Lab 03 Specification – Object Oriented Programming
Due (via Bitbucket) Wednesday, 20 September 2017
50 points

Lab Goals

- To produce a program with a real world application (ATM Simulator)
- To gain experience with Object Oriented Programming techniques such as Classes, Objects, Abstraction, Encapsulation, and Modularity.

Assignment Details

Implement an ATM Simulator with the following core functionalities. You must use at least one abstract class and one interface.

Customer Login Module (20 points)

Customer Login: This functionality is used to create a customer login module, where the customer has the ability to sign up for a new account or login to their existing account. Note: In order to create a new account, the customer id provided by the user during sign up needs to match with one of the customer ids listed in `accounts.txt` with an initial balance. In this lab, we provided you with an initial `accounts.txt` file which was created by a PNC bank teller – the customer accounts that you create while running and testing your program should each use a distinct customer id from this file.

Your code should prompt the user and ask if they are a new or existing user. If it is a new user, then display a sign up prompt, which should capture the customer id, customer name, customer SSN, creation date time (system generated), and login credentials (user id and password) from the user. If the user provided customer id matches with one of the customer ids in `accounts.txt` file, then store the customer record into a new line in the `customers.txt` file, otherwise display a ”failed” message.

If it is an existing user, display a login prompt and capture their login credentials. Then evaluate the user provided credentials against the user id and password stored in the (`customers.txt`) file. Once the user is successfully validated then display a welcome message with their customer name and current Date Time.

Transaction Module (30 points)

Transaction: After successfully executing the login process, a bank customer should be taken to a display that prompts them with a menu such as this:

```
1) Display current balance
2) Deposit Transaction
3) Withdraw Transaction.
4) Exit from ATM.
```

If the customer selects option 1, then display the customers current balance (Hint: read from `accounts.txt`)

If the customer selects option 2, then prompt the customer to provide a deposit amount. Add this amount to the customer's balance. (Hint: update `accounts.txt`)
If the customer selects option 3, then prompt the customer to provide a withdrawal amount. Subtract this amount from the customer's balance. (Hint: update `accounts.txt`)
After every transaction, display the menu again for customer input.
A sample Java program called `FileUtility.java` that can read, write, and replace text in a file is provided in the bitbucket repository. Please include that class in your lab to allow you to access the `accounts.txt` and `customers.txt` files. Below you can see a few example for using the methods of `FileUtility`.

```
//create a FileUtility that uses a file called "accounts.txt"
FileUtility accounts = new FileUtility("accounts.txt");

//read the first line from "acounts.txt" and store it
String first_line = accounts.read();

//read the second line from "acounts.txt" and store it
String second_line = accounts.read();

//append a line to the end of "accounts.txt"
accounts.write("Thus ends this file");

//update the first line to be "This is now the first line"
accounts.update(0, "This is now the first line");

//get how many lines there are in the file and store that
int lines = accounts.size();

//update the last line to be "This is now the last line"
accounts.update(lines-1, "This is now the last line");

//reset the reader to the beginning of the file
accounts.reset();

//read the first line again, and print it
String line = accounts.read();
System.out.println(line);
```

Submission Details

For this assignment, please submit the following to your `cs112f2017-<your user name>` repository (and ensure that the instructor has access to your repository):

1. Commented source code from the "Customer Login Module" program.
2. Commented source code from the "Transaction Module" program.
3. You are required to use an abstract class and an interface in your lab assignment. Failure to do so will result in only partial credit.