**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.**

1. **Write functions for each class to get and display values.**
2. **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,

1 #include <iostream>

2

3 **using namespace std**;

4

5 **class** base

6 {

7 **protected**:

8 **int** x,y;

9

10 **public**:

11 **void** get()

12 {

13 **cout**<<"Enter the value of x:"<<**endl**;

14 **cin**>>x;

15 **cout**<<"Enter the value of y:"<<**endl**;

16 **cin**>>y;

17

18 }

19

20 **void** display()

21 {

22 **cout**<<"The Value of x is: "<<x<<**endl**;

23 **cout**<<"The Value of y is: "<<y<<**endl**;

24 }

25

26

27 };

28 **class** derived1:**public** base

29 {

30 **protected**:

31 **int** z;

32 **public**:

33 **void** getD()

34 {

35 **cout**<<"Enter the value of z:"<<**endl**;

36 **cin**>>z;

37 }

38

39 **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.**

1. **Write set and get functions to enter and display the data members.**
2. **Write main function to implement these classes. Enter the teacher data to show**

**multiple inheritance.**

**Solution:**The code is given below,

1 #include <iostream>

2

3 **using namespace std**;

4

5 **class** person

6 {

7 **protected**:

8 **string** name;

9 **string** Gender;

10 **int** age;

11 **public**:

12 **void** get()

13 {

14 **cout**<<**endl**;

15 **cout**<<"Enter name?"<<**endl**;

16 **cin**>>name;

17 **cout**<<"Enter age?"<<**endl**;

18 **cin**>>age;

19 **cout**<<"Gender?"<<**endl**;

20 **cin**>>Gender;

21

22 }

23

24 **void** display()

25 {

26 **cout**<<"Name: "<<name<<**endl**;

27 **cout**<<"Age : "<<age<<**endl**;

28 **cout**<<"Gender : "<<Gender<<**endl**;

29 }

30 };

31

32 **class** employee

33 {

34 protected:

35 string employername;

36 float dailywages;

37 public:

38 void getE()

39 {

40 cout<<"Enter Employer name?"<<endl;

41 cin>>employername;

42 cout<<"Enter daily wages?"<<endl;

43 cin>>dailywages;

44

45

46 }

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();

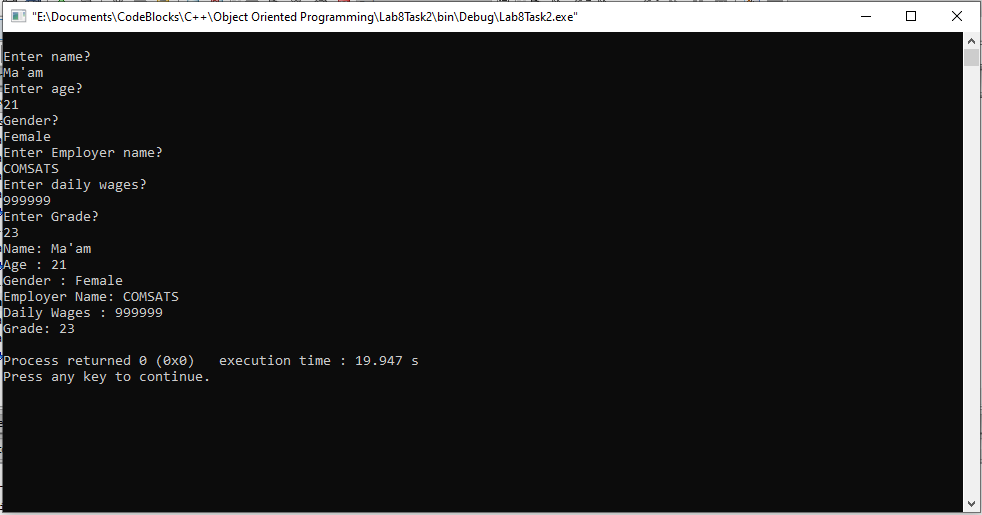
86 t1.displayT();

87

88

89 return 0;

90 }

**Console Output is shown below.**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5.3 Question 3:

1. **Define an instance object of class DateTime called Watch.**
2. **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.**
3. **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,

1 #include <iostream>

2

3 **using namespace std**;

4 **class** Date

5 {

6 **protected**:

7 **int** day;

8 **int** month;

9 **int** year;

10 **public**:

11

12 Date()

13 {

14 day=0;

15 month=0;

16 year=0;

17

18 }

19

20

21 **void** displayD() // displays the date

22 {

23 **cout**<<**endl**;

24 **cout**<<"Day :"<<day<<**endl**;

25 **cout**<<"Month :"<<month<<**endl**;

26 **cout**<<"Year :"<<year<<**endl**;

27 }

28 **void** getD() // accesses the date members

29 { **cout**<<**endl**;

30 **cout**<<"Enter day?"<<**endl**;

31 **cin**>>day;

32 **cout**<<"Enter month?"<<**endl**;

33 **cin**>>month;

34 **cout**<<"Enter Year?"<<**endl**;

35 **cin**>>year;

36 }

37 **void** setvalueT(**int** d,**int** m,**int** y)

38 {

39 day=d;

40 month=m;

41 year=y;

42 }

43 };

44 **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();

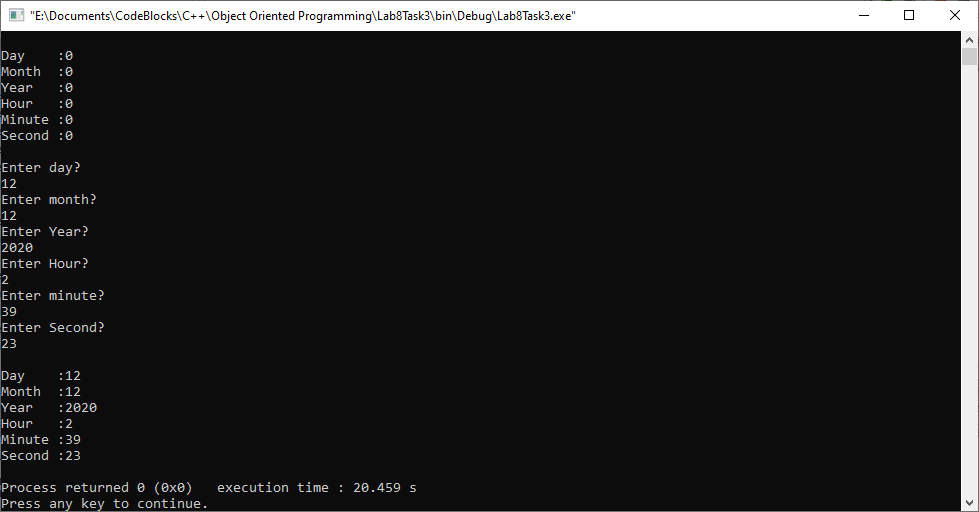
101 Watch.display();

102

103 **return** 0;

104 }

**Console Output is shown below.**

****

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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,

1 #include <iostream>

2

3 **using namespace std**;

4

5 **class** Teacher

6 {

7 **protected**:

8 **string** name;

9 **int** age;

10 **string** address;

11

12 **public**:

13 **void** getT()

14 {

15 **cout**<<**endl**;

16 **cout**<<"Enter name?"<<**endl**;

17 **cin**>>name;

18 **cout**<<"Enter age?"<<**endl**;

19 **cin**>>age;

20 **cout**<<"Enter address?"<<**endl**;

21 **cin**>>address;

22 }

23 **void** displayT()

24 {

25 **cout**<<"Name: "<<name<<**endl**;

26 **cout**<<"Age : "<<age<<**endl**;

27 **cout**<<"Address : "<<address<<**endl**;

28

29 }

30

31 };

32

33 **class** author

34 {

35 **protected**:

36 **string** a\_name;

37 **string** a\_address;

38 **int** noofbooks;

39 **public**:

40

41 **void** getA()

42 {

43 **cout**<<**endl**;

44 **cout**<<"Enter Author name?"<<**endl**;

45 **cin**>>a\_name;

46 **cout**<<"Enter Author address?"<<**endl**;

47 **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();

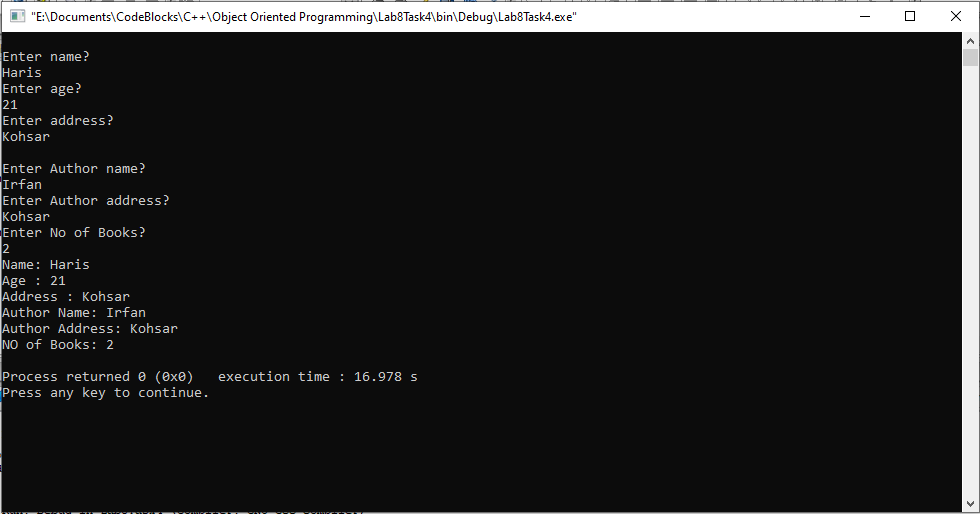
73 s1.displayA();

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,

1 #include <iostream>

2

3 **using namespace std**;

4

5 **class** Employee

6 {

7 **protected**:

8 **string** name;

9 **int** empno;

10 **public**:

11 Employee()

12 {

13 **cout**<<**endl**<<"Enter Name of Employee?"<<**endl**;

14 **cin**>>name;

15 **cout**<<"Enter Employee Number?"<<**endl**;

16 **cin**>>empno;

17 }

18 ~Employee()

19 {

20 **cout**<<"Removing Employee object"<<**endl**;

21 }

22 };

23 **class** regular:**public** Employee

24 {

25 **protected**:

26 **float** basic=0;

27 **float** DA=0;

28 **float** HRA=0;

29 **float** total=0;

30

31 **public**:

32

33 regular()

34 {

35 **cout**<<"Enter the basic pay of this Regular Employee?"<<**endl**;

36 **cin**>>basic;

37 }

38 ~regular()

39 {

40 **cout**<<**endl**<<"Removing regular object"<<**endl**;

41 }

42 **void** salary()

43 {

44 DA=0.1\*basic;

45 HRA=0.3\*basic;

46

47 total=basic+DA+HRA;

48 **cout**<<**endl**<<"Name : "<<name<<**endl**;

49 **cout**<<"Employee : "<<empno<<**endl**;

50

51 **cout**<<"Regular Salary: "<<total<<**endl**;

52 }

53

54

55 };

56 class adhoc:public Employee

57 {

58 protected:

59 int wage,number,tsal;

60 public:

61 adhoc()

62 { number=0;

63 tsal=0;

64 cout<<"Enter wage number of this employee"<<endl;

65 cin>>wage;

66 }

67 ~adhoc()

68 {

69 cout<<endl<<"Removing adhoc object"<<endl;

70 }

71 void days(int n)

72 {

73 number=n;

74 }

75

76 void salary()

77 {

78 tsal=number\*wage;

79 cout<<"Adhoc Salary: "<<tsal<<endl;

80

81 }

82

83 };

84 int main()

85 {

86 regular r1;

87 r1.salary();

88

89 adhoc a1;

90 a1.days(10);

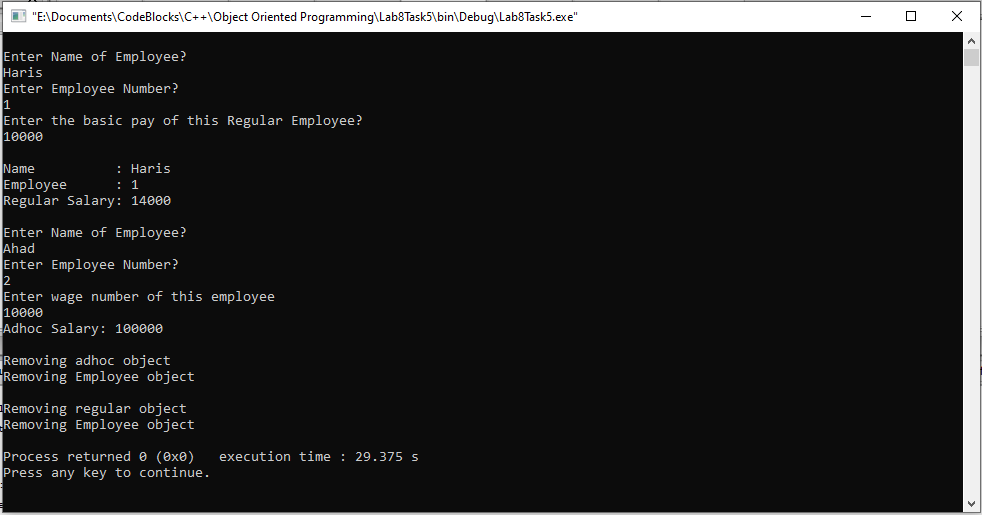
91 a1.salary();

92

93 return 0;

94 }

**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,

1 #include <iostream>

2

3 **using namespace std**;

4

5 **class** student

6 {

7

8 **public**:

9 **int** rollno;

10 **void** getnum()

11 {

12 **cout**<<"Enter roll number?"<<**endl**;

13 **cin**>>rollno;

14 }

15 **void** putnum()

16 {

17 **cout**<<**endl**<<"Roll number: "<<rollno<<**endl**;

18 }

19 };

20 class Test:public student

21 {

22 protected:

23 int marks;

24 public:

25 void getmarks()

26 { getnum();

27 cout<<"Enter Marks ?"<<endl;

28 cin>>marks;

29 }

30 void putmarks()

31 { putnum();

32 cout<<"Marks : "<<marks<<endl;

33 }

34 };

35 class sports: public student

36 {

37 protected:

38 int score;

39 public:

40 void getscore()

41 {

42 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,

1 #include <iostream>

2

3 **using namespace std**;

4

5 **class** LocalPhone

6 {

7 **protected**:

8 **int** tnum;

9

10 **public**:

11 **void** getnum()

12 {

13 **cout**<<"Enter Local Telephone number?"<<**endl**;

14 **cin**>>tnum;

15 }

16 **void** putnum()

17 {

18 **cout**<<**endl**<<"Local Telephone Number: "<<tnum<<**endl**;

19 }

20 };

21

22 **class** NatPhone:**public** LocalPhone

23 {

24 **protected**:

25 **int** citycode;

26 **public**:

27

28 **void** getcode()

29 { getnum();

30 **cout**<<"Enter City Code?"<<**endl**;

31 **cin**>>citycode;

32 }

33 **void** putcode()

34 { putnum();

35 **cout**<<**endl**<<"City Code: "<<citycode<<**endl**;

36 }

37

38 };

39

40 **class** IntPhone:**public** NatPhone

41 {

42 **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

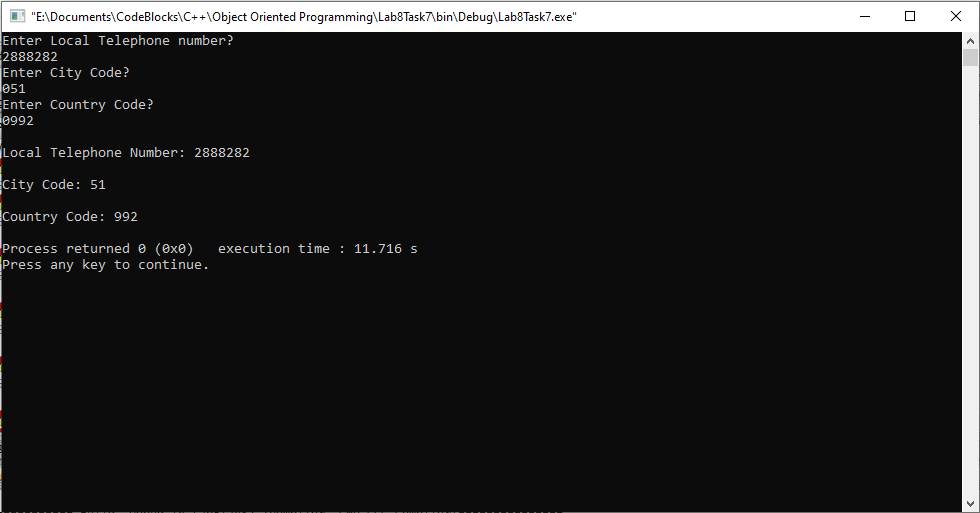
63 i1.putcode();

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,

1 #include <iostream>

2

3 **using namespace std**;

4 **class** publication

5 {

6

7 };

8 **class** sales

9 {

10 **protected**:

11

12

13 **public**:

14 **float** arr[2];

15 **void** getdata()

16 {

17 **for**(**int** i=0;i<=2;i++)

18 {

19 **cout**<<"Enter Sale for month "<<i+1<<"?"<<**endl**;

20 **cin**>>arr[i];

21 }

22 }

23 void putdata()

24 { cout<<endl;

25

26 for(int j=0;j<=2;j++)

27 {

28 cout<<"Sale for month "<<j+1<<": "<<arr[j]<<endl;

29

30 }

31 }

32 };

33 class book:public publication,public sales

34 {

35 protected:

36 string bname;

37 public:

38

39 void getbname()

40 {

41 cout<<endl<<"Enter Book name?"<<endl;

42 cin>>bname;

43

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

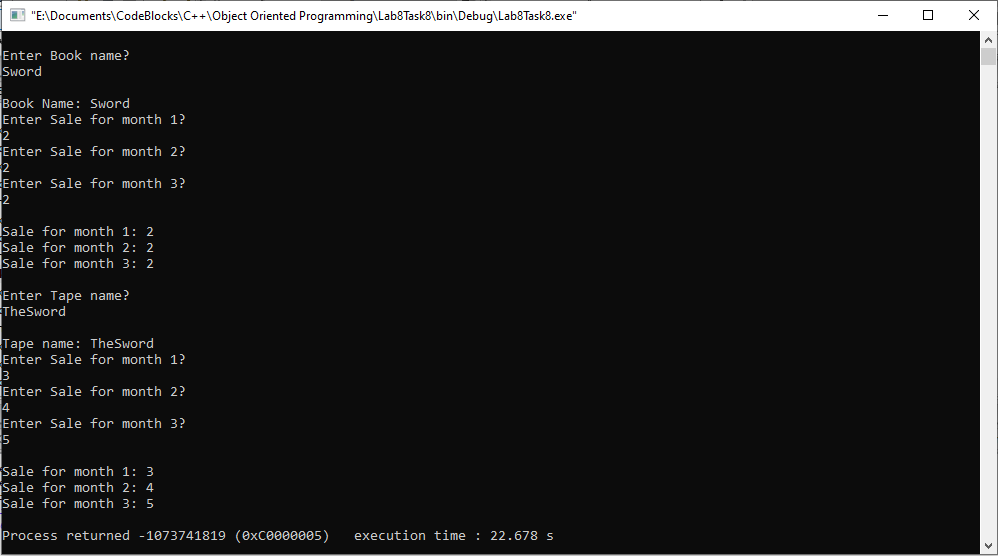
88

89

90 }

91

**The result for this program is shown below,**



\_\_\_THE END\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_