Assignment 1

F2020

Review Project- Bank Account

Complete the Bank Account project we began together. You may begin from my code, or from your own code. I've tried to mark the new additions with an asterisk (*) but may have missed some. Read the instructions carefully. The completed project will consist of at least four classes:

BankAccount

Item	Description
Account number (`int`) field	Must be a value between 1 000 and 9 999 (endpoints inclusive).
Balance (`double`) field	Must not be negative, and must be less than $100 000.
Static value for the "next account number"	This is used to set the account number when none is provided. Incremented each time it is used.
Default constructor	A constructor that takes no arguments. Sets the bank account to reasonable values.
All properties constructor	Has arguments to provide values for all the object properties.
Setters/Getters methods	Accessor and mutator methods for the properties. Must not allow unacceptable values.
*Deposit method (your choice whether it returns anything)	Accepts a value to be added to the balance. Must not allow negative values to be deposited. Must not allow the balance to go out of range.
*Withdraw method (your choice whether it returns anything)	Accepts a value to be subtracted from the balance. Must not allow negative values to be withdrawn (i.e. values that would increase the balance). Must not allow the balance to go out of range.
Static validation methods	Validation methods that will be used by other methods (example constructors) to validate balances and account numbers. They return `Boolean` true when the values provided as arguments are acceptable and false when they are not. After you have built these, make sure the constructors and set methods use them (remove the existing validation from these locations).

SavingsAccount extends BankAccount

Item	Description
Interest rate (`double`) property	Stores an interest rate for the account – must be validated between 0.5% (0.005) and 3.0% (0.03). You can do in place validation or create validation methods.
Default constructor	A constructor that takes no arguments. Sets the bank account to reasonable values.
All properties constructor	Has arguments to provide values for all the object properties.

Assignment 1

F2020

Item	Description
Setters/Getters	Accessor and mutator methods for the new properties. Must not allow unacceptable values.
Apply interest method (your choice whether it returns anything)	Calculates and adds interest to the balance when called. This method may not increase the account balance over the top limit. If you want to make it more realistic, divide the interest rate by 12 before calculating to simulate a monthly payment.

*`ChequingAccount` extends `BankAccount`

Item	Description
Transaction count (`int`)	Stores the number of transactions.
Cheque method (your choice whether it returns anything)	Debits the account by the amount of the cheque (pass as argument). Can delegate to the withdraw method if you like. Besides reducing the balance also increments the transaction count.
??	Do you need a method for incrementing the transaction count? You would if there were other classes that could change this value – but if everything is internal then you can manage it directly. Student's choice.
Default constructor	A constructor that takes no arguments. Sets the bank account to reasonable values.
All properties constructor	Has arguments to provide values for all the object properties.
Setters/Getters	Accessor and mutator methods for the new properties. Must not allow unacceptable values.
Apply fees method (your choice whether it returns anything)	Calculates and subtracts fees from the balance ($0.25 per transaction). Also resets transaction count to zero. This method is allowed to set the balance to a negative amount.

`Customer` Class

Item	Description
Name (`string`)	No validation
Accounts (`BankAccount[]`)	An array of the bank accounts owned by this customer
Number of accounts (`int`)	This will be a helper property to keep track of how many accounts are in the account array. Passing arrays by pointer means you will lose track of how many elements are in the array otherwise.
Default constructor	A constructor that takes no arguments - what will you set the Accounts array to? You might choose to use NULL - make sure you program around that carefully.
All properties constructor	Has arguments to provide values for all the object properties.
*Setters/Getters methods	Accessor and mutator methods for the properties. Must not allow unacceptable values.

Assignment 1

F2020

*Overloaded operators

Overload the insertion and extraction operators to work with all objects (three different kinds of bank accounts and customers). You should use the same validation as the class members do (hint: you can use the set/get methods and/or static validation methods).

Main method

Create a main method that tests the rest of your code

- At least one instance of every class must be created.
- At least one customer should have both a chequing and a savings account.
- Did you allow customers with zero accounts? If so then test a customer with no accounts.
- Every method in your code should be tested at least once – but remember that one method call might end up calling other methods.

Style

Your program must be neat, well formatted, and readable (see the style guide on iLearn). Remember you can lose up to 30% for poor style. **Don't forget comments that identify the programmer, the program, and the date the program was written.**

Upload

Upload your source code file(s) only (they should end in .cpp) to the iLearn dropbox.