Department of IT and Computer Science
Pak-Austria Fachhochschule: Institute of Applied Sciences
and Technology, Haripur, Pakistan

COMP-201L Data Structures and Algorithms Lab

Lab Report: 02

Class: Computer Science

Name: Yaseen Ejaz Ahmed

Registration No.: B20F0283CS014

Semester: Third

Submission Date:

Submitted to: Dr. Rafi Ullah

_____

Instructor Signature

Lab No.2

Structures & Pointers Using C++

Objectives:

- To Revise the concepts of structures & pointers using C++.
- To Implement structures and pointers in C++

Tools/Software Required:

Dev C++ Compiler

Introduction:

Structure is a collection of variables of different data types under a single name. It is similar to a class in that, both holds a collection of data of different data types. We use pointers to point to different addresses and to access the values stored in them.

Lab Tasks:

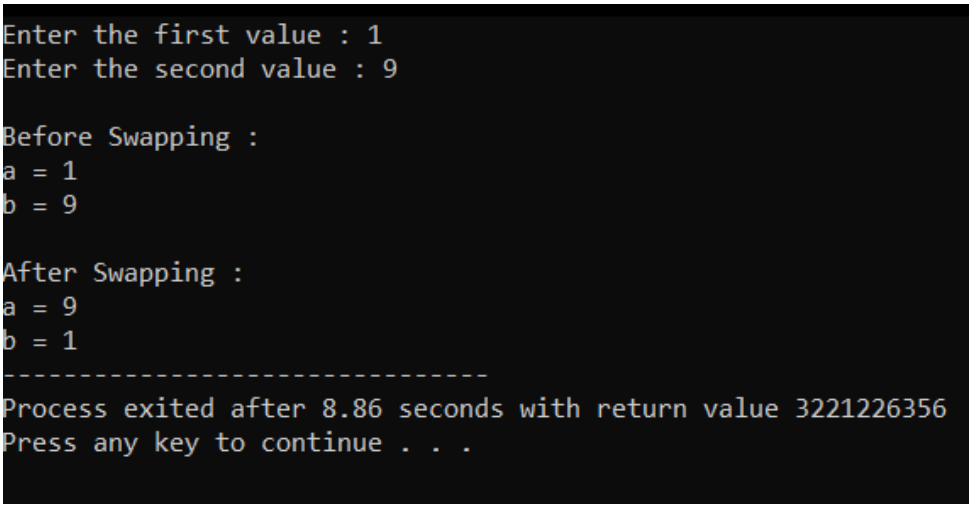
Lab Task 01: Write a program to swap two values by passing pointers as argument to the function.

Code:

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

SwapValues(int *p,int *q)
{
	int *temp;

	temp=p;
	p=q;
	q=temp;

	cout<<"\n\nAfter Swapping :\na = "<<*p<<"\t"<<"\nb = "<<*q;
	delete p;
	delete q;
	p=q=NULL;
}

int main()
{
	int a,b;

	cout<<"Enter the first value : ";
	cin>>a;

	cout<<"Enter the second value : ";
	cin>>b;

	int *aptr=&a;
	int *bptr=&b;

	cout<<"\nBefore Swapping :\na = "<<*aptr<<"\t"<<"\nb = "<<*bptr;

	SwapValues(aptr,bptr);
```

```
    delete aptr;
    delete bptr;
    aptr=bptr=NULL;
}
```

Output:



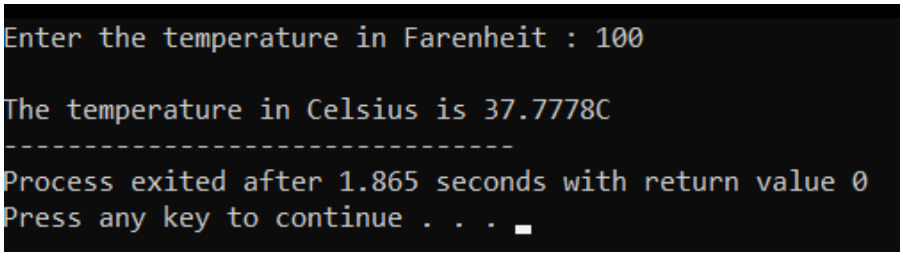
Lab Task 02: Write a program to convert Fahrenheit temperature to Celsius degrees by passing pointers as arguments to the function. The formula for the conversion is:
c = (f-32) *5 .0/9.0.

Code:

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

float ConvertToCelsius(float *fp)
{
    float c;
    float *cp=&c;

    c = (5.0/9.0) * (*fp-32);

    return *cp;
}
```

```
int main()
{
	float f;

	cout<<"Enter the temperature in Farenheit : ";
	cin>>f;

	float *fp=&f;

	cout<<"\nThe temperature in Celsius is "<<ConvertToCelsius(fp)<<"C";

	delete fp;
	fp=NULL;

}
```

Output:



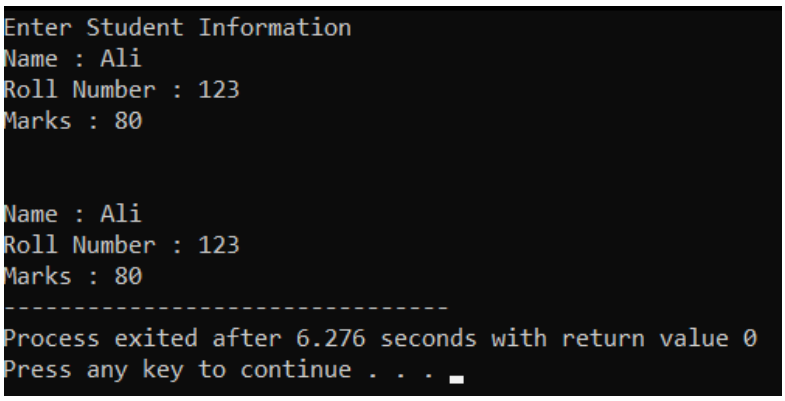
Lab Task 03: Create a structure(student) is which should contain name, roll and marks as its data member. Then, create a structure variable(s). Then take data (name, roll and marks) from user and store it in data members of structure variables. Display the data Entered by the user.

Code:

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

struct student
{
	string Name;
	string RollNo;
	float Marks;

};
```

```
int main()
{
	student S;

	cout<<"Enter Student Information\nName : ";
	cin>>S.Name;

	cout<<"Roll Number : ";
	cin>>S.RollNo;

	cout<<"Marks : ";
	cin>>S.Marks;

	cout<<"\n\nName :"<<S.Name<<"\nRoll Number : "<<S.RollNo<<"\nMarks :
"<<S.Marks;
}
```

Output:



Lab Task 04: Declare a structure to represent a complex number (a number having a real part and imaginary part). Write C++ functions to add, subtract, multiply and divide two complex numbers. (Comment each line with information about it)

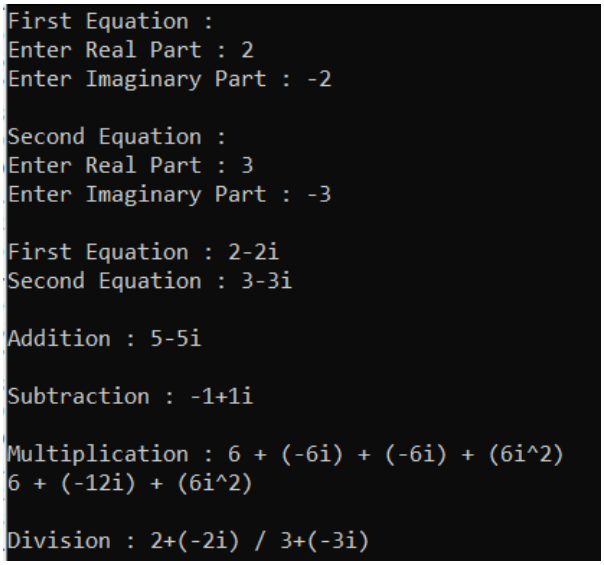
Code:

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

struct complex
{
	double real1;
	double img1;

	double real2;
	double img2;
}c;

void Add()
{
	cout<<"\n\nAddition : ";
	if(c.img1+c.img2>=0)
	cout<<c.real1+c.real2<<"+"<<c.img1+c.img2<<"i";
	else
	cout<<c.real1+c.real2<<c.img1+c.img2<<"i";
}

void Sub()
{
	cout<<"\n\nSubtraction : ";
	if(c.img1+c.img2>=0)
	cout<<c.real1-c.real2<<c.img1-c.img2<<"i";
	else
	cout<<c.real1-c.real2<<"+"<<c.img1-c.img2<<"i";
}

void Mul()
{
	cout<<"\n\nMultiplication : ";
	cout<<c.real1*c.real2<<" + ("<<c.real1*c.img2<<"i) + ("<<c.img1*c.real2<<"i) +
("<<c.img1*c.img2<<"i^2)\n";
```

```
        cout<<c.real1*c.real2<<" + ("<<(c.real1*c.img2)+ (c.img1*c.real2)<<"i) +
("<<c.img1*c.img2<<"i^2)";
}

void Div()
{
        cout<<"\n\nDivision : ";
        cout<<c.real1<<"+("<<c.img1<<"i) / "<<c.real2<<"+("<<c.img2<<"i)";
}



int main()
{
        cout<<"First Equation :\nEnter Real Part : ";
        cin>>c.real1;

        cout<<"Enter Imaginary Part : ";
        cin>>c.img1;

        cout<<"\nSecond Equation :\nEnter Real Part : ";
        cin>>c.real2;

        cout<<"Enter Imaginary Part : ";
        cin>>c.img2;


        cout<<"\nFirst Equation : ";
        if(c.img1>=0)
        cout<<c.real1<<"+"<<c.img1<<"i";
        else
        cout<<c.real1<<c.img1<<"i";

        cout<<"\nSecond Equation : ";
        if(c.img2>=0)
        cout<<c.real2<<"+"<<c.img2<<"i";
        else
        cout<<c.real2<<c.img2<<"i";

        Add();
        Sub();
        Mul();
```

```
    Div();
}
```

Output:



Lab Task 05: Construct a structure to compute time difference of two time periods. The time periods should be entered by the user. (Comment each line with information about it)
Hint* use functions and pointers to struct the structure.

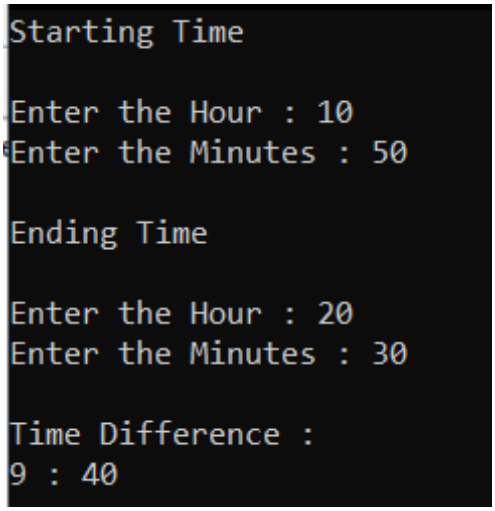
Code:

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

struct Time
{
    int SHour;
    int FHour;
    int SMin;
    int FMin;
};

int main()
{
    Time t;
    int hour, minute;

    cout<<"Starting Time\n\n";
```

```
        do
        {
                cout<<"Enter the Hour : ";
           cin>>t.SHour;
        }
        while(t.SHour<0 || t.SHour > 23);

        do
        {
      cout<<"Enter the Minutes : ";
           cin>>t.SMin;
}
while(t.SMin<0 || t.SMin > 59);

        cout<<"\nEnding Time\n\n";

        do
        {
                cout<<"Enter the Hour : ";
           cin>>t.FHour;
        }
        while(t.FHour<0 || t.FHour > 23);

        do
        {
                cout<<"Enter the Minutes : ";
           cin>>t.FMin;
        }
        while(t.FMin<0 || t.FMin > 59);


        minute = t.FMin - t.SMin;
        hour = t.FHour - t.SHour;

        if(minute<0)
        {
                minute = minute + 60;
                hour--;
        }
```

```
        if(hour<0)
        {
                hour = hour + 24;
        }

        cout<<"\nTime Difference : \n"<<hour<<" : "<<minute;
}
```

Output:



Results & Observations:

In this lab, we have learnt the basics of structures and pointers. We can use structures for holding many types of data with a single handle and we can also use reusability of code in these cases.