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 7

Presented by

Dr. Ghada Maher

Contents:

- Arrays
  - One dimensional array
  - Multi dimensional array

Arrays

Array : is a collection of fixed number of elements, wherein all of elements have same data type.

Array Basics:

- Consecutive group of memory locations that all have the same type.
- The collection of data is indexed, or numbered, and at starts at 0 and
- The highest element index is one less than the total number of elements in the array

Single-dimensional array :

- elements are arranged in list form.

Multi-dimensional array:

- elements are arranged in tabular form.

Syntax for declaring a single-dimensional array:

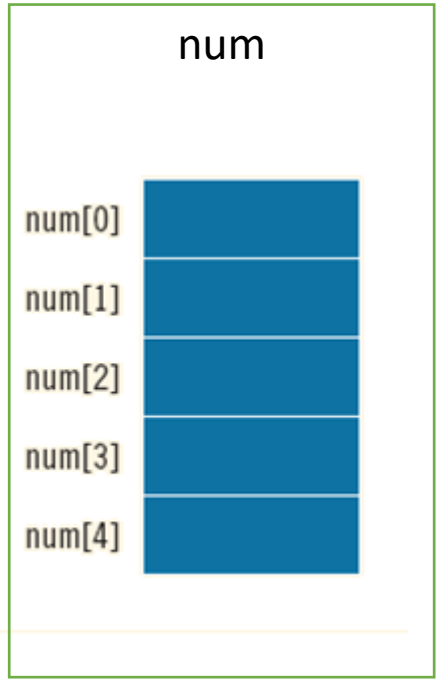
- Data type Array Name [Array Size]
- Array Size : any positive integer or constant
- Example:

  Int num [5];

- Example:

  const int size = 5;

  Int num[size];



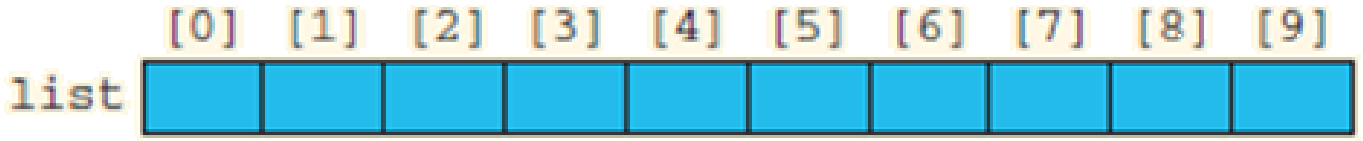
Arrays:

- Accessing array Elements

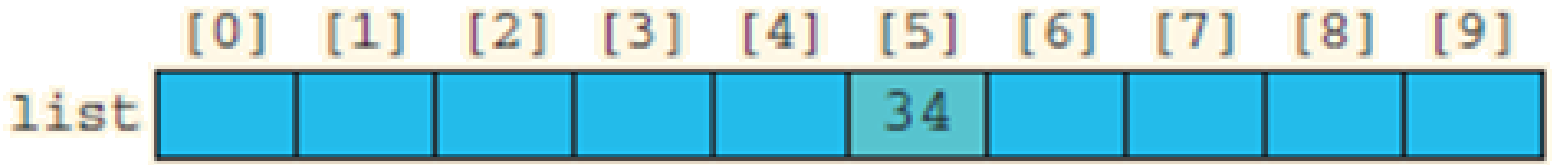
  Array name [element index]

Example:

int list [10];



List[5] = 34;



cout<< list 5

Example 1:


```
int main()
{ double a[3];
  a[2] = 55.55;
  a[0] = 11.11;
  a[1] = 33.33;
  cout << "a[0] = " << a[0] << endl;
  cout << "a[1] = " << a[1] << endl;
  cout << "a[2] = " << a[2] << endl;
}
```

```
a[0] = 11.11
a[1] = 33.33
a[2] = 55.55
```

Example 2:

This program reads five numbers and then prints them in reverse order:

```
int main()
{ const int SIZE=5;  // defines the size N for 5 elements
  double a[SIZE];    // declares the array's elements as type double
  cout << "Enter " << SIZE << " numbers:\t";
  for (int i=0; i<SIZE; i++)
    cin >> a[i];
  cout << "In reverse order: ";
  for (int i=SIZE-1; i>=0; i--)
    cout << "\t" << a[i];
}
```

```
Enter 5 numbers:        11.11   33.33   55.55   77.77   99.99
In reverse order:       99.99   77.77   55.55   33.33   11.11
```

Example 3:

This program initializes the array a and then prints its values:

```
int main()
{ float a[] = { 22.2, 44.4, 66.6 };
  int size = sizeof(a)/sizeof(float);
  for (int i=0; i<size; i++)
    cout << "\ta[" << i << "] = " << a[i] << endl;
}
```

```
        a[0] = 22.2
        a[1] = 44.4
        a[2] = 66.6
```

Example 4:

This program initializes the array a and then prints its values:

```
int main()
{ float a[7] = { 22.2, 44.4, 66.6 };
  int size = sizeof(a)/sizeof(float);
  for (int i=0; i<size; i++)
    cout << "\ta[" << i << "] = " << a[i] << endl;
}
```

```
        a[0] = 22.2
        a[1] = 44.4
        a[2] = 66.6
        a[3] = 0
        a[4] = 0
        a[5] = 0
        a[6] = 0
```

Two 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				

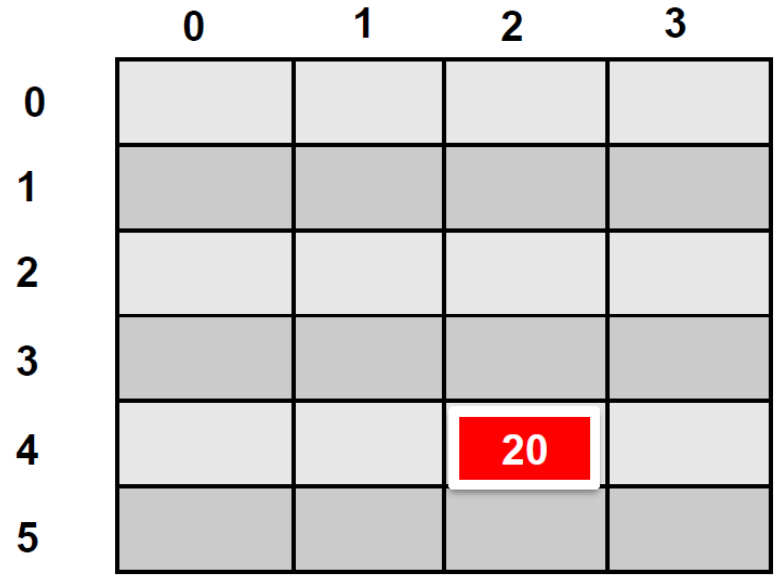
Two dimensional Array declaration

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

```
Example :

Float  marks [6] [4] ;

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



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];
```

Two 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}};
```