

Riphah International University
I-14 Main Campus
Faculty of Computing

Class:	Fall-2024	Subject:	Data Structures & Algorithms
Course Code:	CS 2124	Lab Instructor:	Zeeshan Ali

Learning Objective:

1. What is Two Dimensional Arrays and why do we use Two Dimensional Arrays?
2. Example of Two Dimensional Arrays in C++.
3. What are pointers in C++?
4. Why do we use Pointers?
5. Syntax of Pointers in C++.
6. Example of Pointers in C++.
7. Lab Tasks

1. Two Dimensional Arrays:

In C++ Two Dimensional array in C++ is an array that consists of more than one rows and more than one column. In 2-D array each element is refer by two indexes. Elements stored in these Arrays in the form of matrices. The first index shows a row of the matrix and the second index shows the column of the matrix.

- A row has horizontal elements.
- A column has vertical elements.

Columns				
Rows	2D Array	0	1	2
	0	a[0][0]	a[0][1]	a[0][2]
	1	a[1][0]	a[1][1]	a[1][2]
	2	a[2][0]	a[2][1]	a[2][2]

Use of Two Dimensional Arrays

Two dimensional arrays are especially useful when the data is naturally organized in rows and columns like in a spreadsheet, theater seats or classroom seats.

Syntax of Two-Dimensional Array:-

(Data type) (Name of array) [Number of rows] [Number of columns];
int matrix [7] [7];

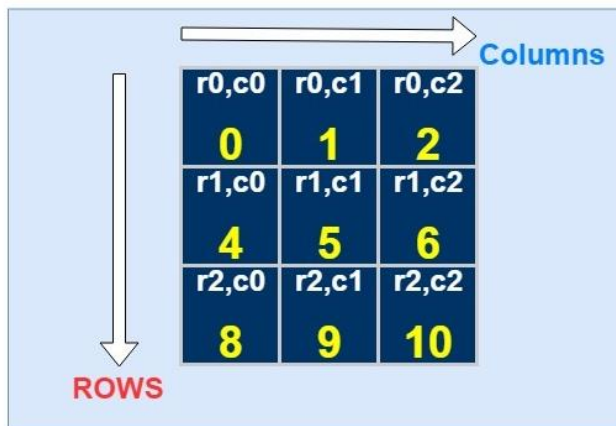
Accessing Elements of Two Dimensional Arrays:-

Nested loop is used to access each individual location of a 2D array.

```
pointer.cpp TwoDArray.cpp
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      cout<<"Two diminsional array"<<endl;
6      int TwoDArr[3][3] = { {0,1,2}, {4,5,6}, {8,9,10} }; /// Declare an initialize 2D integer array with 9 elements
7      // For displaying elements of a TwoDArr on a screen //
8      for(int i=0; i<3; i++) // for rows
9      {
10         for(int j =0 ; j<3; j++) // for columns
11         {
12             //Displying TwoDArr array
13             cout<<"Values of Two Diminsional Array at row ["<<i<<" column["<<j<<" = "<< TwoDArr[i][j]<<endl;
14         }
15     }
16
17     return 0;
18 }
```

```
Two diminsional array
Values of Two Diminsional Array at row [0] column[0] = 0
Values of Two Diminsional Array at row [0] column[1] = 1
Values of Two Diminsional Array at row [0] column[2] = 2
Values of Two Diminsional Array at row [1] column[0] = 4
Values of Two Diminsional Array at row [1] column[1] = 5
Values of Two Diminsional Array at row [1] column[2] = 6
Values of Two Diminsional Array at row [2] column[0] = 8
Values of Two Diminsional Array at row [2] column[1] = 9
Values of Two Diminsional Array at row [2] column[2] = 10
-----
Process exited after 1.387 seconds with return value 0
Press any key to continue . . .
```

```
int x[3][3] = {{0,1,2},{4,5,6},{8,9,10}}
```



	Column 0	Column 1	Column 2	Column 3
Row 0	0,0	0,1	0,2	0,3
Row 1	1,0	1,1	1,2	1,3
Row 2	2,0	2,1	2,2	2,3

2. Example of Two Dimensional Arrays

```
pointer.cpp TwoDArray.cpp TwoDArrayUserInput.cpp
2 using namespace std;
3 int main()
4 {
5     cout<<"Two diminsional arrays"<<endl;
6     int TwoDArr[2][3];
7     // For taking integer inputs in a TwoDArr //
8     for(int i=0; i<2; i++)
9     {
10         for(int j =0 ; j<3; j++)
11         {
12             cout<<"Enter values of Two Diminsional Array at row ["<<i<< " ] column["<< j<<" ] = ";
13             cin>>TwoDArr[i][j];
14         }
15     }
16     // For displaying elements of a TwoDArr on a screen //
17     for(int i=0; i<2; i++)
18     {
19         for(int j =0 ; j<3; j++)
20         {
21             cout<<"Value of matrix at row ["<<i<< " ] column["<< j<<" ] ="<<TwoDArr[i][j]<<endl;
22         }
23     }
24     return 0;
25 }
```

```

Two diminsional arrays
Enter values of Two Diminsional Array at row [0] column[0] = 5
Enter values of Two Diminsional Array at row [0] column[1] = 6
Enter values of Two Diminsional Array at row [0] column[2] = 7
Enter values of Two Diminsional Array at row [1] column[0] = 9
Enter values of Two Diminsional Array at row [1] column[1] = 3
Enter values of Two Diminsional Array at row [1] column[2] = 45
Value of matrix at row [0] column[0] =5
Value of matrix at row [0] column[1] =6
Value of matrix at row [0] column[2] =7
Value of matrix at row [1] column[0] =9
Value of matrix at row [1] column[1] =3
Value of matrix at row [1] column[2] =45

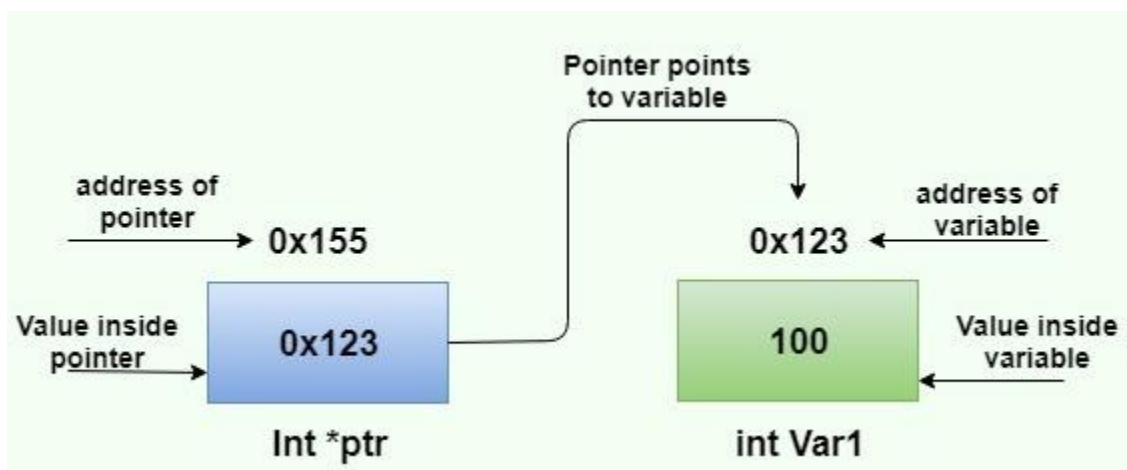
-----
Process exited after 44.86 seconds with return value 0
Press any key to continue . . .

```

3. Pointer

In C++, a pointer refers to a variable that **holds the address** of another variable. Like regular variables, pointers have a data type.

For example, a **pointer of type integer** can hold the address of a **variable of type integer**. A pointer of character type can hold the address of a variable of character type.



4. Why do we use Pointers?

Execution time with pointers is **faster** because data are manipulated with the address, that is, direct access to memory location. Memory is accessed efficiently with the pointers.

Pointers are used to allocate and deallocate memory dynamically on the heap. This enables the creation of data structures of variable size at runtime, such as dynamic arrays and linked lists.

5. Syntax of Pointers:-

(Data type) (*)(Name);

Data type *NAME;

```
int *IPTR;
```

6. Example of pointer in C++

```
pointer.cpp TwoDArray.cpp TwoDArrayUserInput.cpp pointerExample.cpp
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      cout<<"Pointers"<<endl;
6      int *PTR; // declaring int type pointer (holds address of integer variable) : pointer to integer
7      float *Q ; // pointer to float
8      char *C ; // pointer to character
9      int X; // declaring integer variable
10     cout<<"Enter integer number "<<endl;
11     cin>>X; // integer input from user
12     // & is refrencing or address of operator
13     PTR = &X; // intializing pointer
14     cout<<"The value of integer is (*PTR) "<<*PTR<<endl; // * is derefrencing operator (value at PTR)
15     cout<<"Address of integer is (&X) "<< &X<<endl;
16     cout<<"Address of integer (PTR) "<< PTR<<endl;
17     return 0;
18 }
19
```

```
Pointers
Enter integer number
96
The value of integer is (*PTR) 96
Address of integer is (&X) 0x6ffe04
Address of integer (PTR) 0x6ffe04

-----
Process exited after 13.32 seconds with return value 0
Press any key to continue . . .
```

7. Lab Tasks

1. Write a program in C++ that take integer type 2D array from user, calculate sum multiplication and average of all numbers.
2. Write a program in C++ to swap values of two variables using pointers.
3. Write a program that lets the user to enter the 10 values into the array. The program should then display the largest and the smallest values stored in the array.
4. Write a program that lets the user to enter the total rainfall for each of 12 months into an array of doubles. The program should calculate and display the total rainfall for the year, the average monthly rainfall and the month with the highest and lowest rainfall.
5. Write a program that creates a two dimensional array initialized with test data.

Perform the following operations on the array data.

Get total: get total of all the elements in the array.

Get Average: calculate the average of all the values in the array.

Get row total: calculate the total of all the values in the specified row.

Get column total: calculate the total of all the values in the specified column.

Get Highest in row: find the highest value in the specified row.

Get Highest in Column: find the highest value in the specified column.

6. Write a program that dynamically allocates an array of integers. Read the values from user and calculate the sum of odd integers.
7. Define a pointer variable. Assign the address of variable to a pointer variable and access the value of address variable in the pointer variable.
8. Write a program that asks the user to enter integers as inputs to be stored in the variables 'a' and 'b' respectively. There are also two integer pointers named ptrA and ptrB. Assign the values of 'a' and 'b' to ptrA and ptrB respectively, and display them.
9. Write a program for a calculator using functions.
Your program must have the following functions:
 - a. Menu () telling the user to select from the options
 - b. Addition (int a, int b) adding two numbers
 - c. Subtraction (int a, int b)
 - d. Division (int a, int b)
 - e. Multiplication (int a, int b)
 - f. Pow (int number, int pow)In the main function you will only call the menu () function.