Lecture 07

Array

Array

Things of the same type are generally grouped together and are given a name. The simplest form of an Array is a one dimensional array that may be defined as a finite ordered set of homogeneous elements.

eg: group of cattle is called herd;

group of owls is called parliament;

group of fish is called school.

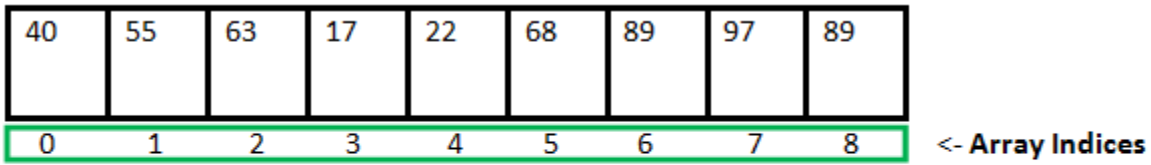
Group of data of same type is called array.

Group of Data of Same Type

`int a`	`int b`	`int c`	`int d`	`int e`
`float a`	`float b`	`float c`	`float d`	`float e`
`char a`	`char b`	`char c`	`char d`	`char e`

Array Continued.

An array is collection of items stored in contiguous memory locations.



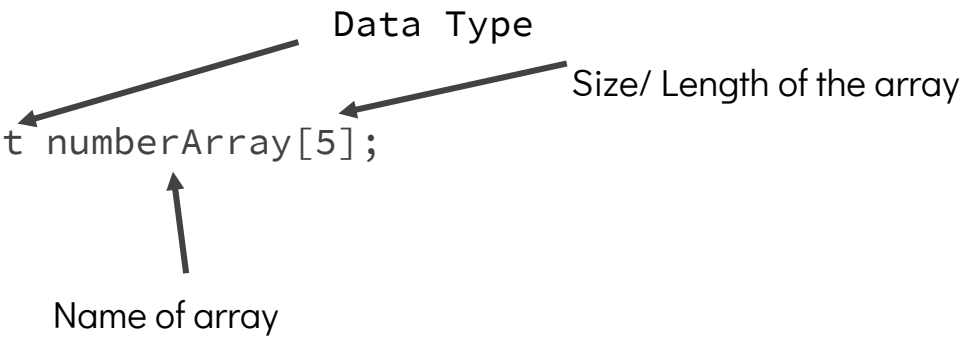
Array Length = 9
First Index = 0
Last Index = 8

Why Do We Need Arrays?

We can use normal variables (v1, v2, v3, ..) when we have small number of objects, but if we want to store large number of instances, it becomes difficult to manage them with normal variables. The idea of array is to represent many instances in one variable.

Declaring an Array

```
Syntax: data_type array_name[size];
```



Declaration By Specifying Size

1. int arr1[10].
2. With recent C/C++ versions, we can also Declare an array of user specified size .

```
int n = 10;
int arr2[n];
```

Declaration By Initializing Elements

1. int arr[] = { 10, 20, 30, 40 }
2. Compiler creates an array of size 4.
3. Above is same as

   int arr[4] = {10, 20, 30, 40}

Declaration By Specifying Size And Initializing Elements

1. Array declaration by specifying size and initializing elements
   `int arr[6] = { 10, 20, 30, 40 }`
2. Compiler creates an array of size 6, initializes first 4 elements as specified by user and rest two elements as 0.
3. Above is same as
   `int arr[] = {10, 20, 30, 40, 0, 0}`

Array of random integers

The size of an array is: the total number of bytes allocated for it

=(number of elements) * (number of bytes for each element)

Examples:

1. **int tests[5]** is an array of 20 bytes, assuming 4 bytes for an int
2. **long double measures[10]** is an array of 80 bytes, assuming 8 bytes for a long double

Array in C++ is:

---

1. Of fixed size
2. Contiguous memory locations
3. Of same data types
4. Used to store large amount of same sort of data
5. Shares same variable name

Accessing Array Elements

Arrays are accessed using their indices.

For example:

```
float temp[5] = {32, 4, -16, 25, 43};
cout << temp[0] << endl;  // prints 32
cout << temp[1] << endl;  // prints 34
cout << temp[2] << endl;  // prints -16
cout << temp[3] << endl;  // prints 25
cout << temp[4] << endl;  // prints 43
```

How about this?

```
float temp[5] = {32, 4, -16, 25, 43};

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

     cout << temp[i] << endl;

}
```

Array of Numbers

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

          numberArray[i] = i;

}

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

     cout<<numberArray[i]<<endl;

}
```

Array of Odd Numbers

```
for (int i=0; i< 5; i++){
     numberArray[i] = i*2 + 1;
}

for (int i=0; i< 5; i++){
     cout<<numberArray[i]<<endl;
}
```

Array of even numbers

```
for (int i=0; i< 5; i++){
     numberArray[i] = i*2;
}

for (int i=0; i< 5; i++){
     cout<<numberArray[i]<<endl;
}
```

Taking Input in Array

```
int marks[10];

for(int i = 0; i < 10; i++){

     cin >> marks[i];

}
```

The above code segment will take 10 integer numbers as input from the user and assign it to 10 elements of array marks

Let’s Summarize

1. Array is a continuous memory location always, no matter what its size is.
2. It is a collection of similar elements.
3. The first element of the array starts with 0, so the last element is numbered one less than its defined size.

Passing An Array To Function

To pass an array to the function as parameter, you only have to write its name without index.

Finding Number Of Odd Numbers

```
#include <iostream>
#include <cmath>
using namespace std;
int countEvenNumber(int array[], int num){
     int count = 0;
     for(int i=0; i<10; ++i){
                    if(array[i]%2 == 0){
               count = count+1;
          }
          }
  return count;
}
```

```
int main(){
          int sizeOfArray = 10;
          int
myarray[10]={1,2,3,4,5,6,7,8,9,10};
          int a=countevenNumber(myarray,10);
          cout<<"Number of odds: "<<a<<endl;
          return 0;
}
```