Lab 10

Object Oriented Programming

Instructions:

- **Attempt the following tasks exactly in the given order.**
- You are required to create a **multi-file project** for each task.
- Indent your code properly.
- Use meaningful variable and function names. Follow the naming conventions.
- Use meaningful prompt lines and labels for all input/output.
- Make sure that there are **NO dangling pointers** or **memory leaks** in your program.

Task # 1

Implement a class named **Employee**. The class should hold the following information in its private member variables:

- Employee name (a CS. btring which can contain at most 50 characters)
- Employee number (an integer)

You are required to implement the following public member functions of the **Employee** class:

- A parameterized (overloaded) constructor (which takes **2** parameters)
- Appropriate getter and setter functions
- A function named **display** to display the details of an Employee on screen

Next, implement a class named **ProductionWorker** that is derived from the **Employee** class. The **ProductionWorker** class should have member variables to hold the following information:

- Shift (an integer)
- Hourly pay rate (a double)

Note: The workday is divided into two shifts: day and night. The shift variable will hold an integer value representing the shift that the employee works. The day shift is shift 1 and the night shift is shift 2.

Implement the following member functions for the **ProductionWorker** class:

- A parameterized (overloaded) constructor (which should take **4** parameters)
- Appropriate getter and setter functions
- A function named **display** to display all details of a **ProductionWorker** on screen

Demonstrate the above classes and their member functions by writing a program that uses a **ProductionWorker** object.

Task #2

In a particular factory, a team leader is a production worker who leads a small team of workers.In addition to hourly pay, team leaders earn a fixed monthly bonus. Team leaders are required to attend a minimum number of hours of training per year. Implement a **TeamLeader** class that inherits from the **ProductionWorker** class you designed in **Task # 1** The **TeamLeader** class should have private member variables to hold the following information:

- Monthly bonus amount (a double)
- Required number of training hours (an integer)
- Number of training hours that the team leader has attended (an integer)

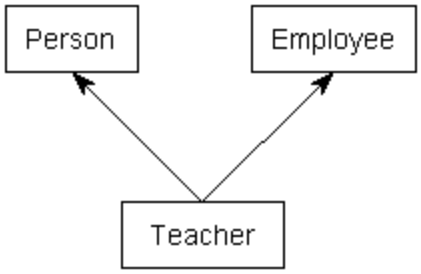
Implement the following member functions for the **TeamLeader** class:

- A parameterized (overloaded) constructor (which should take **7 parameters**)
- Appropriate getter and setter functions
- A function named **display** to display all details of a **TeamLeader** on screen

Demonstrate the **TeamLeader** class by writing a program that uses a **TeamLeader** object.

Task#3

Provide the C++ implementation of the following.



The Person class has name and age as its attributes. It has an overloaded constructor to initialize the values and appropriate accessor and mutator methods. The Employee class has name of the employer and wage as its attributes with an overloaded constructor and appropriate accessor and mutator methods.

The Teacher class is inherited from Person and Employee class with an attribute of Pay Scale of type integer. It has overloaded constructor, appropriate accessor, mutator methods and a display function to print the Name, Age, Name of Employer, Wage and Pay Scale of Teacher.

Task#4

Create a base class, called BankAccount, and two additional classes (each derived from BankAccount), called SavingsAccount and CheckingAccount.

BankAccount:

- Title
- AccountNumber
- Balance
- Deposit()
- Withdraw()

SavingAccount:

- InterestRate
- CalculateInterest()

CheckingAccount:

- fee charged per transaction

Class CheckingAccount should redefine member functions withdraw and deposit so that they subtract the fee from the account balance whenever either transaction is performed successfully.

You will then test the operations of each class in function main() to simulate the transactions of both a checking account and a savings account.