**Logo, company name

Description automatically generated**

**Department of (Computer Science)**

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

**COMP-112L Data Structure** **& Algorithm Lab**

**Lab Journal**

**Class: BS Computer Science**

**Name: Ahmed Raza**

**Registration No.: B20F0436CS031**

**Semester: 4th**

**Submission Date: 26th April 2022**

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

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

**Instructor Signature**

**Lab No. 02**

**Structures &** **Pointers Using C++**

**Objectives:**

In this lab we will be discussing about Structures & Pointers in detail. This is one of the most important concepts in C++ language A struct is a derived data type that consists of members that are each fundamental or derived data types. struct is used to declare a new datatype. Basically, this means grouping variables together.

In C++ it is required to do some meaningful operations on structures they are:

* Structure Declaration
* Structure Definition
* Structure Variables
* Structure Membership
* Arrays of structures

**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:**

**Arrays of Character**

**STRUCTURES**

Arrays require that all elements be of the same data type. Many times it is necessary to group information of different data types. An example is a materials list for a product. The list typically includes a name for each item, a part number, dimensions, weight, and cost. C and C++ support data structures that can store combinations of character, integer floating point and enumerated type data. They are called a **STRUCTS**.

**Structures Definition**

Before a structure is created, it is necessary to define its overall composition. The format of the a structure is provided in the shape of a template or pattern which is then used to creatstructure variables of the same composition. The template is composed of the names and attributes of the data items to be included in the structure. The definition begins with the keyword **struct** which is followed by a structure declaration consist of a set of user-defined data names and data types. These entries are separated by semicolons and enclosed within a pair of curly brackets. The definition ends with a semicolon. Thus, in general, the structure definition has the form

**struct tag-name**

**{**

**type var-1;**

**type-var-2; . . . . . . . . typevar-n;**

**}**;

Where **tag-name** is the user-supplied name that identified the structure template; **type** refers to any valid data type such as char, int, float, and so forth; and var-1, var-2, ….var-n are user-defined variables Page 42 . . . names, arrays or pointers. The components of a structure are commonly referred to as **members or** **field**.

**Pointer**

* Pointers are used for:
* Accessing array elements.
* Passing arguments to a function when the function needs to modify the original argument.
* Passing arrays and strings to functions.
* Obtaining memory from the system.
* Creating data structures such as linked lists.

**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);

**Lab Tasks:**

**Task 1:**

Write a program to swap two values by passing pointers as argument to the function.

**Code:**

**#include<iostream>**

**using namespace std;**

**int swap(int \*p1,int \*p2);**

**int main()**

**{**

**int a, b;**

**int \*p1,\*p2;**

**cout<<"Enter frist number: ";**

**cin>>a;**

**cout<<"Enter second number: ";**

**cin>>b;**

**cout<<"\nBefore Swapping values"<<endl;**

**cout<<"Value of a is "<<a<<" and value of b is "<<b<<endl;**

**p1=&a;**

**p2=&b;**

**swap(\*p1,\*p2);**

**cout<<"After Swapping values"<<endl;**

**cout<<"Value of a is "<<a<<" and value of b is "<<b<<endl;**

**}**

**int swap(int \*p1,int \*p2)**

**{**

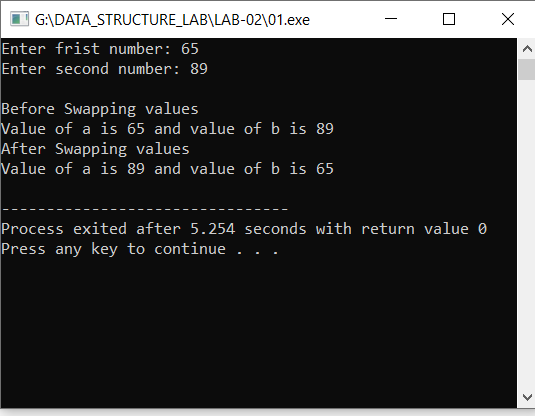
**int s=\*p1;**

**\*p1=\*p2;**

**\*p2=s;**

**}**

**Output:**

****

**Task # 02:**

Write a program to convert Fahrenheit temperature to Celsius degrees by passing pointers as arguments to the function. The formula for the conversion is: c = ( f-32) \*5 .0/9.0.

**Code:**

**#include<iostream>**

**using namespace std;**

**void convert(int \*p);**

**int main()**

**{**

**int temp;**

**int \*p;**

**cout<<"Enter temperature in Fahrenheit: ";**

**cin>>temp;**

**cout<<"\nTemperature in Fahrenheit is: "<<temp<<endl;**

**temp;**

**convert(&temp);**

**cout<<"\nTemperature in Celsius is: "<<temp<<endl;**

**return 0;**

**}**

**void convert(int \*p)**

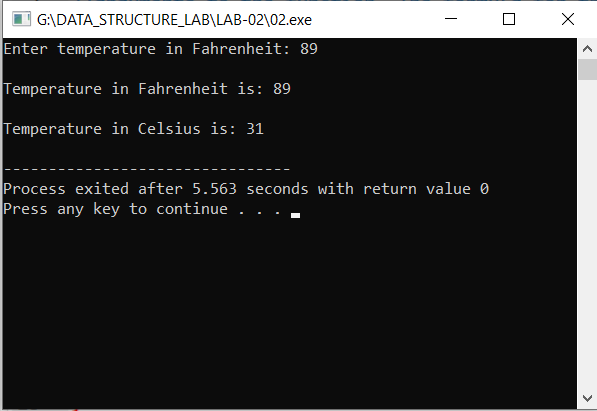
**{**

**\*p = ((\*p-32)\*5)/9;**

**// return \*p;**

**}**

**Output:**

****

**Task # 03:**

Create a structure(student) is which should contain name, roll and marks as its data member. Then, create a structure variable(s). Then take data (name, roll and marks) from user and store it in data members of structure variables. Display the data Entered by the user

**Code:**

**#include<iostream>**

**#include<cstdlib>**

**using namespace std;**

**struct Student**

**{**

**string name;**

**int marks;**

**double Roll\_no;**

**};**

**int main()**

**{**

**int size;**

**cout<<"Enter the number of students you want to put data: ";**

**cin>>size;**

**Student S[size];**

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

**{**

**cout<<"\nEnter information for Student 0"<<i+1<<endl;**

**cin.ignore();**

**cout<<"\nEnter the name of the student: ";**

**getline(cin,S[i].name);**

**cout<<"\nEnter the marks of the student: ";**

**cin>>S[i].marks;**

**cout<<"\nEnter the Roll\_no of the student: ";**

**cin>>S[i].Roll\_no;**

**}**

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

**{**

**cout<<"\nInformation for Student 0"<<i+1<<endl;**

**cout<<"\nName of the student: "<<S[i].name;**

**cout<<"\nMarks of the student: "<<S[i].marks;**

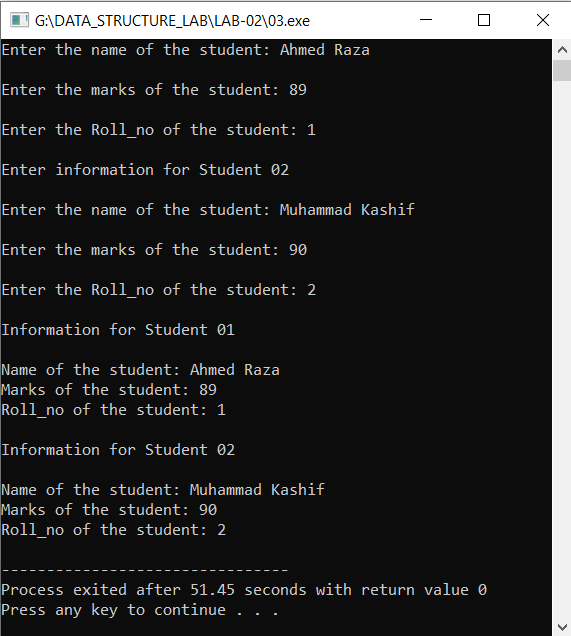
**cout<<"\nRoll\_no of the student: "<<S[i].Roll\_no<<endl;**

**}**

**return 0;**

**}**

**Output:**

****

**Results & Observations:**

In this Lab I've learned about the concept of arrays with pointers in the function & also understand that how array can be passed to a function with the help of pointers & returned to a main function. Further I understand the concept of Structures like: Structure Declaration, Structure Definition, Structure Variables, Structure Membership, Arrays of structures. And I understand the concept of classes and its objects. In the first task, I've used Swap function in which I've passed two pointers that'll swap the values of a and b. In the second task, I've used convert function in which I've passed pointer address. In the Third task, I've used Structure of Student to enter the details of student by using structure.