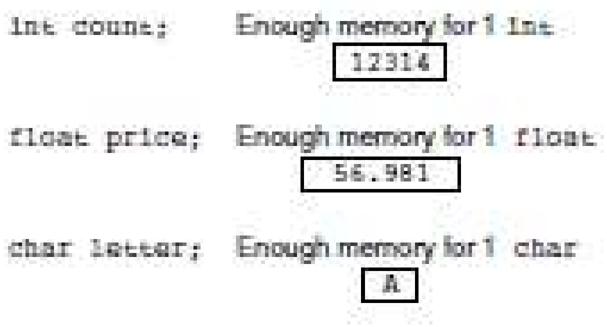
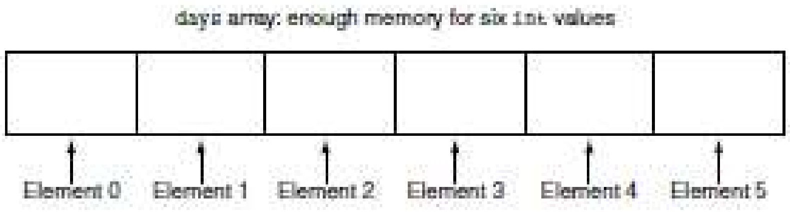
LAB 09: Arrays

Objective(s):
To Understand about:
Apply Arrays in C++
Use of One Dimensional Array
CLOs: CLO1, CLO4
INTRODUCTION:
An array works like a variable that can store a group of values, all of the same type. The values are stored together in consecutive memory locations. Here is a definition of an array of integers:
int days[6];

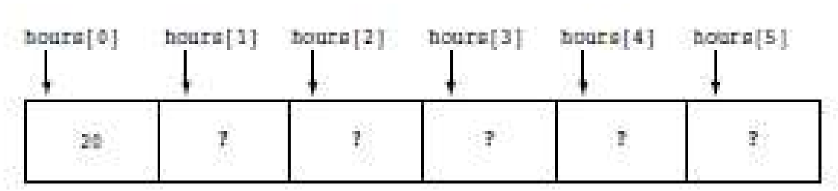


The name of this array is days. The number inside the brackets is the array's *size declarator.* It indicates the number of *elements,* or values, the array can hold. The days array can store six elements, each one an integer.
An array's size declarator must be a constant integer expression with a value greater than zero. It can be either a literal, as in the previous example, or a named constant, as shown in the following:
const int NUM_DAYS = 6;
int days[NUM_DAYS];
Arrays of any data type can be defined. The following are all valid array definitions:
float temperatures[100]; // Array of 100 floats
char name[41]; // Array of 41 characters
long units[50]; // Array of 50 long integers
double sizes[1200]; // Array of 1200 doubles



The following statement stores the integer 30 in hours[3].

hours[3] = 30;



Example:

```
// This program asks for the number of hours worked
// by six employees. It stores the values in an array.
#include <iostream>
using namespace std;

int main()
{
   const int NUM_EMPLOYEES = 6;
   int hours[NUM_EMPLOYEES];

   // Get the hours worked by each employee.
   cout << "Enter the hours worked by "
        << NUM_EMPLOYEES << " employees: ";
   cin >> hours[0];
   cin >> hours[1];
   cin >> hours[2];
   cin >> hours[3];
   cin >> hours[4];
   cin >> hours[5];

   // Display the values in the array.
   cout << "The hours you entered are:";
   cout << " " << hours[0];
   cout << " " << hours[1];
   cout << " " << hours[2];
   cout << " " << hours[3];
   cout << " " << hours[4];
   cout << " " << hours[5] << endl;
   return 0;
}
```

Program Output:

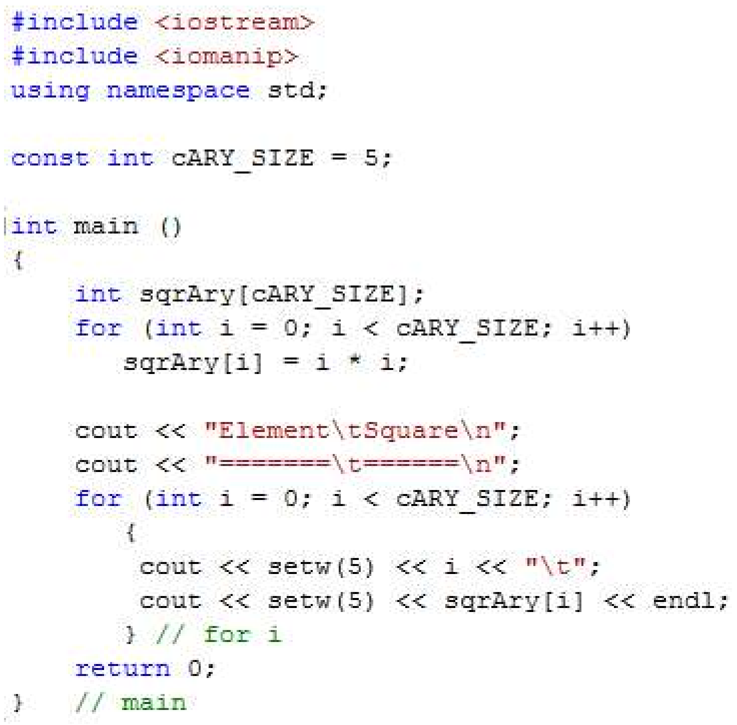
```
Enter the hours worked by 6 employees: 20 12 40 30 30 15 [Enter]
The hours you entered are: 20 12 40 30 30 15
```

Even though the size declarator of an array definition must be a constant or a literal, subscript numbers can be stored in variables. This makes it possible to use a loop to "cycle through" an entire array, performing the same operation on each element. For example, look at the following code:

```
const int ARRAY_SIZE = 5;
int numbers[ARRAY_SIZE];
for (int count = 0; count < ARRAY_SIZE; count++)
numbers[count] = 99;
```

This code first defines a constant int named ARRAY_SIZE and initializes it with the value 5. Then it defines an int array named numbers, using ARRAY_SIZE as the size declarator. As a result, the numbers array will have five elements. The for loop uses a counter variable named count. This loop will iterate five times, and during the loop iterations the count variable will take on the values 0 through 4.

Example:



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

const int cARY_SIZE = 5;

int main ()
{
    int sqrAry[cARY_SIZE];
    for (int i = 0; i < cARY_SIZE; i++)
       sqrAry[i] = i * i;

    cout << "Element\tSquare\n";
    cout << "=======\t======\n";
    for (int i = 0; i < cARY_SIZE; i++)
       {
        cout << setw(5) << i << "\t";
        cout << setw(5) << sqrAry[i] << endl;
       } // for i
    return 0;
}   // main
```

Dry Run:

Dry Run			
i	Condition	sqrAry[i]	sqrAry[i]
0	True	0	0
1	True	1	1
2	True	2	4
3	True	3	9
4	True	4	16
5	False		

Program Output:

```
//                                                                    Output
Elements        Square
========        =======
   0         0
   1         1
   2         4
   3         9
   4         16
```

Lab Tasks:

1) Array 10 elements
    1. Input
    2. Output

2) Array of 15 elements// user input

   Sum all values in array

3) Average of array in task 2

4) Array 10 elements; find highest value in the array

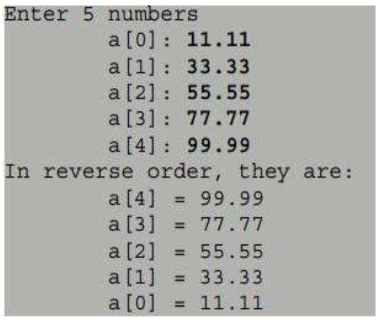
5) Array 10 elements, find lowest in array

6) Create 2 arrays, Array 10 elements each
    1. Input values in arrays
    2. Compare arrays, print equal if these are same otherwise print not equal

7) Write a program which performs the following tasks:
    1. Initialize an integer array of 10 elements in function modify( )
    2. In modify( ) multiply each element of array by 3
    3. And display each element of array

8) Write a program to search the array element. Enter a value from the keyboard and find out the location of the entered value in the array (Array 10 elements). If the entered number is not found in the array display the message ―Number is not found.

9) Write a program to input array elements and output is labeled, like this:



```
Enter 5 numbers
        a[0]: 11.11
        a[1]: 33.33
        a[2]: 55.55
        a[3]: 77.77
        a[4]: 99.99
In reverse order, they are:
        a[4] = 99.99
        a[3] = 77.77
        a[2] = 55.55
        a[1] = 33.33
        a[0] = 11.11
```