**Logo, company name

Description automatically generated**

**Department of (Computer Science)**

**Pak-Austria Fachhochschule: Institute of Applied Sciences and Technology, Haripur, Pakistan**

**COMP-112L Object Oriented Programming Lab**

**Lab Journal**

**Class: BS Computer Science**

**Name: Ahmed Raza**

**Registration No.: B20F0436CS031**

**Semester: 3rd**

**Submission Date: 08 February, 2022**

**Submitted to: Engr. Rafi-Ullah**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Instructor Signature**

**Lab No. 02**

**Pointers in C++**

**Objectives:**

In this lab we will be discussing pointers in detail. This is one of the most important concepts in C++ language. Pointers are used everywhere in C++, so if you want to use the C++ language fully you have to have a very good understanding of pointers. They have to become comfortable for you. C++ uses pointers in three different ways:

* C++ uses pointers to create dynamic data structures -- data structures built up from blocks of memory allocated from the heap at run-time.
* C++ uses pointers to handle variable parameters passed to functions.
* Pointers in C++ provide an alternative way to access information stored in arrays. Pointer techniques are especially valuable when you work with strings. There is an intimate link between arrays and pointers in C++. To fully grasp the concept of pointers all you need is the concept and practice of pointers.

**Tools/Software Required:**

* All the tasks are implemented on DEV C++.

**Introduction:**

**Computer Memory**

* Essentially, the computer's memory is made up of bytes. Each byte has a number, an address, associated with it.

**Variable and Computer Memory**

* A variable in a program is something with a name, the value of which can vary. The way the compiler handles this is that it assigns a specific block of memory within the computer to hold the value of that variable. The size of that block depends on the range over which the variable is allowed to vary.

**Pointer**

* In C++ a pointer is a variable that points to or references a memory location in which data is stored. A pointer is a variable that points to another variable. This means that a pointer holds the memory address of another variable. Put another way, the pointer does not hold a value in the traditional sense; instead, it holds the address of another variable. A pointer "points to" that other variable by holding a copy of its address. Because a pointer holds an address rather than a value, it has two parts. The pointer itself holds the address and that address points to a value.

**Pointer declaration**

* A pointer is a variable that contains the memory location of another variable. The syntax is as shown below. You start by specifying the type of data stored in the location identified by the pointer. The asterisk tells the compiler that you are creating a pointer variable. Finally you give the name of the variable.
* Data\_type \*variable\_name

**Function Pointers**

* A function pointer is a variable that stores the address of a function that can later be called through that function pointer. A useful technique is the ability to have pointers to functions. Their declaration for the function is:
* int func(int a, float b);

**Pointers and arrays**

The concept of arrays is related to that of pointers. In fact, arrays work very much like pointers to their first elements, and, actually, an array can always be implicitly converted to the pointer of the proper type. For example, consider these two declarations:

* int myarray [20];
* int \* mypointer;

The following assignment operation would be valid:

* mypointer = myarray;

After that, mypointer and myarray would be equivalent and would have very similar properties. The main difference being that mypointer can be assigned a different address, whereas myarray can never be assigned anything, and will always represent the same block of 20 elements of type int. Therefore, the following assignment would not be valid:

* myarray = mypointer;

**Lab Tasks:**

**Task 1:**

Write a C++ program to find to add two number using function pointers. Define the function Adder which return the sum of the numbers.

**Code:**

**#include<iostream>**

**using namespace std;**

**int adder(int \*a, int \*b);**

**int main()**

**{**

**int a, b, x;**

**cout<<"Enter value in a : ";**

**cin>>a;**

**cout<<"Enter value in b : ";**

**cin>>b;**

**x = adder(&a, &b);**

**cout<<"Sum of two number is : "<<x<<endl;**

**return 0;**

**}**

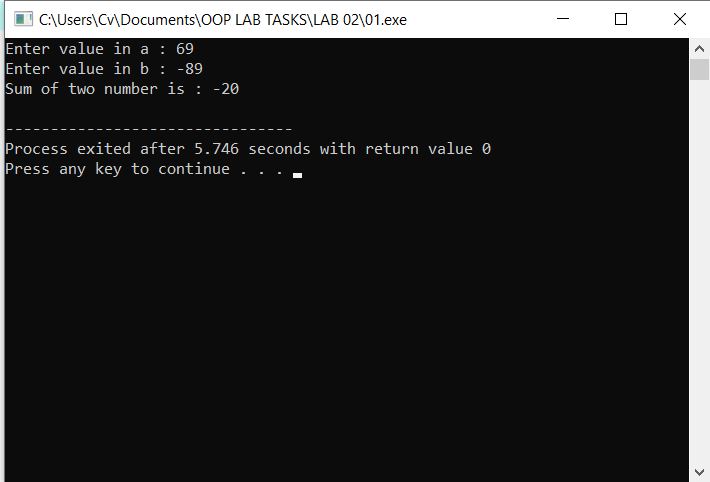
**int adder(int\*x, int\*y)**

**{**

**return (\*x + \*y);**

**}**

**Output:**



**Task # 02:**

Write a C++ program which sorts an array of numbers in ascending order. You can take the array from the user or initialize it on compile time in the program. Use a Function Pointers concept to sort the given array.

**Code:**

**#include<iostream>**

**using namespace std;**

**void sort(int\* arr, int size);**

**int main()**

**{**

**int size;**

**cout<<"Enter the size of array : ";**

**cin>>size;**

**cout<<endl;**

**int arr[size];**

**for(int i=0; i<size; i++)**

**{**

**cout<<"Enter the value at index "<<i<<":";**

**cin>>arr[i];**

**}**

**sort(arr, size);**

**for(int i=0; i<size; i++)**

**{**

**cout<<arr[i]<<" ";**

**}**

**return 0;**

**}**

**void sort(int\* arr, int size)**

**{**

**for(int i=0; i<size; i++)**

**{**

**for(int j=0; j<size; j++)**

**{**

**if(\*(arr+i)<\*(arr+j))**

**{**

**int temp = \*(arr+i);**

**\*(arr+i)= \*(arr+j);**

**\*(arr+j)=temp;**

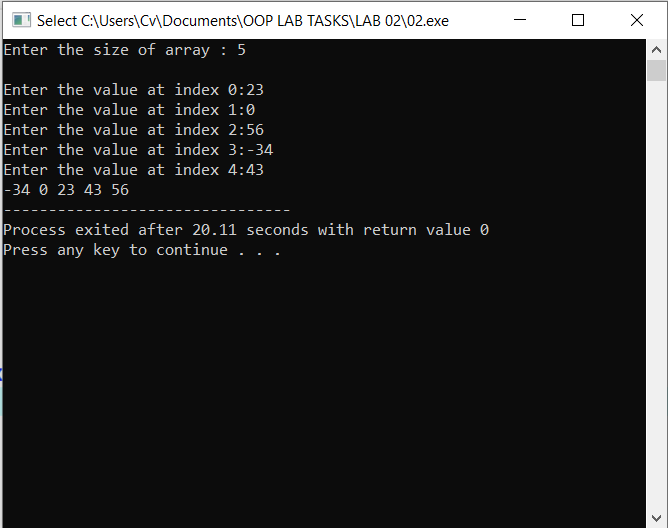
**}**

**}**

**}**

**}**

**Output:**



**Results & Observations:**

In this Lab I’ve learned about the concept of arrays with pointers in the function and also understand that how array can be passed to a function with the help of pointers & returned to a main function. In the first task, I’ve used adder function that’s actually passing argument of pointers and in the second I’ve used a sort function, passing pointer array and size of that array as an argument. To arrange in ascending order of the element of array, I’ve created a temp variable in the sort function.