

LAB MANUAL 08

ARRAYS AND 2D ARRAYS

Lab Objectives:

At the end of this lab students will know about

- **How to use arrays in C++ programs**
- **How to use 2D arrays in C++ programs through nested loops**

Arrays

C++ provides a data structure, the array, which stores a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.

Instead of declaring individual variables, such as number0, number1, ..., and number99, you declare one array variable such as numbers and use numbers[0], numbers[1], and ..., numbers[99] to represent individual variables. A specific element in an array is accessed by an index.

All arrays consist of contiguous memory locations. The lowest address corresponds to the first element and the highest address to the last element.

DECLARING ARRAYS

To declare an array in C++, the programmer specifies the type of the elements and the number of

elements required by an array as follows –

type arrayName [arraySize];

This is called a single-dimension array. The arraySize must be an integer constant greater than zero and type can be any valid C++ data type. For example, to declare a 10-element array called balance of type double, use this statement –

double balance[10];

INITIALIZING ARRAYS

You can initialize C++ array elements either one by one or using a single statement as follows –

```
double balance[5] = {1000.0, 2.0, 3.4, 17.0, 50.0};
```

The number of values between braces { } can not be larger than the number of elements that we declare for the array between square brackets []. Following is an example to assign a single element of the array –

If you omit the size of the array, an array just big enough to hold the initialization is created. Therefore, if you write –

```
double balance[] = {1000.0, 2.0, 3.4, 17.0, 50.0};
```

You will create exactly the same array as you did in the previous example.

```
balance[4] = 50.0;
```

The above statement assigns element number 5th in the array a value of 50.0. Array with 4th index will be 5th, i.e., last element because all arrays have 0 as the index of their first element which is also called base index. Following is the pictorial representation of the same array we discussed above –

ACCESSING ARRAY ELEMENTS

An element is accessed by indexing the array name. This is done by placing the index of the element within square brackets after the name of the array. For example –

```
double salary = balance[9];
```

The above statement will take 10th element from the array and assign the value to salary variable. Following is an example, which will use all the above-mentioned three concepts viz. declaration, assignment and accessing arrays.

SYNTAX OF TWO-DIMENSIONAL ARRAY:-

(Data type) (Name of array) [Number of rows] [Number of columns];

For example:-

```
Int matrix [7] [7];
```

When we type the above statement the compiler will generate a 2-D array of a matrix which consists of 7 rows and 7 columns.

ACCESSING EACH LOCATION OF TWO DIMENSIONAL ARRAYS:-

In order to store values in a C++ two dimensional arrays the programmer have to specified the

number of row and the number of column of a matrix. To access each individual location of a

matrix to store the values the user have to provide exact number of row and number of column.

For Example:-

```
int matrix [2][2];

matrix [0] [0] = 43;

matrix [0] [1] = 44;

matrix [1] [0] = 56;

matrix [1] [1] = 72;
```

HOW TO ENTER DATA IN A TWO DIMENSIONAL ARRAYS:-

Nested loop is used to enter data in 2-D arrays. It depends upon the programmer which loop he wants to use it could be While loop or it could be a For loop. The outer loop acts as the number of rows of a matrix and the inner loop acts as the number of columns of a matrix.

For Example:-

```
#include<iostream>

using namespace std;

main()

{

int matrix [5] [5];

for (int m1=0 ; m1<5 ; m1++)

{

for (int m2=0 ; m2<5 ; m2++)

{

matrix [m1] [m2] = 5 ;

}

}

getch();

}
```

The above portion of a code uses nested For loop to assigns 5 to all the locations of a matrix of a two dimensional array in C++.

HOW TO INITIALIZE TWO-DIMENSIONAL INTEGER ARRAY:-

The 2-D arrays which are declared by the keyword int and able to store integer values are called

two dimensional integer arrays. The process of assigning values during declaration is called initialization. These Arrays can be initialized by putting the curly braces around each row separating by a comma also each element of a matrix should be separated by a comma.

EXAMPLE OF A TWO DIMENSIONAL INTEGER ARRAY:-

```
int mat [3][3]= {
{ 3,6,8 },
{ 5,4,7 },
{ 2,4,7 }
};
```

As you can see the above array is declared by a keyword int therefore it is a 2-D integer array.

Now consider a following example of a complete program which will elaborate the working.

```
#include<iostream>
using namespace std;
main()
{
int matrix [2][3];
// For taking integer inputs in a matrix //
for (int m1=0 ; m1<2 ; m1++)
{
for (int m2=0 ; m2<3 ; m2++){
cout<<"Enter Integer :";
cin>>matrix [m1][m2];
cout<<endl;}}
// For displaying elements of
//a matrix on a screen //
for (int m1=0 ; m1<2 ; m1++)
{
for (int m2=0 ; m2<3 ; m2++)
cout<<"Your Entered Integer are :";
cout<<matrix [m1][m2];
```

```
cout<<endl; }}
```

Explanation:-

In a main function first of all the two dimensional integer array will be declared by the name of a matrix, then the integer elements will be stored in an array with the help of a nested For loop. Then again with the help of another nested loop the program will print the elements stored in the matrix on a computer screen.

Lab Tasks

Question # 1

Read the entries of an array of 10 integers from a user. Compute x as the average of the 10 entries and then compute the average and display those entries that are greater than or equal to x. Print this final average.

Question # 2

Write a C++ code to find the minimum and maximum distance between two numbers of an array.

Question # 3

Take input 10 numbers from user, sort them in ascending and descending order.

Question # 4

Take array of 5 numbers from user, now print them in reverse order.

Question # 5

Take 10 float numbers from user, now find second greatest number from array.

Question # 6

Take array of 10 numbers, now find smallest number in array and make it the greatest number in array and then print new array.

Question # 7

Take 10 numbers from user, now display most occurring element and also its number of occurrence.

Question # 8

Write a C++ program to generate the sum of left diagonal.

Question # 9

Write a C++ program to find the duplicate values in a 2d array.

Question # 10

Write a C++ program to move all negative elements of an array of integers to the end of the array without changing the order of positive element and negative element.

Question # 11

Write a C++ Program to store temperature of two different cities for a week and display it. Find the city with hottest temperature.

Question # 12

Write a C++ program to generate transpose of 3×3 matrix