Noel LeJeune e-mail: lejeunen@mscd.edu SI 1052

# Assignment 02: Simple ATM

## *Performance Measures*

1. **Demonstrate that you can implement Java classes**
2. **Demonstrate that you can create and use UML Class Diagrams to design a solution to a problem**
3. **Demonstrate that you can implement program logic (algorithms)**
4. **Demonstrate that you can use iteration (loops)**
5. **Demonstrate that you can use conditional statements**

## *Description (Requirements)*

Create a Java program that simulates a simple ATM (Automated Teller Machine).

The simple ATM should allow the user to enter commands at the keyboard. The user, acting as the customer, then selects from a menu to do the following:

- Get the customer's account balance (display all the information about the account)
- Deposit money to the customer's account (enter an amount at the keyboard and add that to amount to the customer's account balance)
- Withdraw money from the customer's account (enter the amount to be withdrawn and subtract that amount from the customer's account balance, provided there are sufficient funds)

## *Constraints*

- The first iteration of this program will have only one customer with one account.
- When the program starts, the Simple ATM creates an account for a customer. The program assigns the account number and a message is displayed telling the customer their account number.
- The menu of user choices is displayed giving the user the opportunity to select an action. Once a choice is entered, the action is performed and the menu of choices reappears. The cycle is repeated until the user selects the choice to quit the program.
- Deposits have a fee (a constant value) of $2 if the deposit amount is less that $1000.
- Withdrawals have a fee (a constant value) of $5. Note that an account is not allowed to have a negative value so the withdrawal amount plus the fee must be less than or equal to the balance (a 0.0 balance is allowed). The withdrawal fee is waived if the customer's account is a valued account.
- Deposits and withdrawal amounts of 0.0 or less are ignored (No fees, no changes in balance)
- A valued account is one with a balance of at least $10,000.
- You must provide informative messages when depositing and withdrawing (this allows you to verify that your program is correct).

## *Notes:*

We will work on the high-level design in class. You will be responsible for the complete program (creating all necessary classes). You may use any of the in-class demonstration and discussion as a basis for you're your program.

Use an incremental development process. Complete the program without imposing any conditions on deposits and withdrawals. Then, when the core functionality is working, add the conditions on deposits and withdrawals (also in an incremental approach).

## *Submit - Use the "Assignments Drop Box" in Blackboard - do NOT submit via email*

Create a ***zip file*** containing all the required files *Name the file with your name, e.g.* **`SueJones02.zip`**

***Submit ALL files in one zip file consisting of:***

1. UML Class Diagrams for your classes
2. Your properly documented **source** code– ***do not*** submit the byte code files (i.e. .class files)
3. A completed run of the test cases (with execution results)
4. A sample of the output from running your program (`Output02.txt`)
5. The answers to the "Standard Questions" (`StandardQuestions02.txt`)

## *Standard Questions*

A. How much time did you spend on this assignment?

B. What difficulties did you experience?

C. What were the results of running your test cases/program? Run the program enough times to see that it works correctly – more about testing in subsequent assignments.