Object Oriented Programming

Lab Report

Lab08

Group Members Name & Reg #:	**Muhammad Haris Irfan** **(FA18-BCE-090)**
Class	Object Oriented Programming CSC241 (**BCE-4B**)
Instructor's Name	Maam Amber Madeeha Zeb

In Lab Tasks

5.1 Question 1:

Create a Class named base which has two data members. Then derive a class derived1 from base class which has one data members. Derive a class derived2 from derived1.

i. **Write functions for each class to get and display values.**
ii. **Write main() function to create object of derived2 and through that object access the data member of base class and derived1 class**

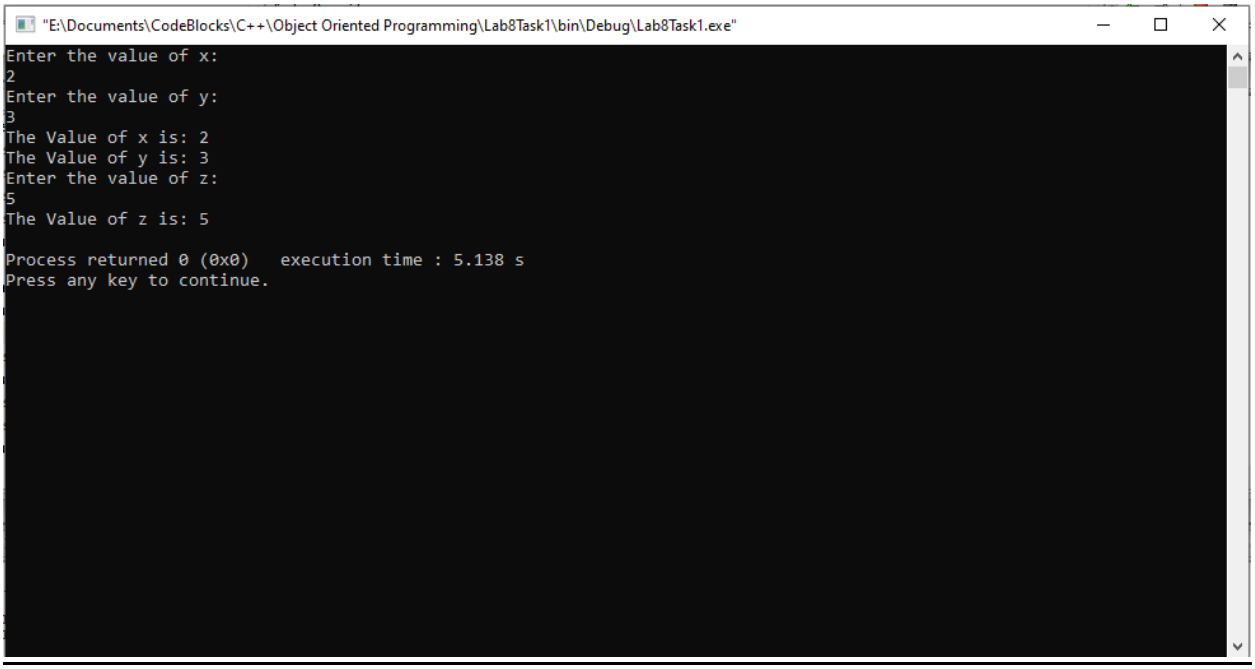
Solution:

The code is given below,

```
#include <iostream>

using namespace std;

class base
{
protected:
    int x,y;

public:
   void get()
    {
        cout<<"Enter the value of x:"<<endl;
        cin>>x;
        cout<<"Enter the value of y:"<<endl;
        cin>>y;

    }

    void display()
    {
        cout<<"The Value of x is: "<<x<<endl;
        cout<<"The Value of y is: "<<y<<endl;
    }


};
class derived1:public base
{
protected:
    int z;
public:
    void getD()
    {
        cout<<"Enter the value of z:"<<endl;
        cin>>z;
    }

    void displayD()
```

```
40      {
41          cout<<"The Value of z is: "<<z<<endl;
42      }
43  };
44  class derived2:public derived1
45  {
46
47  };
48
49
50
51
52
53  int main()
54  {
55  derived2 ob;
56
57  ob.get();
58  ob.display();
59
60  ob.getD();
61  ob.displayD();
62
63      return 0;
64  }
```

Console Output is shown below.



5.2 Question 2:

Create a class Person having name, age and gender as its data members. Create another class Employee which has employername and dailywages as it data member. From these two classes derive another class teacher which contains teacher grade as data member.

i. **Write set and get functions to enter and display the data members.**
ii. **Write main function to implement these classes. Enter the teacher data to show multiple inheritance.**

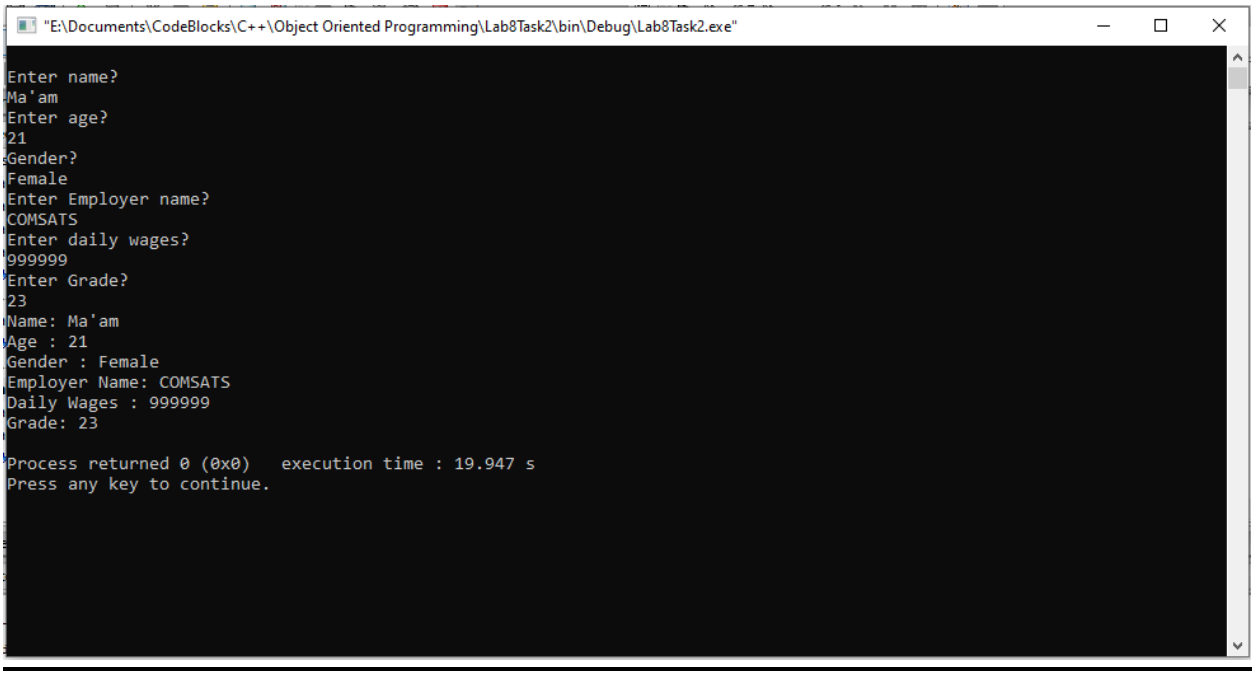
Solution:

The code is given below,

```
#include <iostream>

using namespace std;

class person
{
protected:
    string name;
    string Gender;
    int age;
public:
    void get()
    {
        cout<<endl;
        cout<<"Enter name?"<<endl;
        cin>>name;
        cout<<"Enter age?"<<endl;
        cin>>age;
        cout<<"Gender?"<<endl;
        cin>>Gender;

    }

    void display()
    {
        cout<<"Name: "<<name<<endl;
        cout<<"Age : "<<age<<endl;
        cout<<"Gender : "<<Gender<<endl;
    }
};

class employee
{
protected:
    string employername;
    float dailywages;
public:
    void getE()
    {
        cout<<"Enter Employer name?"<<endl;
        cin>>employername;
        cout<<"Enter daily wages?"<<endl;
        cin>>dailywages;


    }
```

```
47
48      void displayE()
49      {
50          cout<<"Employer Name: "<<employername<<endl;
51          cout<<"Daily Wages : "<<dailywages<<endl;
52
53      }
54  };
55
56  class teacher: public person,public employee
57  {
58  protected:
59      int grade;
60  public:
61      void getT()
62      {
63          cout<<"Enter Grade?"<<endl;
64          cin>>grade;
65
66      }
67
68      void displayT()
69      {
70          cout<<"Grade: "<<grade<<endl;
71
72      }
73
74  };
75
76  int main()
77  {
78      teacher t1;
79
80      t1.get();
81      t1.getE();
82      t1.getT();
83
84      t1.display();
85      t1.displayE();
86      t1.displayT();
87
88
89      return 0;
90  }
```

Console Output is shown below.


"E:\Documents\CodeBlocks\C++\Object Oriented Programming\Lab8Task2\bin\Debug\Lab8Task2.exe"
Enter name?
Ma'am
Enter age?
21
Gender?
Female
Enter Employer name?
COMSATS
Enter daily wages?
999999
Enter Grade?
23
Name: Ma'am
Age : 21
Gender : Female
Employer Name: COMSATS
Daily Wages : 999999
Grade: 23
Process returned 0 (0x0)   execution time : 19.947 s
Press any key to continue.

5.3 Question 3:

i. **Define an instance object of class DateTime called Watch.**

ii. **Write a main () function that would initialize the values through the constructor functions, and then allows them to be reset through the set () functions. Be sure and display the results following the constructor before you use the set functions.**

iii. **Through the use of the display () function, the time and date are to be displayed. Note that the display () functions in all three classes need to be defined, as well as the constructor and all the access functions.**

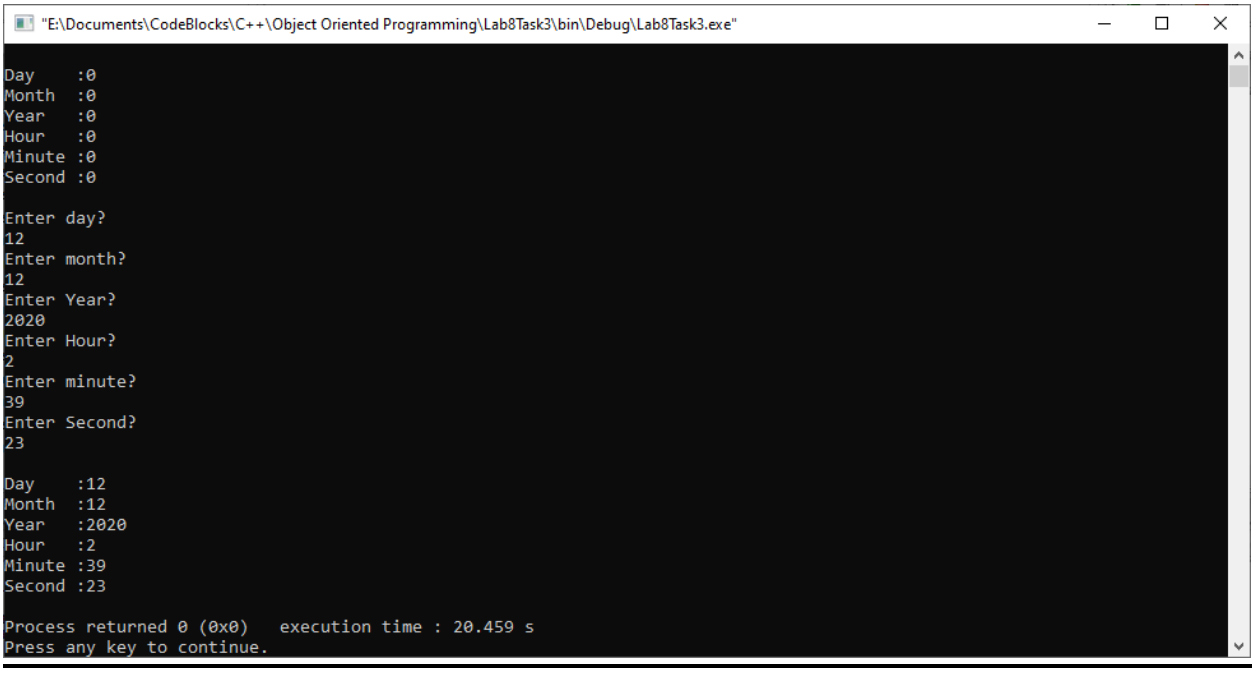
Solution:

The code is given below,

```
#include <iostream>

using namespace std;
class Date
 {
     protected:
         int day;
         int month;
         int year;
    public:

         Date()
         {
             day=0;
             month=0;
             year=0;

         }


         void displayD()     // displays the date
         {
             cout<<endl;
        cout<<"Day    :"<<day<<endl;
        cout<<"Month  :"<<month<<endl;
        cout<<"Year   :"<<year<<endl;
         }
         void getD()         //  accesses the date members
         {  cout<<endl;
        cout<<"Enter day?"<<endl;
        cin>>day;
        cout<<"Enter month?"<<endl;
        cin>>month;
        cout<<"Enter Year?"<<endl;
        cin>>year;
         }
         void setvalueT(int d,int m,int y)
         {
          day=d;
          month=m;
          year=y;
         }
 };
 class Time
```

```
 45   {
 46       protected:
 47            int hour;
 48            int minute;
 49            int second;     // sets the date members
 50     public:
 51             Time()
 52             {
 53                 hour=0;
 54                 minute=0;
 55                 second=0;
 56             }
 57
 58            void getT()
 59            {
 60           cout<<"Enter Hour?"<<endl;
 61           cin>>hour;
 62           cout<<"Enter minute?"<<endl;
 63           cin>>minute;
 64           cout<<"Enter Second?"<<endl;
 65           cin>>second;
 66            }
 67           void displayT() // displays the time          Time get();   // accesses the time
members
 68           {
 69           cout<<"Hour   :"<<hour<<endl;
 70           cout<<"Minute :"<<minute<<endl;
 71           cout<<"Second :"<<second<<endl;
 72
 73           }
 74           void setvalueT(int h, int m, int s)         // sets the time members
 75           {
 76               hour=h;
 77               minute=m;
 78               second=s;
 79           }
 80
 81   };
 82
 83  class DateAndTime : public Date, public Time
 84   {
 85            int digital;
 86
 87          public:
 88              void display()// prints date and time
 89              {
 90                displayD();
 91                displayT();
 92              }
 93   };
 94
 95  int main()
 96  {
 97      DateAndTime Watch;
 98      Watch.display();
 99      Watch.getD();
100      Watch.getT();
101      Watch.display();
102
103      return 0;
104  }
```

Console Output is shown below.


"E:\Documents\CodeBlocks\C++\Object Oriented Programming\Lab8Task3\bin\Debug\Lab8Task3.exe"
Day   :0
Month :0
Year   :0
Hour   :0
Minute :0
Second :0
Enter day?
12
Enter month?
12
Enter Year?
2020
Enter Hour?
2
Enter minute?
39
Enter Second?
23
Day   :12
Month :12
Year   :2020
Hour   :2
Minute :39
Second :23
Process returned 0 (0x0)   execution time : 20.459 s
Press any key to continue.

5.4 Question 4:

Write a class Teacher that contains the attribute teacher name, age and address. It also contains member function to input and display its attributes. Write another class Author that contains the attributes author name, address and number of books written by him. It also contains member functions to input and display its attributes. Write a third class Scholar that inherits both Teacher and Author classes. Test these classes from main() by creating objects of derived classes and testing functions in a way that clear concept of multiple Inheritance.

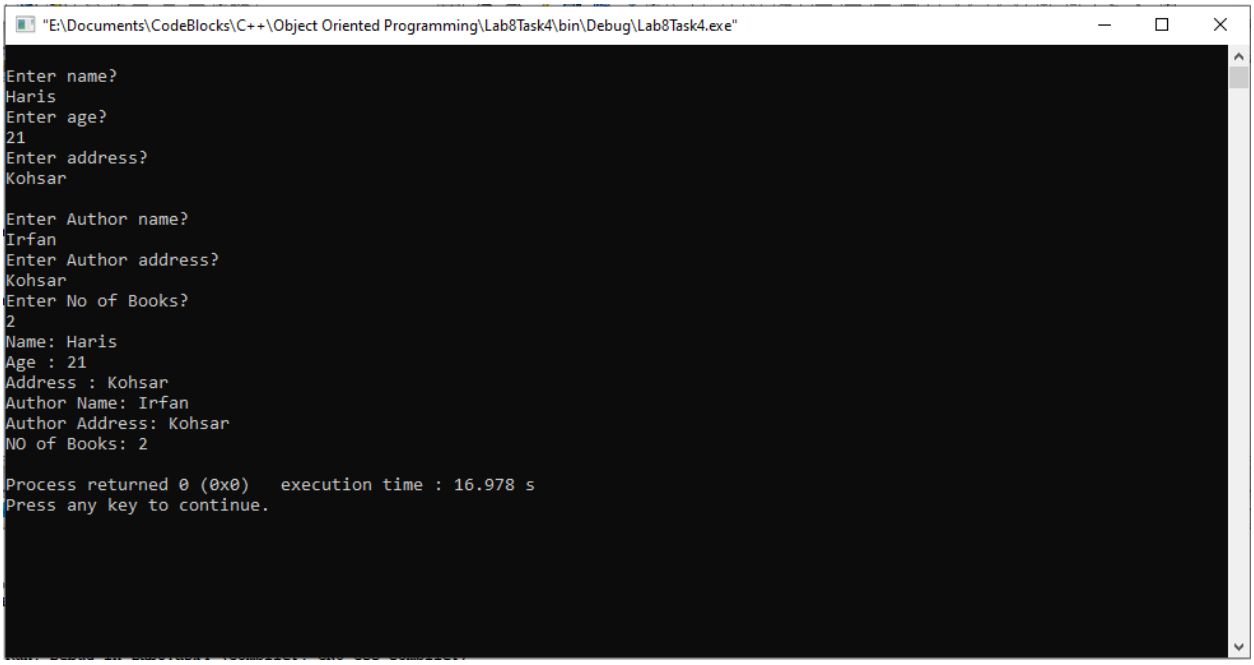
Solution:

I am attaching my code below,

```
#include <iostream>

using namespace std;

class Teacher
{
protected:
        string name;
        int age;
        string address;

public:
      void getT()
    {
        cout<<endl;
        cout<<"Enter name?"<<endl;
        cin>>name;
        cout<<"Enter age?"<<endl;
        cin>>age;
         cout<<"Enter address?"<<endl;
        cin>>address;
    }
      void displayT()
    {
        cout<<"Name: "<<name<<endl;
        cout<<"Age : "<<age<<endl;
        cout<<"Address : "<<address<<endl;

    }

};

class author
{
protected:
    string a_name;
    string a_address;
    int noofbooks;
public:

      void getA()
    {
        cout<<endl;
        cout<<"Enter Author name?"<<endl;
        cin>>a_name;
        cout<<"Enter Author address?"<<endl;
        cin>>a_address;
```

```
48          cout<<"Enter No of Books?"<<endl;
49          cin>>noofbooks;
50
51      }
52        void displayA()
53      {
54          cout<<"Author Name: "<<a_name<<endl;
55          cout<<"Author Address: "<<a_address<<endl;
56          cout<<"NO of Books: "<<noofbooks<<endl;
57      }
58
59  };
60
61  class scholar:public Teacher, public author
62  {
63
64  };
65  int main()
66  {
67     scholar s1;
68
69     s1.getT();
70     s1.getA();
71
72     s1.displayT();
73     s1.displayA();
74
75      return 0;
76  }
```

The result for this program is shown below,



POST LAB

6.1 Question 5:

An organization has two types of employees: regular and adhoc. Regular employees
get a salary which is basic + DA + HRA where DA is 10% of basic and
HRA is 30% of basic. Adhoc employees are daily wagers who get a salary which is equal to Number * Wage.
(ii) Define the constructors. When a regular employee is created, basic must be a parameter.
When adhoc employee is created wage must be a parameter.
(iii) Define the destructors.
(iv) Define the member functions for each class. The member function days ( ) updates number of the Adhoc employee.
(v) Write a test program to test the classes.

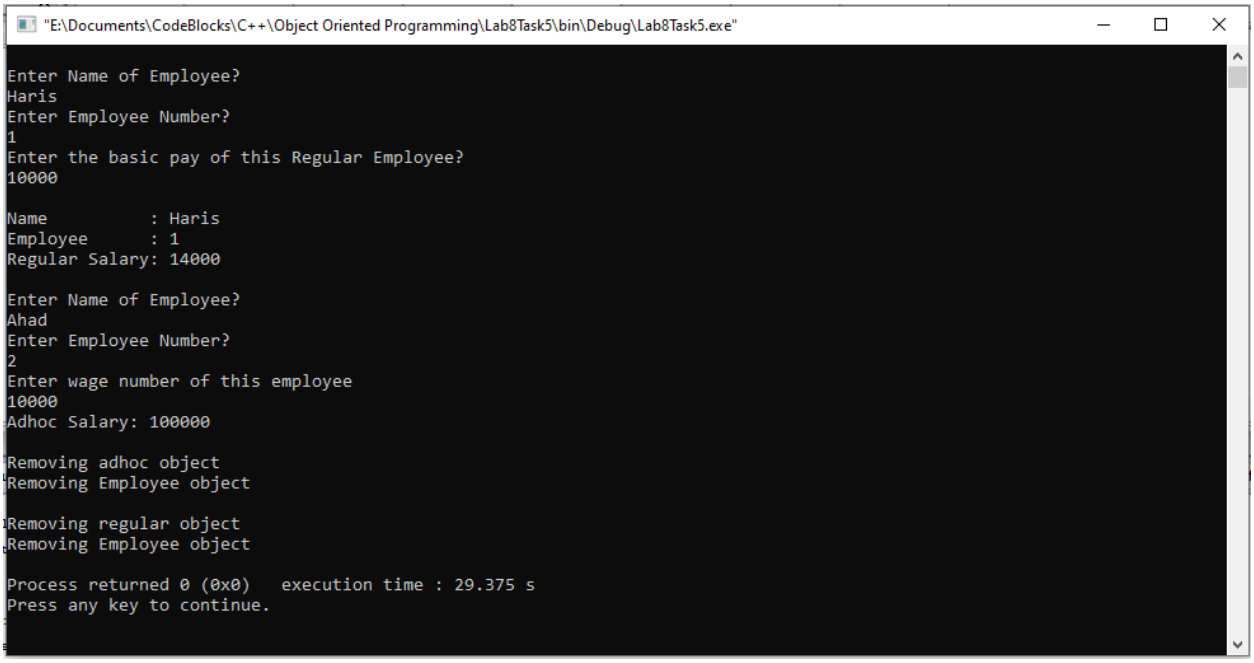
Solution:

I am attaching my code below,

```
#include <iostream>

using namespace std;

class Employee
{
protected:
    string name;
    int empno;
public:
    Employee()
    {
       cout<<endl<<"Enter Name of Employee?"<<endl;
       cin>>name;
       cout<<"Enter Employee Number?"<<endl;
       cin>>empno;
    }
    ~Employee()
    {
         cout<<"Removing Employee object"<<endl;
    }
};
class regular:public Employee
{
protected:
        float basic=0;
        float DA=0;
```

```
        float HRA=0;
        float total=0;

        public:

            regular()
            {
                cout<<"Enter the basic pay of this Regular Employee?"<<endl;
                cin>>basic;
            }
            ~regular()
            {
                cout<<endl<<"Removing regular object"<<endl;
            }
            void salary()
            {
                DA=0.1*basic;
                HRA=0.3*basic;

                total=basic+DA+HRA;
               cout<<endl<<"Name          : "<<name<<endl;
               cout<<"Employee      : "<<empno<<endl;

               cout<<"Regular Salary: "<<total<<endl;
            }


};
class adhoc:public Employee
{
protected:
    int wage,number,tsal;
public:
    adhoc()
    {   number=0;
         tsal=0;
        cout<<"Enter wage number of this employee"<<endl;
        cin>>wage;
    }
    ~adhoc()
    {
         cout<<endl<<"Removing adhoc object"<<endl;
    }
    void days(int n)
    {
      number=n;
    }

   void salary()
    {
        tsal=number*wage;
        cout<<"Adhoc Salary: "<<tsal<<endl;

    }

};
int main()
{
    regular r1;
    r1.salary();

    adhoc a1;
    a1.days(10);
    a1.salary();

    return 0;
}
```

The result for this program is shown below,



6.2 Question 6:

Write a program having a base class Student with data member rollno and member functions getnum() to input rollno and putnum() to display rollno. A class Test is derived from class Student with data member marks and member functions getmarks() to input marks and putmarks() to display marks. Class Sports is also derived from class Student with data member score and member functions getscore() to input score and putscore() to display score. The class Result is inherited from two base classes, class Test and class Sports with data member total and a member function display() to display rollno, marks, score and the total(marks + score).

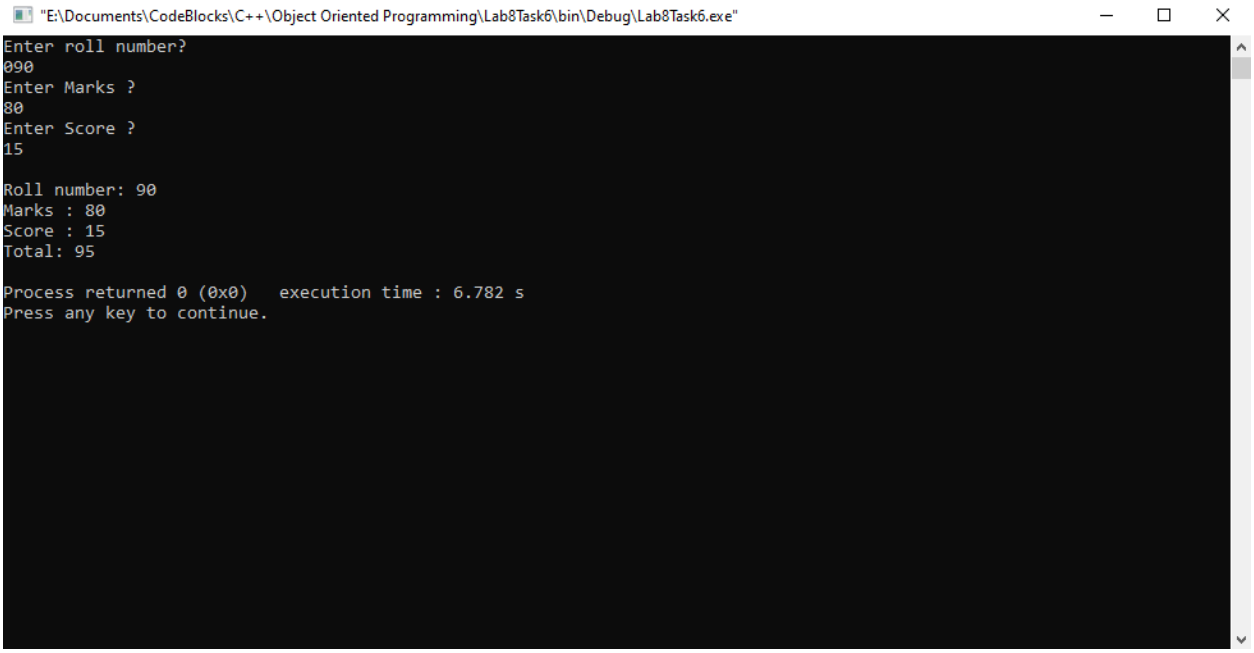
Solution:

I am attaching my code below,

```
#include <iostream>

using namespace std;

class student
{

public:
     int rollno;
    void getnum()
    {
        cout<<"Enter roll number?"<<endl;
        cin>>rollno;
    }
    void putnum()
    {
        cout<<endl<<"Roll number: "<<rollno<<endl;
    }
};
class Test:public student
{
protected:
    int marks;
public:
    void getmarks()
    {   getnum();
        cout<<"Enter Marks ?"<<endl;
        cin>>marks;
    }
    void putmarks()
    {   putnum();
        cout<<"Marks : "<<marks<<endl;
    }
};
class sports: public student
{
protected:
    int score;
public:
    void getscore()
    {
        cout<<"Enter Score ?"<<endl;
```

```
43          cin>>score;
44      }
45      void putscore()
46      {
47          cout<<"Score : "<<score<<endl;
48      }
49  };
50  class result:public Test, public sports
51  {
52  protected:
53      int total;
54  public:
55      void display()
56      {
57      putmarks();
58      putscore();
59      total=score+marks;
60      cout<<"Total: "<<total<<endl;
61      }
62  };
63
64  int main()
65  {
66      result r1;
67      r1.getmarks();
68      r1.getscore();
69      r1.display();
70      return 0;
71  }
```

The result for this program is shown below,



6.3 Question 7:

Write a class LocalPhone that contains an attribute phone to store a local telephone number. The class contains member functions to input and display phone number. Write a child class NatPhone for national phone numbers that inherits LocPhone class. It additionally contains an attribute to store city code. It also contains member functions to input and show the city code. Write another class IntPhone for international phone numbers that inherit NatPhone class. It additionally contains an attribute to store country code. It also contains member functions to input and show the country code. Test these classes from main() by creating objects of derived classes and testing functions in a way that clear concept of multi-level Inheritance.

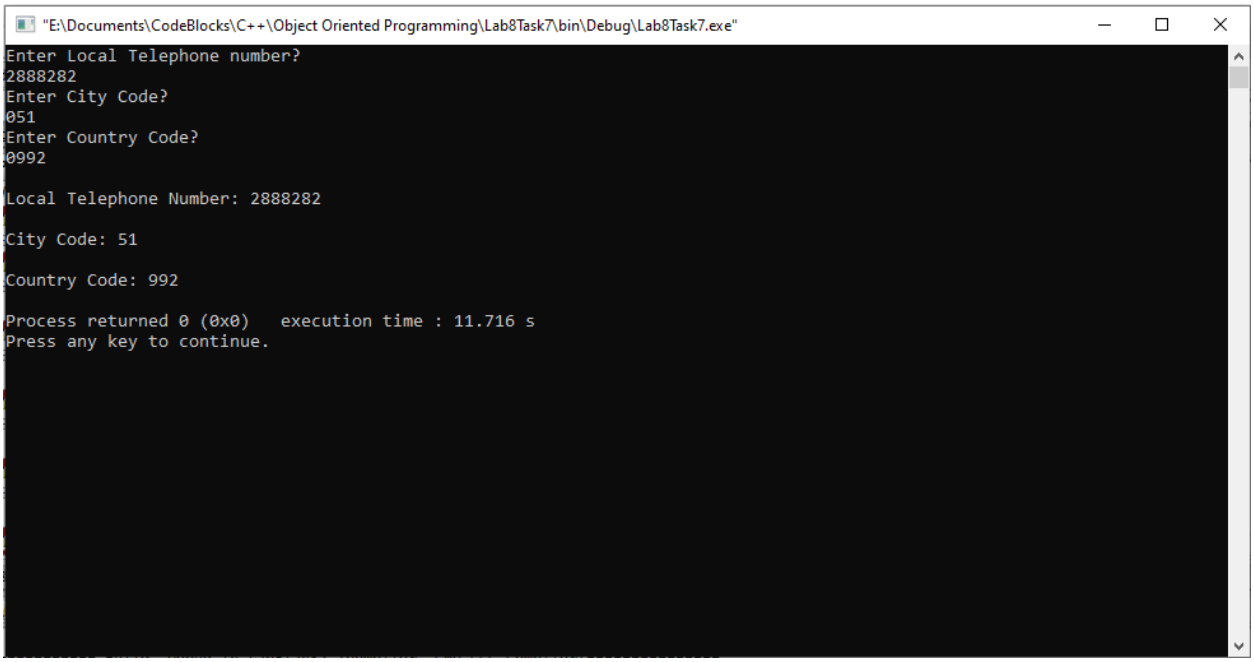
Solution:

I am attaching my code below,

```
#include <iostream>

using namespace std;

class LocalPhone
{
protected:
    int tnum;

public:
     void getnum()
    {
        cout<<"Enter Local Telephone number?"<<endl;
        cin>>tnum;
    }
    void putnum()
    {
        cout<<endl<<"Local Telephone Number: "<<tnum<<endl;
    }
};

class NatPhone:public LocalPhone
{
protected:
    int citycode;
public:

     void getcode()
    {   getnum();
        cout<<"Enter City Code?"<<endl;
        cin>>citycode;
    }
    void putcode()
    {   putnum();
        cout<<endl<<"City Code: "<<citycode<<endl;
    }

};

class IntPhone:public NatPhone
{
protected:
```

```
43      int countrycode;
44
45  public:
46       void getcountrycode()
47      {
48          cout<<"Enter Country Code?"<<endl;
49          cin>>countrycode;
50      }
51      void putcountrycode()
52      {
53          cout<<endl<<"Country Code: "<<countrycode<<endl;
54      }
55  };
56
57  int main()
58  {
59      IntPhone i1;
60      i1.getcode();
61      i1.getcountrycode();
62
63      i1.putcode();
64      i1.putcountrycode();
65      return 0;
66  }
```

The result for this program is shown below,



6.4 Question 8:

Start with the publication, book and tape classes. Add base class sales that holds an array of three floats so that it can record the dollar sales of a particular publication for the last three months. Include a getdata() function to get three sale amount from the user and a putdata() function to display the sales figure.
Alter the book and tape classes, so they are derived from both publication and sales. An object of book or tape should should input and output ans sales data along with other data.
Write a main function to create a book and tape object and exercise their input/output capabilities.

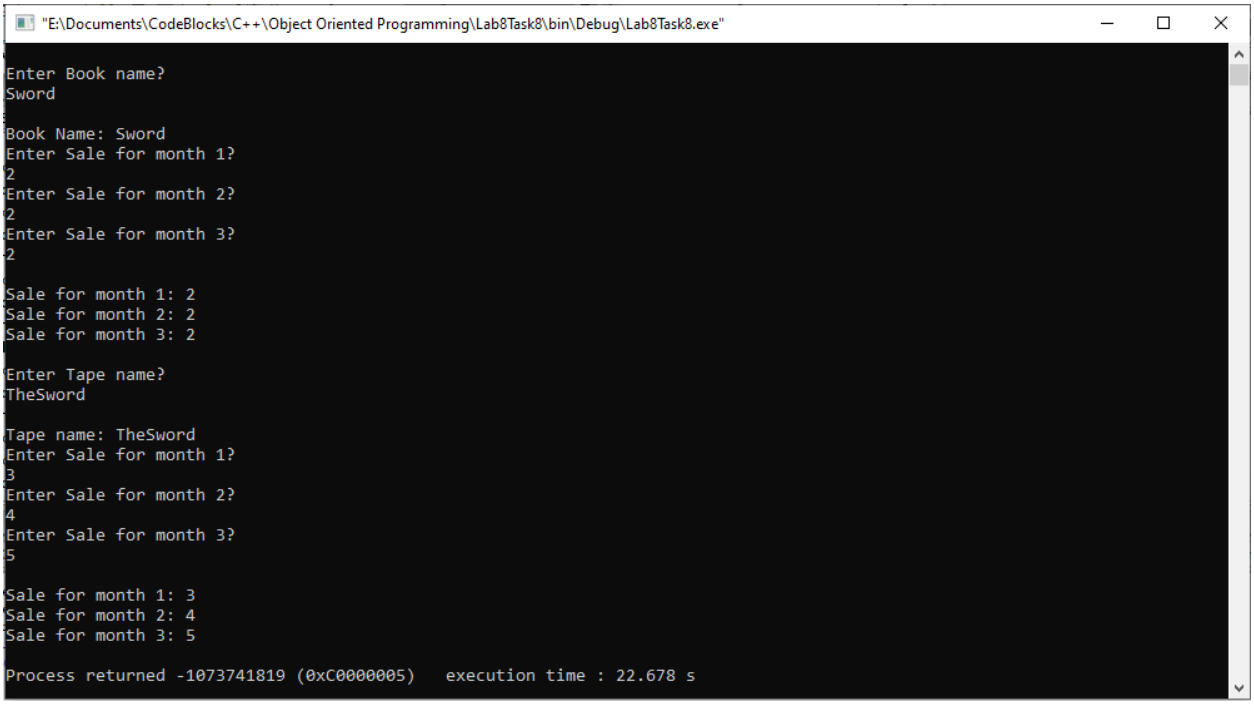
Solution:

I am attaching my code below,

```
#include <iostream>

using namespace std;
class publication
{

};
class sales
{
protected:


public:
    float arr[2];
    void getdata()
    {
        for(int i=0;i<=2;i++)
        {
            cout<<"Enter Sale for month "<<i+1<<"?"<<endl;
            cin>>arr[i];
        }
    }
    void putdata()
    { cout<<endl;

         for(int j=0;j<=2;j++)
        {
            cout<<"Sale for month "<<j+1<<": "<<arr[j]<<endl;

        }
    }
};
class book:public publication,public sales
{
 protected:
     string bname;
public:

      void getbname()
    {
        cout<<endl<<"Enter Book name?"<<endl;
        cin>>bname;

```

```
44      }
45      void putbname()
46      {
47          cout<<endl<<"Book Name: "<<bname<<endl;
48
49      }
50  };
51  class tape:public publication,public sales
52  {
53      protected:
54  string tname;
55      public:
56            void getTname()
57      {
58          cout<<endl<<"Enter Tape name?"<<endl;
59          cin>>tname;
60      }
61      void putTname()
62      {
63          cout<<endl<<"Tape name: "<<tname<<endl;
64
65
66      }
67  };
68
69
70  int main()
71  {
72     book b1;
73     tape t1;
74
75
76
77     b1.getbname();
78     b1.putbname();
79     b1.getdata();
80     b1.putdata();
81
82     t1.getTname();
83     t1.putTname();
84     t1.getdata();
85     t1.putdata();
86
87
88
89
90  }
91
```

The result for this program is shown below,


"E:\Documents\CodeBlocks\C++\Object Oriented Programming\Lab8Task8\bin\Debug\Lab8Task8.exe"
Enter Book name?
Sword
Book Name: Sword
Enter Sale for month 1?
2
Enter Sale for month 2?
2
Enter Sale for month 3?
2
Sale for month 1: 2
Sale for month 2: 2
Sale for month 3: 2
Enter Tape name?
TheSword
Tape name: TheSword
Enter Sale for month 1?
3
Enter Sale for month 2?
4
Enter Sale for month 3?
5
Sale for month 1: 3
Sale for month 2: 4
Sale for month 3: 5
Process returned -1073741819 (0xC0000005) execution time : 22.678 s

___THE END_____