**Assignment - In progress**

|  |  |
| --- | --- |
| **Title** | Assignment 7 |
| **Due** | Nov 14, 2017 11:55 pm |
| **Number of resubmissions allowed** | Unlimited |
| **Accept Resubmission Until** | Nov 14, 2017 11:50 pm |
| **Status** | Not Started |
| **Grade Scale** | Points (max 100.0) |
| **Modified by instructor** | Nov 7, 2017 12:58 pm |

**Instructions**

Assignment  7

Assignment 7 is based on a Group Project in your book, Chapter 15, page 987, problem 14. We’ll do a slightly different version of the problem this week, finishing out the implementation next week.

**Bank  Accounts Project**

Write a C++ project that defines two classes: **BankAccount** and **SavingsAccount**. **SavingsAccount** derives from **BankAccount** through public inheritance.

**BankAccount:  Attributes and Methods**

**Attributes:**

* Balance
* Number of deposits this month
* Number of withdrawals this month
* Annual Interest Rate: a constant
* Monthly Service Charge: a constant
* Current Month: an enum, JANUARY, FEBRUARY, MARCH ….

**Consider**:

1.What are the data types for each of the attributes?

2.What is their visibility, private or public?

**Methods**

* Constructor: parameterized constructor with arguments for starting balance of account
* **deposit**: a virtual function whose argument is the amount of the deposit; this method should add the deposit amount to the balance and record that a deposit occurred.
* **withdraw**: a virtual function whose argument is the amount of the withdrawal; this method should subtract the amount from the balance and record that a deposit occurred.
* **processMonthEnd:**a virtual function that processes end of month interest and charges. This mothod subtracts the monthly service charges from the balances, calls **calculateInterest**, then updates the current month indicator.
* **calculateInterest:**a virtual function that updates the balance by calculating the interest earned by the account, and adding the interest to the balance. This is performed by the following formulas in the table below.

|  |
| --- |
| **Calculate Interest** |
| Monthly Interest Rate = (Annual Interest Rate / 12) |
| Monthly Interest = Balance \* Monthly Interest Rate |
| Balance = Balance + Monthly Interest |

* **printMonthlyStatement:** a pure virtual method
* **clockTick**: a method that, when invoked, should precipitate the end of month processing (**processMonthEnd** ) of service fees, interest, and so on.

**SavingsAccount: Attributes and Methods**

**SavingsAccount** is a subclass of **BankAccount**.

**Attributes**

* **Status**: indicating if the account is active or not. If the balance of a savings account falls below $25, it becomes inactive. (Consider how to represent this through the attribute **status**.) No more withdrawals may be made until the balance is raised above $25, at which time the account becomes active, again.

**Methods**

* Constructor: parameterized constructor with argument for starting balance of account
* **deposit**: a virtual function that checks if the account is inactive before a deposit is made. If the account is inactive and the deposit brings the account above $25, the account becomes active, again. The deposit is made by calling the super class method.
* **withdraw**: a virtual function that checks if the account is inactive before a withdraw  is made. If the account is inactive, no withdraw is allowed. .Do not allow withdrawals that result in a negative balance.  Invoke the base class to perform the withdrawals.
* **processMonthEnd:**a virtual function that checks the number of withdrawals for the month. If there are more than 4 withdrawals, a service charge of $1 is added for each withdrawal over 4. Don’t forget to check if the balance after service fees falls below the $25 amount, rendering the account inactive. Invoke the superclass method to calculate the end month charges and credits.
* **printMonthlyStatement:** a virtual method that prints out information on the account for the current month. It should only be called by **processMonthEnd**. This method should print out the account balance, number of deposits, number of withdrawals, service fees, and interest. Also, should indicate if the account is active or not.

**Further Considerations**

* There are a lot of constants listed in the class descriptions, e.g., $25 minimum balance, a 4 withdrawal limit without charges, $1 charge for withdrawals over 4 – where should these values be declared?
* How should these values be declared, following good software engineering practice?
* Should we allow a class of type **BankAccount** to be instantiated? Does it make sense?
* Can we invoke **clockTick** on **SavingsAccount** to start the end of month processing?
* How might we demonstrate that our classes are working correctly? How can we test them to be sure they are working correctly?

**Submission**

Top of Form

**Attachment**

No attachment yet

Choose File:

  Don't forget to submit

Bottom of Form