CISC 2010 Midterm Prof. Kadri

Lab Exam

Fall 2016

1. Create a class named ***SavingsAccount***.

2. The class should be defined in its' own header file. This class should have **at least** the following data members and methods.

   Private Data members:
   - customer*, an object of class Name to represent the Customer of this account.*
   - openDate*, an object of class Date to represent the date that the account was opened.*
   - annualInterestRate*, data member to represent the interest currently on the account.* ***This value should be the same for all objects created.***
   - accountNumber*, data member to represent the account number of the account. This field gets its value from a static variable numberOfAccounts (accounts created so far).*
   - savingsBalance*, data member to represent the current balance of the account.*

   Note that this class is composed of several classes that have already been built namely Name and Date. You should not need to make any changes to these classes.

   Member functions:
   - Default Constructor()
     This constructor should initialize the data members *appropriately*.
   - Constructor()
     This constructor should initialize the data members *customer, openDate, and savingsBalance*, to the corresponding values passed to it.

     Note that in both constructors, the data member openDate should be initialized to the todayDate (today's date entered by the user) object. accountNumber should be set to be equal to the current value of numberOfAccounts.

   - getYearlyInterest ()
     This function should calculate *and return* the yearly interest. *Note that the yearly interest is calculated by multiplying the current savings balance by the interest rate,*
     (e.g. savingsBalance * annualInterestRate)*.*

   - Accessor functions:
     getAccountNumber, getAccountDate, getAccountName, getAccountBallance, and the getYearlyInterest..

Organization and style will be taken into account to receive full credit.

3. Write a driver program to test this class. The program should:

   - ✓ Create three SavingsAccount objects. These accounts should have the necessary information as in the displayed table at the bottom.

   - ✓ Create an array of pointers for up to some maximum number of accounts that can be created (say 3). Use a constant for the maximum number of accounts. Populate the array with the three SavingsAccount objects you created.

4. The program should then:

   - ✓ Invoke a local function whose purpose is to output a balance summary for each account created and **release the memory** for that account once it has been displayed.

     Note that this function should take as an argument the array of accounts objects.

Sample Run:

Account	*Date Opened*	*Customer*	*Current Balance*	*Yearly Interest*
1	*3/14/2016*	*Alice Wonderland*	*10000.00*	*500.00*
2	*3/14/2016*	*Snow White*	*1100.00*	*55.00*
3	*3/14/2016*	*Cindy Cinderella*	*100.00*	*5.00*

Organization and style will be taken into account to receive full credit.