Object Oriented Lab

Spring2025

Assignment #7

Umar Farooq
09-131242-088
BSE-2B

DEPARTMENT OF SOFTWARE ENGINEERING
BAHRIA UNIVERSITY ISLAMABAD CAMPUS

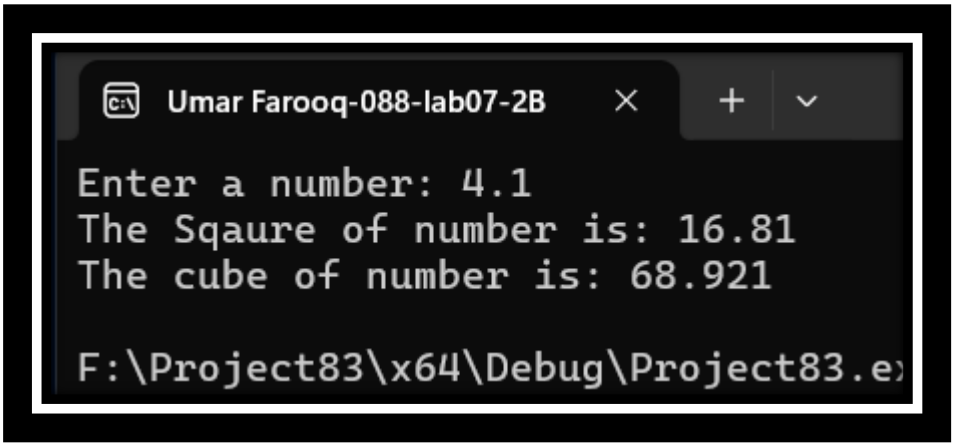
Task 01: Write a C++ program that demonstrates single inheritance by computing the square and cube of a given number. Create a base class Square containing a computeSquare() function to calculate the square of a number. Derive a class Cube from Square that implements a computeCube() function, which should utilize the base class's computeSquare() method to calculate the cube (by multiplying the square result with the original number).

Code:

```
#include <iostream>
using namespace std;
class Square {
protected:
	float num;
public:
	void setValue(float n)
	{
		num = n;
	}
	float computeSquare();
};
float Square::computeSquare()
{
	float result;
	result = num * num;
	cout << "The Sqaure of number is: " << result << endl;
	return result;
}
class Cube : public Square {
public:
	float computeCube();
};
float Cube::computeCube()
{
	float result;
	result = num * num * num;
	cout << "The cube of number is: " << result << endl;
	return result;
}
int main()
```

```
{
        Cube c1;
        float n;
        cout << "Enter a number: ";
        cin >> n;
        c1.setValue(n);
        c1.computeSquare();
        c1.computeCube();
        return 0;
}
```

Output:



Task 02: Create a base class Student that stores name, roll number and address (type string) of the student. Student class must have getdata () function to get its data (name, roll no and address of the student) from the user.

From this class derive another class Marks. This class must have inputmarks() function to get its data (marks in three subjects) from the user. Then calculate the total marks and average marks in the three subjects. Create another function show_detail() to display the marks in three subjects, total marks and average marks.

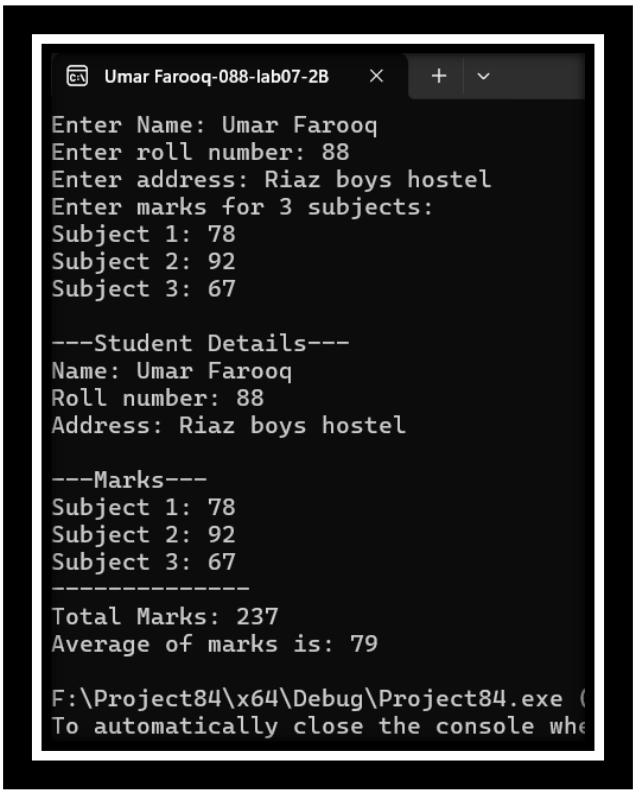
Code:

```
#include <iostream>
#include <string>
using namespace std;
class Student {
protected:
        string name;
        int rollno;
        string address;
```

```
public:
        void getData();
};
void Student::getData()
{
        cout << "Enter Name: ";
        getline(cin, name);
        cout << "Enter roll number: ";
        cin >> rollno;
        cin.ignore();
        cout << "Enter address: ";
        getline(cin, address);
}
class Marks : public Student {
protected:
        float marks[3];
        float avg, result;
public:
        void inputMarks();
        void showDetails();
};
void Marks::inputMarks()
{
        cout << "Enter marks for 3 subjects: " << endl;
        for (int i = 0; i < 3; i++)
        {
                cout << "Subject " << i + 1 << ": ";
                cin >> marks[i];
        }
        result = marks[0] + marks[1] + marks[2];
        avg = result / 3.0;
}
void Marks::showDetails()
{
        cout << endl;
        cout << "---Student Details---" << endl;
        cout << "Name: " << name << endl;
        cout << "Roll number: " << rollno << endl;
```

```
        cout << "Address: " << address << endl;
        cout << endl;
        cout << "---Marks---" << endl;
        for (int i = 0; i < 3; i++)
        {
                cout << "Subject " << i + 1 << ": ";
                cout << marks[i] << endl;
        }
        cout << "--------------" << endl;
        cout << "Total Marks: " << result << endl;
        cout << "Average of marks is: " << avg << endl;
}
int main()
{
        Marks m1;
        m1.getData();
        m1.inputMarks();
        m1.showDetails();
        return 0;
}
```

Output:



Task 03: Create a base class Employee that stores name (a string) and identification number (type integer) of an employee. From this class derive three more classes:
Manager: that stores the title and salary (string) of the manager.
Scientist: that stores the number of articles they have published (type integer) and salary (a string) of the scientist.
Clerks: that stores the overtime (type int) of the clerk.
Each of the four classes should have an input () function to input data from the user, and show () function to display data. Write a main function to test the three classes by creating instances of them, asking the user to fill in data with input (), and then displaying data with show () function.

Code:

```
#include <iostream>
#include <string>
using namespace std;
class Employee {
protected:
   string name;
   int id;
public:
   void input();
   void show();
};
void Employee::input()
{

        cout << "Enter employee name: ";
        getline(cin, name);
        cout << "Enter the identification number of the employee: ";
        cin >> id;
   cin.ignore();
}
void Employee::show()
{
        cout << "Name: " << name << endl;
        cout << "ID Number: " << id << endl;
}
class Manager : public Employee {
private:
```

```
   string title;
   int salary;
public:
   void input();
   void show();
};
void Manager::input()
{

        cout << "Enter employee name: ";
        getline(cin, name);
        cout << "Enter the identification number of the employee: ";
        cin >> id;
   cin.ignore();
        cout << "Enter the title: ";
        getline(cin, title);
   cin.ignore();
        cout << "Enter the salary: $";
        cin >> salary;
}
void Manager::show()
{
   Employee::show();
   cout << "Title: " << title << endl;
   cout << "Salary: $" << salary << endl;
}
class Scientist : public Employee {
private:
   int articles;
   int salary;
public:
   void input();
   void show();
};
void Scientist::input()
{
   Employee::input();
   cout << "Enter number of articles published: ";
```

```
   cin >> articles;
   cout << "Enter the salary: $";
   cin >> salary;
}
void Scientist::show()
{
        Employee::show();
        cout << "Number of Articles Published: " << articles << endl;
        cout << "Salary: $" << salary << endl;
}
class Clerks : public Employee {
private:
   int overtime;
public:
   void input();
   void show();
};
void Clerks::input()
{
        Employee::input();
        cout << "Enter the number of overtime hours worked: ";
        cin >> overtime;
}
void Clerks::show()
{
        Employee::show();
        cout << "Overtime Hours: " << overtime << endl;
}
int main() {
   Manager m;
   Scientist s;
   Clerks c;

   cout << "---Enter details for Manager---" << endl;
   m.input();
   cin.ignore();
   cout << "---Enter details for Scientist---" << endl;
   s.input();
```

```
    cin.ignore();
    cout << "---Enter details for Clerk---" << endl;
    c.input();
    cin.ignore();

    cout << endl;
    cout << "---Displaying Manager Details---" << endl;
    m.show();
    cout << endl;
    cout << "---Displaying Scientist Details---" << endl;
    s.show();
        cout << endl;
    cout << "---Displaying Clerk Details---" << endl;
    c.show();

    return 0;
```

Output: