**CMIS242 Rubric Checklist for**

**Project 2**

***GUI Class & main () method (35 points)***

* Contains the main () method as last entry in the program. It then calls a method that reads in the values of a file containing account data. It also declares an instance of the GUI class and calls that class and its methods.
* In addition to the main () method, there is inserted into the program (above the main () method the following:
  + The GUI class that contains as elements:
    - * Two text fields: (1) one for the account number with a label and (2) one for money involved in the transaction with a label. [Different than the assignment text.]
  + Use JOptionPane to display all results of transactions and errors.
  + Contains event handlers to handle each of the four buttons
  + Contains Withdrawal checks to ensure the value in the text field is numeric.
  + Contains Withdrawal checks to ensure the amount is in increments of $20.
  + Provides ability to attempt to withdraw the funds is made from the account selected by the radio buttons.
  + An exception is thrown for insufficient funds, or if value is not numeric or is value is not in $20 increment using a JOptionPane window explaining the error.
  + Upon successful withdrawal, a window is displayed confirming that the withdrawal has succeeded.
  + Provides ability to attempt Deposit when Deposit button is clicked.
  + Contains Deposit checks to ensure the value in the text field is numeric.
  + Contains Transfer button functionality providing transferring funds to the selected account from the other account.
  + Contains transfer checks to confirm that the amount supplied is numeric and that there are sufficient funds in the account from which the funds are being transferred.
  + Contains a Balance button will cause a JOptionPane window to be displayed showing the current balance in the selected account.
  + The main class contains two Account objects, one for the checking account and another for the savings account.

***Account Class (25 points)***

* Contains a constructor plus a method that corresponds to each of the four buttons in the GUI.
  + A balance method. Searches for account number and type displays amount or no account.
  + A Deposit method. Adds to the balance and displays new total.
  + A Withdrawal method. Subtracts from the balance and displays the new total.
  + A Transfer method. Displays the decrease from the account in the window and displays (second label) the increase to the account from the other account. [Assume there is a savings and checking account for each account number.]
* Checks that there must be sufficient funds to pay for service charge. (The method that performs the withdrawals throws an InsufficientFunds exception whenever an attempt is made to withdraw more funds than are available in the account.)
* Incorporates logic to deduct a service charge of $1.50 when more than four total withdrawals are made from either account.

***Insufficient Funds class (20 points) [This class should be in the Account class.]***

* Is a user defined checked exception class
* Handles all user-defined exceptions.

***Test Cases (10 points)***

* Test cases are supplied in the form of table with columns indicating the input values, expected output, actual output and if the test case passed or failed.
* Enough scenarios selected to completely test the program.
* Test cases were included in the supporting word or PDF documentation.

***Documentation and Style Guide (10 points)***

* Screen captures were provided and labeled for compiling your code, and running each of your test cases.
* Header comments include filename, author, date and brief purpose of the program.
* In-line comments used to describe major functionality of the code.
* Meaningful variable names and prompts applied.
* Class names are written in UpperCamelCase.
* Variable names are written in lowerCamelCase.
* Constant names are in written in All Capitals.
* Braces use K&R style.
* Declare all instance variables private.
* Avoids the duplication of code.