OBJECT ORIENTED PROGRAMMING LAB

ASSIGNMENT #1

NAME: AHTISHAM KHAN
SECTION: BSE-2B
ENROLLMENT: 09-131242-014

ASSIGNMENT#1

OBJECT ORIENTED PROGRAMMING LAB

STRUCTURES:

A structure is a collection of related elements or data items. A structure is a collection of simple variables. The data items in a structure are called the “members” of the structure

OBJECTIVES:

- Structure declarations definitions
- Accessing structure members
- Nested structures
- Structures as objects and data types

LAB TASKS

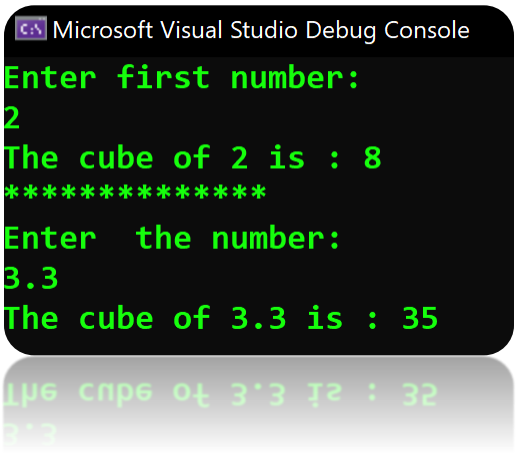
TASK#1:

Write a program to implement the cube root with the help of function overloading.

CODE:

```
#include<iostream>
using namespace std;
int cube(int);
int cube(float);
int main() {
	int a;
	float b;
	cout << "Enter first number: " << endl;
	cin >> a;
	cout << "The cube of " << a << " is : " << cube(a) << endl;
	cout << "**************" << endl;
	cout << "Enter  the number: " << endl;
	cin >> b;
	cout << "The cube of " << b << " is : " << cube(b) << endl;
}
int cube(int a) {
	return a * a * a;
}
int cube(float b) {
	return b * b * b;
}
```

OUTPUT:



TASK#2:

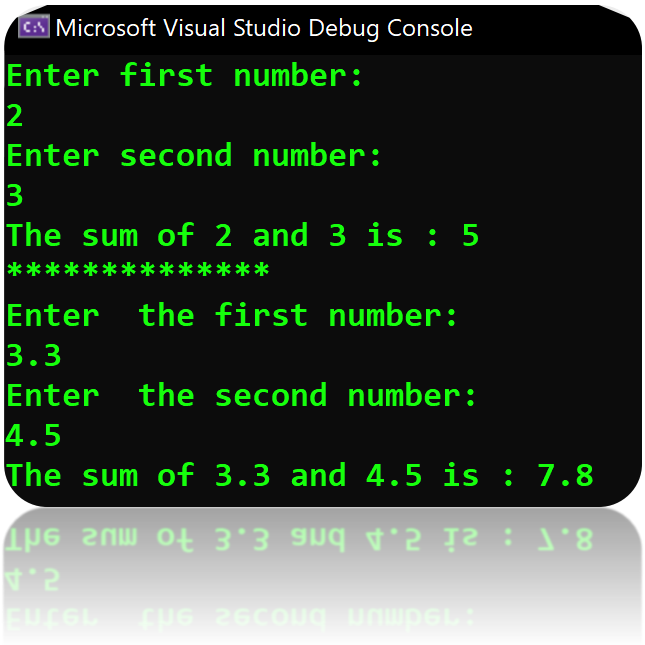
Write a program that uses function overloading for adding the two given integer and double precision values separately.

CODE:

```
#include<iostream>

using namespace std;

int add(int,int);

double add(double,double);

int main() {

        int a,b;

        double x,y;

        cout << "Enter first number: " << endl;

        cin >> a;
```

```
        cout << "Enter second number: " << endl;
        cin >> b;
        cout << "The sum of " << a <<" and "<<b << " is : " << add(a,b) << endl;
        cout << "***************" << endl;
        cout << "Enter  the first number: " << endl;
        cin >> x;
        cout << "Enter  the second number: " << endl;
        cin >> y;
        cout << "The sum of "<<x<< " and " << y << " is : " << add(x,y) << endl;
}
int add(int a,int b) {
        return a + b; ;
}
double add(double w,double z) {
        return w+z;
}
```

OUTPUT:



TASK#3:

Create a structure called time. Its three members, all type int, should be called hours, minutes, and seconds. Write a program that prompts the user to enter a time value in hours, minutes, seconds. This can be in 12:59:59 format, or each number can be entered at a separate prompt (“Enter hours:”, and so forth). The program should then store the time in a variable of type struct time, and finally print out the total number of seconds represented by this time value:

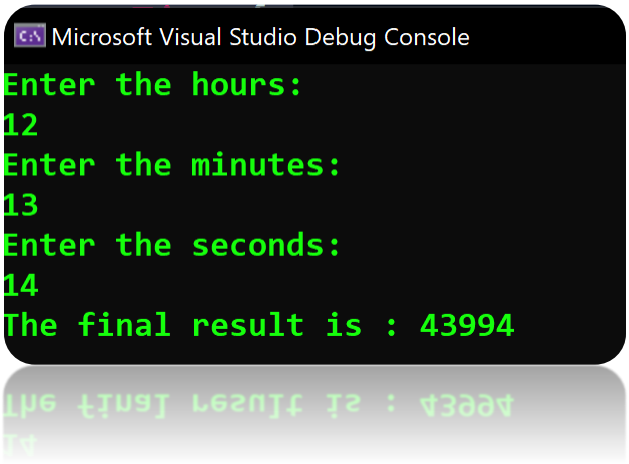
long totalsecs = t1.hours*3600 + t1.minutes*60 + t1.seconds

CODE:

```
#include<iostream>

using namespace std;

struct Time {
```

```
        int hour;
        int minutes;
        int second;
        long totalsec;
};
int main() {
        Time t1;
        cout << "Enter the hours: " << endl;
        cin >> t1.hour;
        cout << "Enter the minutes: " << endl;
        cin >> t1.minutes;
        cout << "Enter the seconds: " << endl;
        cin >> t1.second;
        if (t1.hour < 24 || t1.minutes < 60 || t1.second < 60) {
                t1.totalsec = t1.hour * 3600 + t1.minutes * 60 + t1.second;
                cout << "The final result is : " << t1.totalsec << endl;
        }
        else {
                cout << "You enter trhe wrong input" << endl;
        }
        return 0;
}
```

OUTPUT:



TASK#4:

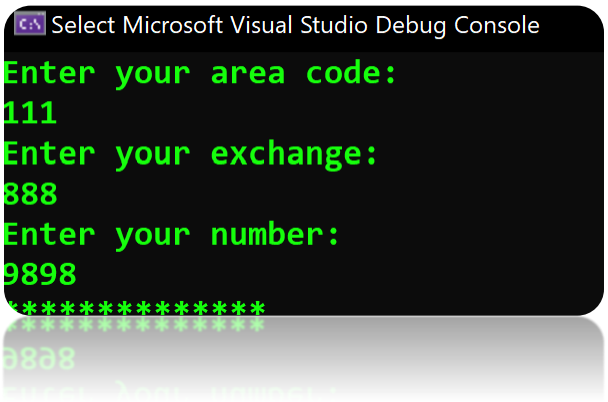
A phone number, such as (212) 767-8900, can be thought of as having three parts: the area code (212), the exchange (767), and the number (8900). Write a program that uses a structure to store these three parts of a phone number separately. Call the structure phone. Create two structure variables of type phone. Initialize one, and have the user input a number for the other one. Then display both numbers. The interchange might look like this:

CODE:

```
#include<iostream>

using namespace std;

struct phone {

        int area_code;

        int exchange;

        int number;

};

int main() {

        phone Pmy;
```

```
	Pmy.area_code = 222;
	Pmy.exchange = 123;
	Pmy.number = 4242;
	phone Pyour;
	cout << "Enter your area code: " << endl;
	cin >>Pyour.area_code;
	cout << "Enter your exchange: " << endl;
	cin >> Pyour.exchange;
	cout << "Enter your number: " << endl;
	cin >> Pyour.number;
	cout << "***************" << endl;
	if (Pyour.area_code <300 || Pyour.exchange <300 || Pyour.number <300) {
		cout << "My phone number is : (" << Pmy.area_code << ")" <<
Pmy.exchange << "-" << Pmy.number << endl;
		cout << "Your phone number is : (" << Pyour.area_code << ")" <<
Pyour.exchange << "-" << Pyour.number << endl;
	}
	else {
		cout <<"Your enter invlid value." << endl;
	}
	return 0;
}
```

OUTPUT:


Select Microsoft Visual Studio Debug Console
Enter your area code:
111
Enter your exchange:
888
Enter your number:
9898
****