CIS 269 Data Structures Programming Assignment #1

Problem Description

Write a program to process bank accounts. You will write a program called Bank that will read a random access file for account information and populate two arrays: an array of Strings to hold account numbers, and a parallel array of doubles to hold account balances. The program will then provide a menu and accept various inputs from the user.

An initial version of the data file is provided to you (bank.dat), and has the structure noted below. When the Exit option is chosen, your program should write overwrite the contents of the file with the current contents of the arrays before exiting.

Requirements

1. You may use any classes you like, or you may put all your code in one source file.
2. Your program should compile and run correctly with the same inputs and outputs as given in the sample run.
3. Read the random access file to populate the two arrays. Your code should be able to handle ANY SIZE file. The test file you are given contains 3 accounts, but I will use files of varying lengths to test your program. (Hint: Use array sizes of 3 until you get the logic of your program working, then figure out how you will handle files that contain more or less than 3 accounts.)
4. Your program must display the menu, prompts and output as shown below.
5. When Deposit or Withdrawal is chosen, prompt the user for the account number, and perform a simple linear search on the array of accounts for that number. If the account number is not found, display a message to the user. If the account number is found, update the parallel balance array appropriately.
6. The menu option to display all accounts lists the account number and corresponding balance. Use the format shown below. Balances must be displayed with a dollar sign and 2 decimal places.
7. The menu option to calculate daily interest should calculate 3% interest for each account and add the interest to the account's balance.

IMPORTANT NOTE! – READ THIS!

You will FAIL this assignment if you do not use random access file processing or if you do not use TWO simple, single-value parallel arrays as noted above. The algorithm for this assignment is MEANT to be inefficient, so do not attempt to improve upon it by using other data structure libraries. The rest of the programming assignments in this class will improve upon this one. We can better appreciate beauty after we've seen ugly, yes? This assignment is meant to be ugly. Still use the **best practices** you've learned for Java or C++ programming, however. Let's not write ugly code on top of an ugly algorithm!

Bank.dat Structure:

AccountNumber (as a String)
Balance (as a double)
AccountNumber (as a String)
Balance (as a double)

AccountNumber (as a String)
Balance (as a double)

Sample Run

(Note: user input is in red color)

```
Enter one of the following
1) Deposit
2) Withdrawal
3) Display All Accounts
4) Calculate Daily Interest
5) Exit

3
*********************************
Account C123 has balance $12.33
Account S456 has balance $45.66
Account S789 has balance $78.99
*********************************
1
Account number: C123
Amount of deposit: 50
3
*********************************
Account C123 has balance $62.33
Account S456 has balance $45.66
Account S789 has balance $78.99
*********************************
2
Account number: S456
Amount of withdrawal: 40
3
*********************************
Account C123 has balance $62.33
Account S456 has balance $5.66
Account S789 has balance $78.99
*********************************
4
Account C123 has balance $64.20
Account S456 has balance $5.83
Account S789 has balance $81.36
3
*********************************
Account C123 has balance $64.20
Account S456 has balance $5.83
Account S789 has balance $81.36
```

****

5