OOP Lab 7: Inheritance types and function overriding

Date: 20-11-2020

Total marks: 15+15=30

Deadline: same day (Friday, 20-11-2020) before 11:59 pm (sharp)

Submission instructions

1. Submit your file in a .zip or .rar format. Name your file with your Roll number (e.g. BSEF19M009.zip or BSEF19M009.rar)
2. Make a separate folder for each lab task (e.g. task1, task2…). All the headers and .cpp files must be in respective folder with a screenshot of the output of a program of the respective task.
3. Every Cpp file must contain your name and roll number(In comments) at the top of each program
4. Don't enclose your code in comments otherwise it will not be evaluated.

Instructions: (MUST READ)

- **Understanding the problem statement is a part of that question. Do whatever you are asked and what you understand.NO QUERIES ARE ALLOWED as this lab is overly simple**
- **No compensation or makeup lab.**
- **Don't discuss with peers. Changing variable names/ changing for to while loop will not help you in hiding cheating attempt!**
- **Cheating cases will simply be dropped out and their future lab submissions will NOT be graded.**
- **You are not allowed to consult Internet. Plagiarism cases will be strictly dealt.**
- **You must submit the lab solution BEFORE 11:59 pm. Even a few minutes late submissions will not be considered. Make sure to do proper management of time/Internet connectivity/power failure or whatever issue is possible!**

General instructions for all tasks: (marks will be deducted if the instructions are violated)

Note: All the programs should be implemented using class. You can take input in main() function and then call appropriate methods/ member functions of a designed class to set and get values. YOU CAN CREATE A TASK IN SINGLE CPP FILE FOR CLASS DECLARATION AND DEFINITION.

• The attributes of class should be declared as **private** and member functions as **public**.

• You should not initialize the attributes while declaring them in class. The values should be assigned using member functions only. E.g. you cannot declare like:

```
Class Person
{
        Private:
```

int age=25;
}

• The values should be initialized using a constructor. There must be a constructor in your defined class.

• All inputs should be taken in *main()* and all the final results should also be reported/ displayed in the main function.

• All the logic should be implemented in class' member functions. Main() should only input and output relevant values by calling relevant functions of the class.

Hierarchical inheritance

Design a class of Book that has two members: title of book and author. It should have three methods as:

- show() overriden method that prints the data members of Book class on the screen.
- setTitle(string) that sets the title of book
- setAuthor(string) that sets the author of book.

Also design a class Fiction that has one data integer data member "level". It should have following methods:

- setLevel(int, string, string) that invokes setTitle and setAuthor methods of Book class and sets the data member (level).
- show() overriden method that invokes Book class method and prints level.

Also design a class NonFiction that has one data integer data member "number of pages". It should have following methods:

- setPages(int, string, string) that invokes setTitle and setAuthor methods of Book class and sets the data member (number of pages).
- show() overriden method that invokes Book class method and prints number of pages.

In main(), create a static array of objects for both NonFiction and Fiction class. For each NonFiction object, you should invoke/ call the methods (setPages(int, string, string) and show()). Similarly, for Fiction class, invoke a method (setLevel(int, string, string) and show()).

Evaluation criteria:

Part	Marks distribution
Correct identification and implementation of relationships between three classes	3
Book Class	4
Fiction class	3
Non-fiction class	3
Main()	2
Total	15

Multi-level inheritance

Consider a class of **Employee** having employee ID, name, and designation as data members. This class should have at least two member functions. One function takes input from users and update the data members. The second function should display the value of data members.
Design another class **Salary** that has five data members: basic pay, human resource allowance, dearness allowance, profitability fund, and net pay. This class should have following functions:

- **getEmployeeDetails()** that take input from users in all the data members and calculate net pay using formula:

Net pay = basic pay + human resource allowance + dearness allowance - profitability fund
This function, at the start, should invoke the function of employee class that takes input from the user.

- Display() that first invokes display function of employee class and then display the data members of Salary class (including net pay)

Design another class named **BankCredit** that has two data members: bank name and account number. It should have two member functions as:

- getBankDetails() that first invokes the getter method of Salary class and then takes input from user for bank name and account number.
- Print() that first invokes display method of Salary class and then display the data members of BankCredit class.

In main(), declare array of objects of BankCredit class with size 5. It should then take input from the user about the number of employees. It should then invoke getBankDetails() and print() function for each employee.
You need to invoke/ call methods in the same way as asked. Otherwise this will result in marks deduction.

Evaluation criteria:

Part	Marks distribution
Correct relationships created between three classes, respectively.	3
Employee class	3
Salary class	3
BankCredit class	3
Main()	3
Total	15