Practical Lecture : Array

Quick Recap

Let's take a quick recap of previous lecture –

A)

B)

C)

D)

E)

Today's Agenda

Today we are going to cover -

- Array.

- Array declaration and accessing array.

- Advantage and Disadvantage of Array.

- Multidimentional array

- Accessing Multidimensional Array

- Array Object

Let's Get Started-

Array

An array is a collection of similar items stored in contiguous memory locations. In programming, sometimes a simple variable is not enough to hold all the data. For example, let's say we want to store the marks of 500 students, having 500 different variables for this task is not feasible, we can define an array with size 500 that can hold the marks of all students
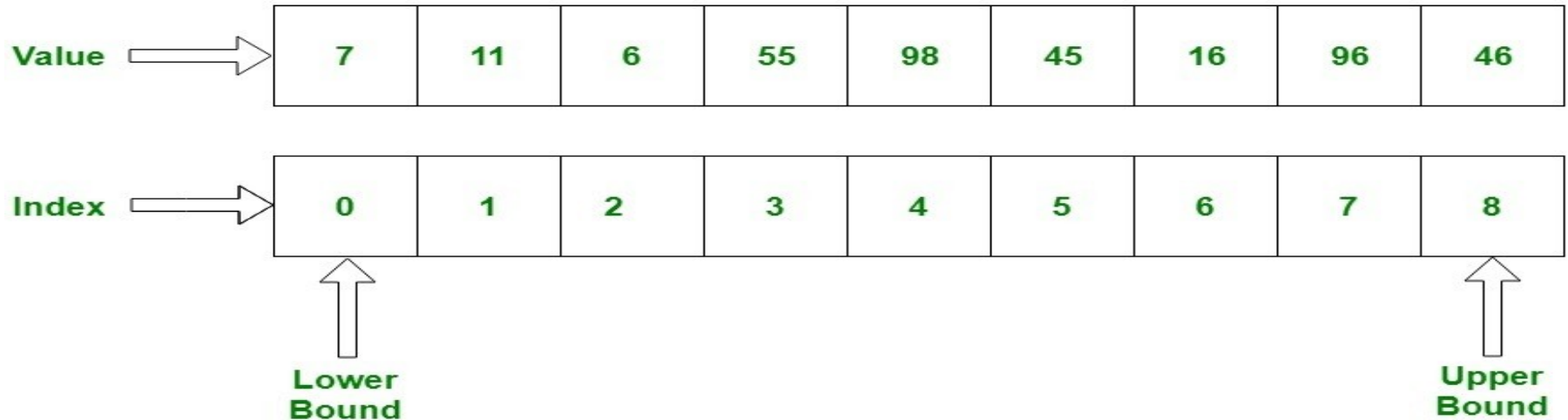
Declaring array

Method 1:-int arr[5];

         arr[0] = 10;

         arr[1] = 20;

         arr[2] = 30;

         arr[3] = 40;

         arr[4] = 50;

Method 2:-

         int arr[] = {10, 20, 30, 40, 50};

Method 3:-

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

Accessing Array

```cpp
#include <iostream>
using namespace std;


int main(){
    int arr[] = {11, 22, 33, 44, 55};
    cout<<arr[0]<<endl;
    cout<<arr[1]<<endl;
    cout<<arr[2]<<endl;
    cout<<arr[3]<<endl;
    cout<<arr[4]<<endl;
    return 0;
}
```

Advantage of Array

- Random access of elements using array index.

- Use of less line of code as it creates a single array of multiple elements.

- Easy access to all the elements.

- Traversal through the array becomes easy using a single loop.

- Sorting becomes easy as it can be accomplished by writing less line of code.

Disadvantage of Array

- Allows a fixed number of elements to be entered which is decided at the time of declaration. Unlike a linked list, an array in C is not dynamic.

- Insertion and deletion of elements can be costly since the elements are needed to be managed in accordance with the new memory allocation.

Array Rotation

Write a program of array rotation .

Input array:-
1 2 3 4 5 6

 The rotation of array by one will make :-
2 3 4 5 6 1

Input array:-
1 2 3 4 5 6

 The rotation of array by two will make :-
 3 4 5 6 1 2

Move all zeroes to end of array

Input : arr[] = {1, 2, 0, 0, 0, 3, 6}
Output : 1 2 3 6 0 0 0

Input: arr[] = {0, 1, 9, 8, 4, 0, 0, 2, 7, 0, 6, 0, 9}
Output: 1 9 8 4 2 7 6 9 0 0 0 0 0

Multidimensional Array

General Syntax:

type name[size1][size2]...[sizeN];

Eg.

int arr[3][2];

Multidimensional Array

Two Dimensional Array:

General Syntax:
type arrayName [x][y];

Eg.
int arr[3][4];

column

rows

	Column 0	Column 1	Column 2	Column 3
Row 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
Row 1	a[1][0]	a[1][1]	a[1][2]	a[1][3]
Row 2	a[2][0]	a[2][1]	a[2][2]	a[2][3]

Multidimensional Array

Initializing 2D arrays:

int a[3][4] = { {0, 1, 2, 3} , /* initializers for row indexed by 0 */
        {4, 5, 6, 7} , /* initializers for row indexed by 1 */
        {8, 9, 10, 11} /* initializers for row indexed by 2 */ };

Accessing Multidimensional Array

Takes Element from the 4th column

a[2][3]

Takes Element from the 3rd row

Suppose there is a class:

Class c1{
        int a[5][7];
};

It has an Object:
c1 *obj1*;

obj1.a[2][3]

Takes Element from the 4th column

Takes Element from the 3rd row

Array of Objects

Suppose there is a class:

```
class seat{
        int id;
        int studentid;
};
```

It has object array:
seat xii_b[10][10];

Takes Element from the 4th column

Xii_b[2][3]

Takes Element from the 3rd row

	0	1	2	3	4	5
0						
1						
2						
3						
4						
5						

Program to add two matrix

```cpp
#include <iostream>
using namespace std;
int main()
{
    int r, c, a[100][100], b[100][100], sum[100][100], i, j;

    cout << "Enter number of rows (between 1 and 100): ";
    cin >> r;

    cout << "Enter number of columns (between 1 and 100): ";
    cin >> c;

    cout << endl << "Enter elements of 1st matrix: " << endl;
     // Storing elements of first matrix entered by user.
    for(i = 0; i < r; ++i)
      for(j = 0; j < c; ++j)
      {
          cout << "Enter element a" << i + 1 << j + 1 << " : ";
          cin >> a[i][j];
      }
```

Program to add two matrix

```
// Storing elements of second matrix entered by user.
    cout << endl << "Enter elements of 2nd matrix: " << endl;
    for(i = 0; i < r; ++i)
      for(j = 0; j < c; ++j)
      {
          cout << "Enter element b" << i + 1 << j + 1 << " : ";
          cin >> b[i][j];
      }

    // Adding Two matrices
    for(i = 0; i < r; ++i)
      for(j = 0; j < c; ++j)
          sum[i][j] = a[i][j] + b[i][j];
```

Program to add two matrix

```cpp
// Displaying the resultant sum matrix.
   cout << endl << "Sum of two matrix is: " << endl;
   for(i = 0; i < r; ++i)
      for(j = 0; j < c; ++j)
      {
         cout << sum[i][j] << "  ";
         if(j == c - 1)
            cout << endl;
      }

   return 0;
}
```

Program to add two matrix

```
Enter number of rows (between 1 and 100): 2
Enter number of columns (between 1 and 100): 2

Enter elements of 1st matrix:
Enter element a11: -4
Enter element a12: 5
Enter element a21: 6
Enter element a22: 8

Enter elements of 2nd matrix:
Enter element b11: 3
Enter element b12: -9
Enter element b21: 7
Enter element b22: 2

Sum of two matrix is:
-1    -4
13    10
```

QNA Time

Any Questions ??
Any Questions??

Thank You!

See you guys in next class.