CL-217 OBJECT ORIENTED PROGRAMMING LAB

LAB TASK 11

INSTRUCTOR: MUHAMMAD HAMZA

SEMESTER SPRING 2020

Banks offer various types of accounts, such as savings, checking, certificate of deposits, and money market, to attract customers as well as meet their specific needs. Two of the most commonly used accounts are savings and checking. Each of these accounts has various options. For example, you may have a savings account that requires no minimum balance but has a lower interest rate. Similarly, you may have a checking account that limits the number of checks you may write. Another type of account that is used to save money for the long term is certificate of deposit (CD).

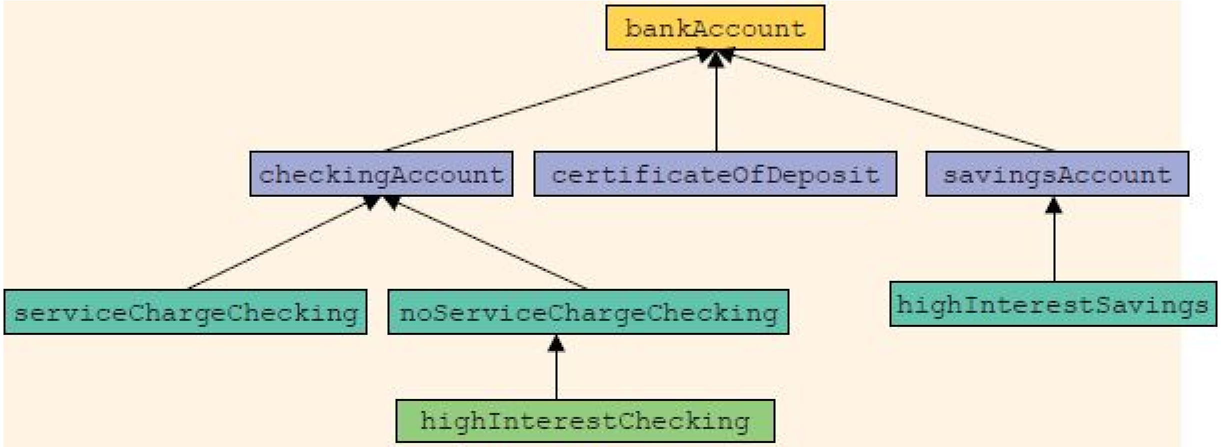
In this programming exercise, you use abstract classes and pure virtual functions to design classes to manipulate various types of accounts. For simplicity, assume that the bank offers three types of accounts: savings, checking, and certificate of deposit, as described next.

Savingsaccounts:

Suppose that the bank offers two types of savings accounts:one that has no minimum balance and a lower interest rate and another that requires a minimum balance and has a higher interest rate.

Checking accounts: Suppose that the bank offers three types of checking accounts: one with a monthly service charge, limited check writing, no minimum balance, and no interest; another with no monthly service charge, a minimum balance requirement, unlimited check writing and lower interest; and a third with no monthly service charge, a higher minimum requirement, a higher interest rate, and unlimited check writing.

Certificate of deposit (CD): In an account of this type, money is left for some time, and these accounts draw higher interest rates than savings or checking accounts. Suppose that you purchase a CD for six months. Then we say that the CD will mature in six months. Penalty for early withdrawal is stiff.



Note that the classes bank Account and checking Account are abstract. That is, we cannot instantiate objects of these classes. The other classes in the figure above are not abstract.

bankAccount: Every bank account has an account number, the name of the owner, and a balance. Therefore, instance variables such as name,accountNumber, and balance should be declared in the abstract class bankAccount. Some operations common to all types of accounts are retrieve account owner's name, account number, and account balance; make deposits; withdraw money; and create monthly statements. so include functions to implement these operations. Some of these functions will be pure virtual.

checkingAccount:A checking account is a bank account. Therefore, it inherits all the properties of a bank account. Because one of the objectives of a checking account is to be able to write checks, including the pure virtual function writeCheck to write a check.

serviceChargeChecking:A Service charge checking account is a checking account. Therefore, it inherits all the properties of a checking account. For simplicity, assume that this type of account does not pay any interest, allows the account holder to write a limited number of checks each month, and does not require any minimum balance. Include appropriate named constants, instance variables, and functions in this class.

noServiceChargeChecking:A checking account with no monthly service charge is a checking account. Therefore, it inherits all the properties of a checking account. Furthermore, this type of account pays interest, allows the account holder to write checks, and requires a minimum balance.

highInterestChecking:A checking account with high interest is a checking account with no monthly service charge. Therefore, it inherits all the properties of a no service charge checking account. Furthermore, this type of account pays higher interest and requires a higher minimum balance than the no service charge checking account.

savingsAccount:A savings account is a bank account. Therefore, it inherits all the properties of a bank account. Furthermore, a savings account also pays interest.

highInterestSavings:A high-interest savings account is a savings account. Therefore, it inherits all the properties of a savings account. It also requires a minimum balance.

certificateOfDeposit:A certificate of deposit account is a bank account. Therefore, it inherits all the properties of a bank account. In addition, it has instance variables to store the number of CD maturity months, interest rate, and the current CD month.

INSTRUCTIONS:

- Read the description thoroughly and understand the hierarchy of the classes, the given figure explains it very well.
- First of all implement the hierarchy of classes using inheritance.
- Add functionality as described under each bank account.
- Make use of pure virtual functions to make desired classes abstract.
- For the functions like withdraw: think of what happens usually when you withdraw some money from the account, the withdrawal amount gets deducted from the balance and a receipt is generated describing how much amount has been withdrawn and how much balance is left. Do the same in your withdraw function: deduct the amount from balance and print out the receipt info on the screen.
- For all the other functions follow the same procedure as described for the withdraw function.
- **Ask any query related to the lab task if you still have any doubts.**