**STRUCTURES**

**LAB # 11**



**Spring 2019**

**CSE102L Computer Programming Lab**

Submitted by: **SHAH RAZA**

Registration No. : **18PWCSE1658**

Class Section: **B**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submitted to:

**Engr. Madiha Sher**

June 21, 2019

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

## Objectives:

To learn how to declare a C++ record (structure) data type.

By the end of this activity, students will be able to apply the concept of structures to real world problems and make their own data types.

**TASK #1**

A phone number such is (091) 767-8900 can be thought of as having three parts: the area code (091), exchange (767) and number 8900. Write a program that uses structure to store these 3 parts of a phone number separately. Assume name of the structure is ‘phone’. Create two structures of type ‘phone’, initialize one structure and take the other one from the user. Display both numbers.

**Code:**

#include <iostream>

using namespace std;

struct Phone

{

int area;

int exchange;

int number;

};

void input(Phone \*p)

{

cout<<"Area No.: ";

cin>>p->area;

cout<<"Exchange: ";

cin>>p->exchange;

cout<<"Number: ";

cin>>p->number;

}

void display (Phone p)

{

cout<<"(0"<<p.area<<")"<<" "<<p.exchange<<"-"<<p.number<<endl;

}

int main()

{

Phone p1={340,150,2575},p2;

input(&p2);

cout<<"My number is: ";

display(p1);

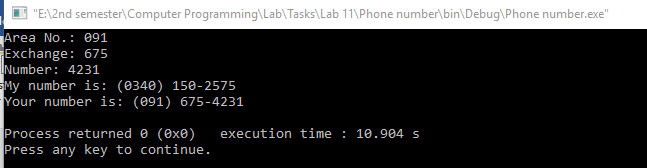
cout<<"Your number is: ";

display(p2);

return 0;

}

**Output (Compilation, testing and debugging):**



**TASK #2**

A point on a two dimensional space has two numbers: an ‘X’ coordinate and a ‘Y’ coordinate such as (4, 5). Write a program that uses a structure called ‘point’ to model a point and draw a rectangle of ‘\*’ based on the entered point and height and width of the rectangle.

**Code:**

#include <iostream>

using namespace std;

struct Phone

{

int area;

int exchange;

int number;

};

void input(Phone \*p)

{

cout<<"Area No.: ";

cin>>p->area;

cout<<"Exchange: ";

cin>>p->exchange;

cout<<"Number: ";

cin>>p->number;

}

void display (Phone p)

{

cout<<"(0"<<p.area<<")"<<" "<<p.exchange<<"-"<<p.number<<endl;

}

int main()

{

Phone p1={340,150,2575},p2;

input(&p2);

cout<<"My number is: ";

display(p1);

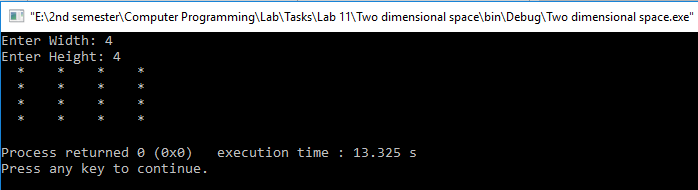
cout<<"Your number is: ";

display(p2);

return 0;

}

**Output (Compilation, testing and debugging):**



**TASK #3**

Write a program that uses a structure named ‘complex’ and performs all mathematical operations such as sum, subtract, multiply and divide on complex numbers.

**Code:**

#include <iostream>

using namespace std;

struct Complex

{

int real;

int imaginary;

};

void input(Complex \*a)

{

cout<<"Enter Real part: ";

cin>>a->real;

cout<<"Enter Imaginary part: ";

cin>>a->imaginary;

}

Complex product(Complex a,Complex b)

{

Complex prod;

prod.real=(a.real\*b.real)+(a.imaginary\*b.imaginary)\*(-1);

prod.imaginary=(a.real\*b.imaginary)+(a.imaginary\*b.real);

return prod;

}

Complex sum (Complex a,Complex b)

{

Complex s;

s.real=a.real+b.real;

s.imaginary=a.imaginary+b.imaginary;

return s;

}

Complex sub(Complex a, Complex b)

{

Complex s;

s.real=a.real-b.real;

s.imaginary=a.imaginary-b.imaginary;

return s;

}

Complex Div(Complex a, Complex b)

{

Complex d,nom,temp;

int denom;

temp.real=b.real;

temp.imaginary=(-1)\*b.imaginary;

nom=product(a,temp);

denom=(a.real\*b.real)-(a.imaginary\*b.imaginary)\*(-1);

d.real=nom.real/denom;

d.imaginary=nom.imaginary/denom;

return d;

}

int main()

{

Complex a,b,A,S,P,D;

input(&a);

input(&b);

A=sum(a,b);

S=sub(a,b);

P=product(a,b);

D=Div(a,b);

cout<<"Addition: "<<A.real<<" + "<<A.imaginary<<"i"<<endl;

cout<<"Subtraction: "<<S.real<<" + "<<S.imaginary<<"i"<<endl;

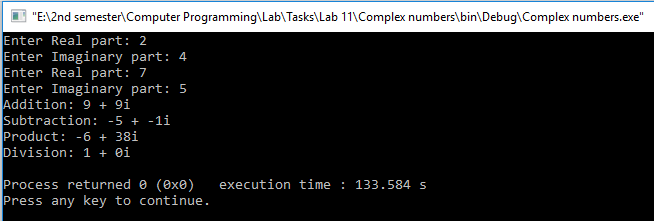
cout<<"Product: "<<P.real<<" + "<<P.imaginary<<"i"<<endl;

cout<<"Division: "<<D.real<<" + "<<D.imaginary<<"i"<<endl;

return 0;

}

**Output (Compilation, testing and debugging):**



**TASK #4**

Write a program to develop student information system. This system should be able to take the student’s credentials, like name, reg no and semester no, as input and store them in a structure variable Student\_t. The system should be capable of storing the information of multiple students. And it should be capable of displaying the student’s information, when asked to do so, via reg no.

**Code:**

#include <iostream>

#include <stdio.h>

using namespace std;

struct student

{

int reg;

char name[30];

int semester;

}Student\_t[120];

int strlen(char \*A) //strlen Function Definition

{

int i=0;

for(;A[i]!='\0';i++);

return i;

}

void input(student \*s)

{

cout<<"Reg no: ";

cin>>s->reg;

cout<<"Name: ";

cin>>s->name;

cout<<"Semester: ";

cin>>s->semester;

}

void display(int reg)

{

for (int i=0;i<120;i++)

{

if(Student\_t[i].reg==reg)

{

cout<<"Name: "<<Student\_t[i].name<<endl;

cout<<"Semester: "<<Student\_t[i].semester;

}

}

}

int main()

{

for(int i=0;i<3;i++)

{

input(&Student\_t[i]);

}

int reg;

cout<<"Enter the registration Number: ";

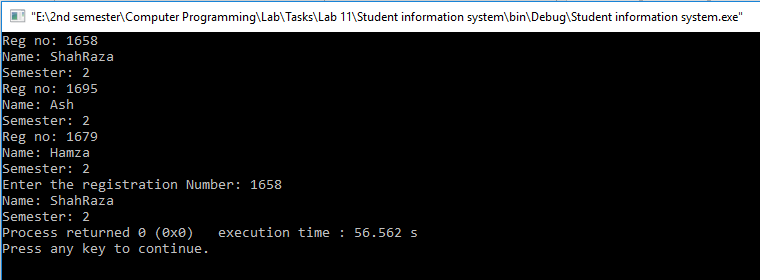
cin>>reg;

display(reg);

return 0;

}

**Output (Compilation, testing and debugging):**



**TASK #5**

Write a program to model a structure for date and performs mathematical operations such as add and subtract.

**Code:**

#include <iostream>

using namespace std;

struct Date

{

int year;

int month;

int day;

}d1,d2;

void input(Date \*d)

{

cout<<"Enter Year: ";

cin>>d->year;

cout<<"Enter Month: ";

cin>>d->month;

cout<<"Enter Day: ";

cin>>d->day;

}

Date add(Date d1, Date d2)

{

Date s;

s.day=d1.day+d2.day;

s.month=d1.month+d2.month;

s.year=d1.year+d2.year;

if(s.day>30)

{

s.month++;

s.day-=30;

}

if(s.month>12)

{

s.year++;

s.month-=12;

}

return s;

}

Date sub(Date d1, Date d2)

{

Date c;

c.day=d1.day-d2.day;

c.month=d1.month-d2.month;

c.year=d1.year-d2.year;

if(c.day<1)

{

c.month--;

c.day+=30;

}

if(c.month<1)

{

c.year--;

c.month+=12;

}

return c;

}

void display(Date d)

{

cout<<"The new date is: \n";

cout<<d.year<<" : "<<d.month<<" : "<<d.day;

}

int main()

{

cout<<"Enter the first date\n";

input(&d1);

cout<<"Enter the second date\n";

input(&d2);

Date A=add(d1,d2);

Date S=sub(d1,d2);

cout<<"After Addition: \n";

display(A);

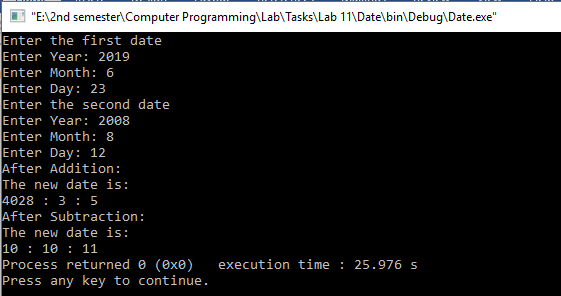
cout<<"\nAfter Subtraction: \n";

display(S);

return 0;

}

**Output (Compilation, testing and debugging):**



**TASK #6**

Write a program to model a structure for time and performs mathematical operations such as add and subtract.

**Code:**

#include <iostream>

using namespace std;

struct time

{

int hours;

int mins;

int sec;

}t1,t2;

void input(time \*t)

{

cout<<"Hours: ";

cin>>t->hours;

cout<<"Mins: ";

cin>>t->mins;

cout<<"Seconds: ";

cin>>t->sec;

}

time add(time t1,time t2)

{

time sum;

sum.sec=t1.sec+t2.sec;

sum.mins=t1.mins+t2.mins;

sum.hours=t1.hours+t2.hours;

if(sum.sec>=60)

{

sum.mins++;

sum.sec-=60;

}

if(sum.mins>=60)

{

sum.hours++;

sum.mins-=60;

}

if (sum.hours>=24)

sum.hours-=24;

return sum;

}

time subtract(time t1,time t2)

{

time s;

s.sec=t1.sec-t2.sec;

s.mins=t1.mins-t2.mins;

s.hours=t1.hours-t2.hours;

if(s.sec<0)

{

s.mins--;

s.sec+=60;

}

if(s.mins<0)

{

s.hours--;

s.mins+=60;

}

if (s.hours<0)

s.hours+=24;

return s;

}

void display(time a)

{

cout<<a.hours<<" : "<<a.mins<<" : "<<a.sec;

}

int main()

{

input(&t1);

input(&t2);

time s=add(t1,t2);

cout<<"Sum = ";

display(s);

s=subtract(t1,t2);

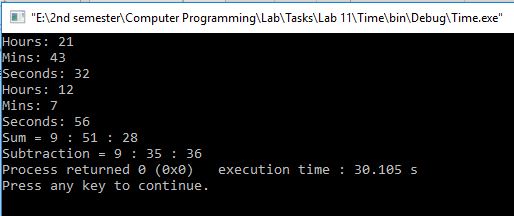
cout<<"\nSubtraction = ";

display(s);

return 0;

}

**Output (Compilation, testing and debugging):**



**TASK #7**

C++ program to create a class to read and add two distance. Performs mathematical operations such as add and subtract.Use the following structure:

class Distance

{

public:

int feet;

int inch;

}

**Code:**

#include <iostream>

using namespace std;

class Distance

{

public:

int feet;

int inch;

}d1,d2;

void input(Distance \*d)

{

cout<<"Feet: ";

cin>>d->feet;

cout<<"Inch: ";

cin>>d->inch;

}

Distance add(Distance a,Distance b)

{

Distance s;

s.inch=a.inch+b.inch;

s.feet=a.feet+b.feet;

if(s.inch>=12)

{

s.feet++;

s.inch-=12;

}

return s;

}

Distance sub(Distance a,Distance b)

{

Distance c;

c.inch=a.inch-b.inch;

c.feet=a.feet-b.feet;

if(c.inch<0)

{

c.feet--;

c.inch+=12;

}

return c;

}

void display(Distance a)

{

cout<<a.feet<<" Feet "<<a.inch<<" in.";

}

int main()

{

input(&d1);

input(&d2);

Distance s=add(d1,d2);

cout<<"Addition = ";

display(s);

s=sub(d1,d2);

cout<<"\nSubtraction = ";

display(s);

return 0;

}

**Output (Compilation, testing and debugging):**

