

Unit Seven Assignment Two:

Revise your BankAccount class in the following ways...

- 1 - Eliminate the static instance variable *lastAssignedNum*.
- 2 - Add an instance variable *accountType* which will hold either the character *s* (for savings account) or the character *c* (for checking account).
- 3 - Eliminate the default constructor
- 4 - Change the parametric constructor so that it will accept a String account number i.e.(105s or 105c) and a double balance. It should call the mutator method *setAccountType()* which will extract the 's' or 'c' from the account number and place it into the instance variable *accountType*.
- 5 - Change the *calculateInterest()* method so that it adds interest only to accounts with an *accountType* of 's'.

Write a driver class that extends *JFrame*. This class should have the following methods...

BuildList() - fills an array list of *BankAccounts* from the text file *myCreditUnion.txt*.

PrintList() - prints the contents of the array list to a *JTextArea*.

Deposit(String acctNum, double dep) - deposits the value of *dep* into the appropriate *BankAccount* by calling the *deposit* method of the *BankAccount* class.

Withdraw(String acctNum, double with) - withdraws the value of *with* from the appropriate *BankAccount* by calling the *withdraw* method of the *BankAccount* class.

InsertNewAcct(String acctNum, double initdep) - If the account number ends in 'c' it is inserted after the account with the same number but ending in 's'. If the account number ends in 's' it is inserted at the end of the array list.

DeleteDormantAccts() - Eliminates all accounts with a balance of zero from the array list.

CorrectError(String acctNum, double bal) - Finds acctNum in the array list and overwrites it with the value in bal.

ApplyInterest() - Determines and adds the interest earned to all appropriate BankAccounts by calling the calculateInterest method of the BankAccounts class.

FileUpDated() - Prints the newly updated information in the array list to a text file called temp.txt located on the server or on a flash drive.

public static void main(String args[]) - Calls the default constructor of the driver class, sets the size of the JFrame to 500x500, makes the JFrame visible, and ends the application when the JFrame is closed. Use the object created to call the default constructor to call the methods of the JFrame.

public u7a2() (default constructor of the driver class) - Should look exactly like this...

```
public u7a2()  
{  
    BuildList();  
    PrintList();  
  
    Deposit("103s", 500);  
    Withdraw("110s", 304.52);  
    InsertNewAcct("105c", 300);  
    DeleteDormantAccts();  
    CorrectError("107s", 1113.88);  
    ApplyInterest();  
    InsertNewAcct("111s", 100);  
    FileUpDated();  
}
```

All methods except for the default constructor and main should output a string to the JTextArea stating their task has been completed. The JTextArea should be placed in a Container, which is attached to the JFrame. The JTextArea should contain the original array list as well as the strings printed when tasks were completed. Your JFrame should look like the one attached to this assignment. Paste the JFrame into a Word document and turn it in with your code and a print out of temp.txt.