CSC/CIS 276
Problem Solving and Programming II in Java
Assignment 1 due Jan 29, 2014 – 6%

Objective of this Assignment:

This is your first Java programming assignment. The purpose is to synthesize the introductory object-oriented programming concepts in Chapters 1 to 5 that include the C++ concepts learned in CSC/CIS 175. You are expected to demonstrate a working knowledge of UML class diagrams, class definition, variable declarations, standard input and output, use of data types and expressions. You should use an editor of your choice to write the source code and to compile and run your application using a command line application such as command prompt in Windows or terminal app in Mac.

Problem

A financial institution has approached you to build an application for them. This application will allow the user to login a bank account and to do standard transactions on the account.

Deliverables

1) Create a class diagram (i.e. showing attributes and their data types, methods with parameters and return type in appropriate data types) for your system design to meet the following requirements.

2) Create an application that meets the following requirements:

   Login process: The user is asked to enter the bank account (an integer 123) and the password (an integer 888). If the user entered an incorrect password, the user should be told so and asked to reenter the password. The user may have three chances and a message to advise the user to contact the bank will be displayed after three incorrect password entries, and the program will quit.

   Transaction process: The application should offer transactions on three account types: saving, checking, and retirement accounts, and each should be initialized to $1000. Once the user has successfully log in, a menu consisting of the following choices will be displayed and the user will be asked for input.

   - *quit* - when the user enters this command, print the information of each account to the screen in some legible and meaningful format and exit the program.
   - *deposit* <*account*> <*amount*> – Deposit the requested amount (which you may assume is a floating point number). Afterwards print the information of the updated account to the screen in some legible and meaningful manner.
   - *withdraw* <*account*> <*amount*> - Withdraw the requested amount (which you may assume is a floating point number) from the account specified by the account number. Afterwards print the information of the updated account to the screen in some legible and meaningful manner.

- *transfer* <*from-account*> <*to-account*> <*amount*> - Transfer the requested amount from the specified account to the specified account. Afterwards print the information of both updated accounts to the screen.

For withdrawals and transfers, if the balance in the (from-)account is less than the amount to withdraw or transfer, a message should be displayed and the transaction will not be completed. The user should be allowed to perform these transactions as long as they choose to.

Submission

Your assignment must be submitted on Blackboard. You are required to submit a single zip file called A1-LastnameFirstname.zip (replace LastName with your last name and FirstName with your first name, e.g., my submission will be called A1-TangCharlotte.zip) containing:

a) the class diagram
b) the source file (.java file)
c) the compiled file (.class file)

For generating .zip file, you may use the following free software:
http://www.7-zip.org/download.html

Collaboration and Plagiarism

For more details, please refer to ACADEMIC INTEGRITY in the course schedule or http://www.umflint.edu/departments/catalog/