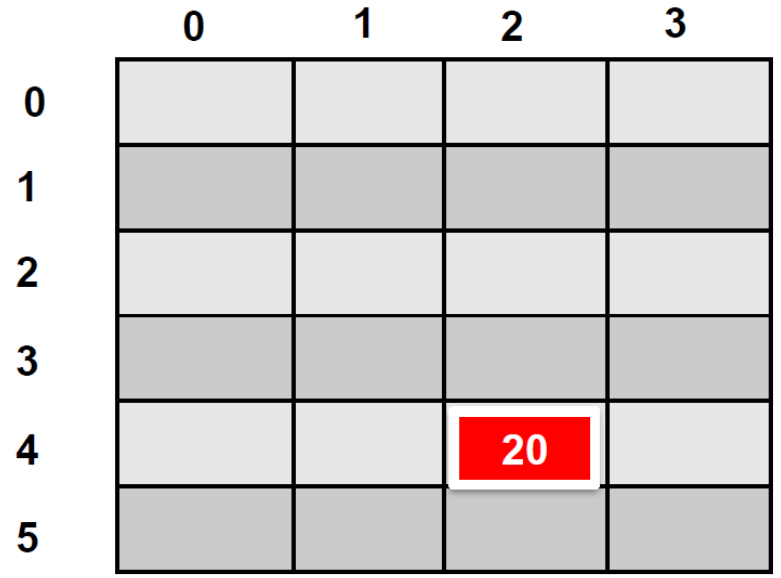
# Multi-dimensional Array declaration

- **Datatype Array Name [ Rows] [Columns]**

```
Example :

Float  marks [6] [4] ;

marks [4][2]= 20 ;
```




The multidimensional array is also known as rectangular arrays in C++. It can be two dimensional or three dimensional. The data is stored in tabular form (row * column) which is also known as matrix.

# Two dimensional Array Initialization


- Consider the declaration

```
float marks[6][4];
```

- After declaring the array you can use the For .. Loop to initialize it with values submmited by the user.

- Using 2 nested `for` loops to access array elements:

```
for (int row = 0; row < 6; row++)
    for (int col = 0; col < 4; col++)
     cin >> marks[ row ][col];
```

# Multi-dimensional Array


## Two dimensional Array Initialization

- Two dimensional Arrays can be initialized during declaration

- Example 1:

```
float marks[4][3] = {{20,30,35},
                     {40,45,65},
                     {60,65,75},
                     {80,65,45}};
```

# C++ Multidimensional Array Example


Let's see a simple example of multidimensional array in C++ which declares, initializes and traverse two dimensional arrays.

```
#include <iostream>
using namespace std;
int main()
{
  int test[3][3];  //declaration of 2D array
    test[0][0]=5;  //initialization
    test[0][1]=10;
    test[1][1]=15;
    test[1][2]=20;
    test[2][0]=30;
    test[2][2]=10;
```

```
    //traversal
    for(int i = 0; i < 3; ++i)
    {
        for(int j = 0; j < 3; ++j)
        {
            cout<< test[i][j]<<" ";
        }
        cout<<"\n"; //new line at each row
    }
    return 0;
}
```

Output:

```
5 10 0
0 15 20
30 0 10
```

# C++ Multidimensional Array Example: Declaration and initialization at same tim


```
#include <iostream>
using namespace std;
int main()
{
  int test[3][3] =
    {
        {2, 5, 5},
        {4, 0, 3},
        {9, 1, 8}  };  //declaration and initialization
    //traversal
    for(int i = 0; i < 3; ++i)
    {
        for(int j = 0; j < 3; ++j)
        {
            cout<< test[i][j]<<" ";
        }
        cout<<"\n"; //new line at each row
    }
    return 0;
}
```

Output:"

```
2 5 5
4 0 3
9 1 8
```

# C++ Passing 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";
    }
}
```

Output:

```
rinting array elements:
0
0
0
0
0
rinting array elements:

5
5
5
5
```

# C++ Passing Array to Function Example: Print minimum number


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

Output:

```
Minimum element is: 10
Minimum element is: 5
```

# C++ Passing Array to Function Example: Print maximum number


```
#include <iostream>
using namespace std;
void  printMax(int arr[5]);
int main()
{
       int arr1[5] = { 25, 10, 54, 15, 40 };
       int arr2[5] = { 12, 23, 44, 67, 54 };
       printMax(arr1); //Passing array to function
        printMax(arr2);
}
void  printMax(int arr[5])
{
   int max = arr[0];
       for (int i = 0; i < 5; i++)
       {
          if (max < arr[i])
          {
              max = arr[i];
          }
       }
       cout<< "Maximum element is: "<< max <<"\n";
}
```

Output:

```
Maximum element is: 54
Maximum element is: 67
```