ENGR 211 - Introduction to Programming

Mini Project 1

December 5, 2016

(Due: by 5 pm on Dec. 19, 2016)

In this mini project, you are going to develop a software for a commercial bank. This software will be used by the customers and employees of the bank to manage the bank accounts (i.e., create, delete, etc.) and perform various money transactions (deposit, transfer, etc.). The detailed explanations are provided below. The primary goal of this mini project is to make you practice the data structures. Hence, please try to take very opportunity to use a data structure in your code. At various points, you will be forced to use particular data structure as explained in the "Implementation Notes" section at the end.

Text-based User Interface:

When your program first runs, it will ask the user to log in by providing a user name and password as follows. There should be a default user that always exist with user name 'manager' and password 'manager123'. This account will initially be used to create or load other accounts. The following shows the interface for initial login screen with a scenario where the user first enters incorrect credentials, and then successfully logs in during the second trial. As long as the user enters invalid credentials, the tool should ask the user to try again as shown below.

```
Welcome to Banking Account Management Tool (v. 1.0)
Please log in by providing your user credentials:
       User Name: Ayse
       Password: 123
Your user name and/or password is not correct. Please try again!
       User Name: Ayse
       Password: 12345
```

Once the user enters valid credentials, your program should greet the user with the following message, and provide a menu of the things to do. User will do a selection by entering the corresponding menu number, and accordingly different information will be shown and/or requested. If the user enters an invalid menu entry, the program should warn the user to provide a valid menu number as demonstrated below.

```
Welcome, Ayse! Please choose one of the following options by entering the corresponding menu number.
   1. Check Balance
   2. Transfer Money
   3. Deposit
   4. Withdraw
   5. Close Account
```

```
6. Update Password
7. See the Latest Transactions
8. Admin Menu
9. Quit

Please make your selection: 11

11 is not a valid entry. Please choose from the above menu.
Please make your selection: 1
```

During any operation below, at any point where an input is requested from the user, if the user types a secret code, '*', the program should abort the current operation and display the main program menu to the user. This is important to get out of the current step with minimal effort. Please make sure that you implement this check at each input point at any operation below.

When menu item 1 is chosen, the current balance of the customer will be shown as follows. Then, the user should be given options to go to the main menu (1) or quit (2). In case of invalid choice number, the user should be warned and requested to try again.

```
===== Check Balance =====
   You have 100 TL in your account.

   What do you want to do next?
       1: Go to the main menu
       2: Quit
   Your choice: 1
```

When menu item 2 is chosen, the user will be asked to provide the recipient name. The recipient should be an account holder. If the provided name does not match any existing account holders, then the user should be warned and requested to try again. Next, transfer amount should be requested. Then, the user's confirmation should be obtained as demonstrated below. Then, the balances of sender and recipient accounts should be updated in the internal data structure(s). Next, the user should be informed about the completed transfer process. The transaction should be recorded in proper internal data structure(s) to show it to the user when requested.

```
===== Transfer Money =====
   Please enter the recipient: AAA

   AAA is not a valid recipient. Please try again.

   Please enter the recipient: Veli
   Please enter the amount: 80
```

```
80 TL will be transferred to Veli today.

Do you approve (yes/no?): yes


80 TL has been transferred to Veli. Your current balance
is 20 TL.
```

In the above confirmation step, if the user changes his/her mind, and do not approve the transfer, then the transfer should be cancelled with a proper information message as shown below.

```
Do you approve (yes/no?): no

Your transfer request has been canceled.
```

Finally, the user should be given options to go to the main menu (1) or quit (2).

```
What do you want to do next?

    1: Go to the main menu

    2: Quit

Your choice: 1
```

When menu item 3 is chosen, the user will be asked to provide the deposit amount. Then, the user account balance should be updated in the internal data structure(s). Next, the user should be informed about the completed deposit process as follows. The transaction should be recorded in proper internal data structure(s) to show it to the user when requested.

```
===== Deposit Money =====

   Please enter the amount: 80

   80 TL has been deposited to your account. Your current
   balance is 180 TL.
```

Finally, the user should be given options to go to the main menu (1) or quit (2).

```
What do you want to do next?

    1: Go to the main menu

    2: Quit

Your choice: 1
```

When menu item 4 is chosen, the user will be asked to provide the amount to be withdrawn. Then, the user account balance should be updated in the internal data structure(s). Next, the user should be informed about the completed withdrawal process as follows. The transaction should be recorded in proper internal data structure(s) to show it to the user when requested.

```
===== Withdraw Money =====

   Please enter the amount: 80
```

```
80 TL has been withdrawn from your account. Your current
balance is 20 TL.
```

Finally, the user should be given options to go to the main menu (1) or quit (2).

```
What do you want to do next?
    1: Go to the main menu
    2: Quit
Your choice: 1
```

When menu item 5 is chosen, the program should first check if the user has non-zero balance. If that is the case, then the user should not be allowed to close his/her account, and an informative message explaining the situation should be shown as follows.

```
===== Close Your Account =====
   Sorry, we cannot close your account at this point, as you
   still have some balance in your account. You should
   withdraw this balance before closing your account.
```

If the user has zero balance, then, the user's confirmation should be obtained as demonstrated below. If the user approves, then the user account should be closed and removed from the account list in internal data structure(s). Next, the user should be informed about the completed transfer process.

```
===== Close Your Account =====
   Do you approve (yes/no?): yes
   Your account has been closed now, and your session has
   ended.
   Thanks for being our customer.
```

If the user does not approve, then the account closing process should be canceled, followed by displaying a proper message to the user as shown below.

```
Do you approve (yes/no?): no
Your account closing request has been canceled.
```

Finally, the user should be given options to go to the main menu (1) or quit (2).

```
What do you want to do next?
    1: Go to the main menu
    2: Quit
Your choice: 1
```

When menu item 6 is chosen, the user will be asked to provide his/her current password. If the provided password is not correct, the user should be warned, and asked to re-enter his/her password. Then, the user should provide the new password, and then confirm the password by re-entering it again. If the two passwords do not match, the user should be warned and asked to re-enter the old and new passwords again as shown below. Then, the user password should be updated in the internal data structure(s). Next, the user should be informed about the completed transfer process.

```
===== Update Password =====
   Please enter your current password: 12345
   Please enter your new password: abcd123
   Please re-enter your new password abcd12
   Sorry, your new password entries do not match! Please try
   again!

   Please enter your current password: 12345
   Please enter your new password: abcd123
   Please re-enter your new password abcd123

   Your password has been successfully updated.
```

Finally, the user should be given options to go to the main menu (1) or quit (2).

```
What do you want to do next?
    1: Go to the main menu
    2: Quit
Your choice: 1
```

When menu item 7 is chosen, the user will be shown the latest N transactions (at least deposit, withdrawal, and transfer operations should be included) starting from the most recent one. The user should be asked to provide the value of N. For each transaction, the type of the operation, amount, and the balance after the operation should be shown. Also, for transfers, the recipient info should be included as well. A sample run is shown below.

```
===== See the Latest Transactions =====
   Here, you may view your latest transactions starting from
   the most recent one.
   Please enter how many transactions you want to see: 3
     1. Deposit: 200 TL, Balance: 500 TL
     2. Withdrawal: 100 TL, Balance: 300 TL
     3. Transfer: 50 TL, Recipient: Ali, Balance: 400 TL
```

Finally, the user should be given options to go to the main menu (1) or quit (2).

```
What do you want to do next?
    1: Go to the main menu
    2: Quit
Your choice: 1
```

When menu item 8 is chosen, the user will be asked to provide admin credentials. If the provided password is not correct, the user should be warned, and asked to re-enter his/her password. For this mini project, you will assume that the credentials are as follows: user name = admin, password = 123abc. If the provided credentials are correct, then the user should be greeted and shown a special admin menu as demonstrated below.

```
===== Admin Operations =====
   Please provide admin credentials:
          User Name: admin
          Password: 123abc
   Your user name and/or password is not correct. Please try
   again!
          User Name: admin
          Password: abc123
   Welcome! Please choose one of the following options by
   entering the corresponding menu number.

       1. Load Customer Account Data from a File
       2. Create Account
       3. Close Account
       4. Search for an Account
       5. See the Stats
       6. Go to the Main Menu
```

When admin menu item 1 is chosen, the customer accounts data should be read and loaded from a local file into the internal data structure(s), and the total number of loaded accounts should be shown afterwards as shown below.

```
===== Admin: Load Customer Account Data =====
Loading...
The account data for 24 customers have been loaded.
```

You may assume that the customer accounts file is located under the same directory where your code file is saved, and its name is 'customer_accounts.txt'. The file content will be as shown below where each line will contain a separate account's information. On each line, account holder's name, password, and the current balance info will be given with *** between each field.

```
Mehmet Tanar***mhmt123***200
Ali Kara***adf542***100
```

```
Ahmet Ozturk***23asd***400
```

You can read from a file by slightly modifying the following piece of code. In order to learn about the basics of reading from a file, please check this link.

```
with open("customer_accounts.txt") as f:
    for line in f:
        print line,
```

Finally, the user should be given options to go to the admin menu (1) or quit (2).

```
What do you want to do next?
      1: Go to the admin menu
      2: Quit
  Your choice: 1
```

When admin menu item 2 is chosen, the user will be asked to provide account holder name, password, and opening balance. If the provided account holder name exists, then the operation should be canceled with proper information message provided to the user as shown below. Otherwise, the account should be created and inserted into proper internal data structure(s). The user should be shown an informative message regarding the just completed process.

```
===== Admin: Create Account =====
  Please enter account holder name: Sami
  Please create a password for Sami: ab12
  Opening balance: 50


  An account has been created for Sami with starting
  balance of 50
```

Finally, the user should be given options to go to the admin menu (1) or quit (2).

```
What do you want to do next?
      1: Go to the admin menu
      2: Quit
  Your choice: 1
```

When admin menu item 3 is chosen, the user will be asked to provide account holder name. If the provided account holder name does not exist, then the user should be warned and requested to provide a valid account holder name. Once a valid account holder name is provided, a final conformation should be sought from the user. If the user approves, the account should be closed, and removed from the internal data structures appropriately. Otherwise, the operation should be canceled. At the end, a proper information message should be provided to the user as shown below.

```
===== Admin: Close Account =====
  Please enter account holder name: Mehmet


  There is no available account for this account holder.
  You may try again with another name.


  Please enter account holder name: Sami


  The account for customer Sami will be closed.
  Do you approve (yes/no?): yes


  The account for customer Mehmet has been closed.
```

The following use case demonstrates the scenario where the user does not approve the account closing.

```
  Do you approve (yes/no?): no
  Your account closing request has been canceled.
```

Finally, the user should be given options to go to the admin menu (1) or quit (2).

```
  What do you want to do next?
      1: Go to the admin menu
      2: Quit
  Your choice: 1
```

When admin menu item 4 is chosen, the user will be asked to provide account holder name. If the provided account holder name does not exist, then the user should be warned and requested to provide a valid account holder name. Once a valid account holder name is provided, then the account information should be shown as follows.

```
===== Admin: Search for an Account =====
  Please enter account holder name: Sami


  There is no available account for this account holder.
  You may try again with another name.


  Please enter account holder name: Mehmet
  Account Holder: Mehmet
```

```
Current Balance: 100 TL
```

Finally, the user should be given options to go to the admin menu (1) or quit (2).

```
What do you want to do next?
    1: Go to the admin menu
    2: Quit
Your choice: 1
```

When admin menu item 5 is chosen, the admin will be shown some stats regarding the current set of existing bank accounts as shown below. Here, the average account balance and total balance should be rounded to show two digits after the dot.

```
===== Admin: See the Stats =====
  Total Number of Accounts: 980
  Number of Accounts with Non-Zero Balance: 240
  Total Balance: 32467 TL
  Average Balance: 33.13 TL
```

Finally, the user should be given options to go to the admin menu (1) or quit (2).

```
What do you want to do next?
    1: Go to the admin menu
    2: Quit
Your choice: 1
```

When admin menu item 6 is chosen, the user should be shown the program main menu. In the program main menu, when menu item 9 is chosen, the program should terminate.

Implementation Notes:

- A dictionary must be used to keep user account information.
- A list of tuples structure must be used to keep the past user transactions. It may be either part of the above dictionary, or a separate transactions dictionary.
- Whenever an operation is performed by the user, the above data structures should be updated properly to reflect the changes.

Warnings:

- **Do not** talk to your classmates on project topics when you are implementing your projects. **Do not** show or email your code to others. If you need help, talk to your TAs or myself, not to your classmates. If somebody asks you for help, explain them the lecture slides, but do not explain any project related topic or solution. Any similarity in your source codes will have **serious** consequences for both parties.

- Carefully read the project document, and pay special attention to sentences that involve "**should**", "**should not**", "**do not**", and other underlined/bold font statements.

- If you use code from a resource (web site, book, etc.), make sure that you reference those resource at the top of your source code file in the form of comments. You should give details of which part of your code is from what resource. Failing to do so **may result in** plagiarism investigation.

- Even if you work as a group of two students, each member of the team should know every line of the code well. Hence, it is **important** to understand all the details in your submitted code. You may be interviewed about any part of your code.

How and when do I submit my project? :

- Projects may be done individually or as a small group of two students (doing it individually is recommended for best learning experience). If you are doing it as a group, only **one** of the members should submit the project. File name will tell us group members (Please see the next item for details).
- Submit your own code in a **single** Python file. **Do not** submit image files. Your implementation should assume that label images are located under a directory named as "assets" in your current working directory. We will test your code file under this setting. Name your code file with your and your partner's first and last names (see below for naming).
    - If your team members are Deniz Barış and Ahmet Çalışkan, then name your code file as deniz_baris_ahmet_caliskan.py (Do **not** use any Turkish characters in file name).
    - If you are doing the project alone, then name it with your name and last name similar to the above naming scheme.
    - Those who **do not** follow the above naming conventions **will get 5 pts off** of their grade.
    - Submit it online on LMS (Go to the Assignments Tab) by **5 pm on Monday, Dec. 19, 2016**

      Late Submission Policy:

        - -10%: Submissions between 17:01 – 18:00 on the due date
        - -20%: Submissions between 18:01 – midnight (00:00) on the due date
        - -30%: Submissions which are 24 hour late.
        - -50%: Submissions which are 48 hours late.
        - Submission more than 48 hours late will not be accepted.

Grading Criteria? :

Code Organization			Functionality
Meaningful variable names (3 pts)	**Proper use of functions, compact code with no unnecessary repetitions (4 pts)**	**Sufficient commenting (4 pts)**	**There are 13 menu items in total (except quit). Proper implementation of each of these menu items will be 7 points.**

- Interview evaluation

- Your grade from interview will be between 0 and 1, and it will be used as a coefficient to compute your final grade. For instance, if your initial grade was 80 before the interview, and your interview grade is 0.5, then your final grade will be 80*0.5 = 40. Not showing up for the interview appointment will **result in** grade 0.

Have further questions?:
Please contact your TAs if you have further questions. If you need help with anything, please use the office hours of your TAs and the instructor to get help. **Do not walk in randomly (especially on the last day) into your TAs' or the instructor's offices. Make an appointment first. This is important. Your TAs have other responsibilities. Please respect their personal schedules!S**

Good Luck!