HW 8: Inheritance

Write a main program that creates three accounts (one from each class: BankAccount, MoneyMarketAccount, and CDAccount) and transfers funds from each account to another one. Show relationship between these three classes using a UML diagram (see a Bb link: UML class diagram tutorial).

Create a base class called **BankAccount** that has member variables:

- the **name** of the owner of the account (a string)
- the **balance** in the account (double)

and member functions:

- ***getName***
- ***getBalance***
- ***deposit(float amount)//*** adds the amount (passed as a parameter) to the balance (if the amount is nonnegative),
- ***withdraw(float amount), //*** subtracts the amount from the balance (if the amount is nonnegative and less than or equal to the balance)
- ***transfer(float amount, BankAccount &ToAccount,)*** withdraws from the current account (on which it is called) and deposits into another account (passed as a parameter), implemented calling ***withdraw(amount)*** and ***ToAccount.deposit(amount)*.**

Also, create a class called **MoneyMarketAccount** that is derived from **BankAccount**. In a MoneyMarketAccount the user gets 1 free withdrawal. After the free withdrawal has been used, a withdrawal fee of $1.50 is deducted from the balance per withdrawal. Hence, the class must have a data member to keep track of the **number of withdrawals** and override the ***withdraw function***.

Finally, create a **CDAccount** class derived from **BankAccount,** which in addition to having the name and balance has a member variable ***interest rate*** **(<1)** and a member function ***apply_interest()***, which increases balance by adding interest: ***balance+= interest rate*balance***.

CD accounts incur penalties for the early withdrawal of funds. A withdrawal of any amount incurs a penalty = 25% of an interest rate applied to the balance. Assume the amount withdrawn plus the penalty are deducted from the account balance. Again the ***withdraw function*** must override the one in the base class.

For all 3 classes, the ***withdraw function*** should return an integer indicating the status and output a message (either 0:"ok" or 1: "insufficient funds for the withdrawal to take place"). For ***transfer*** to work properly, ***withdraw*** should be declared ***virtual.***

For all 3 classes create constructors (default and with parameters) and overloaded << (output) operator, reuse constructors and operator << of the base class in the derived classes.

In main

- create 3 accounts of various types
- for each account perform withdraw and deposit, output the type (BankAccount/MoneyMarket/CDAccount) and status (values of all member variables) of the account before and after
- for each pair of accounts transfer money between them, output the type and status of the account before and after transfer

Submit UML diagram, code, and output of the program.