CMPS 150
Spring 2017
Programming Assignment #8, part 2
Date Assigned: Monday, April 10, 2017
Due Date: 11:55 PM, Tuesday, April 25, 2017

Objectives:
- selection & repetition, functions, object-oriented programming

1) Include the following information as comments at the beginning of your source code. Name it **`pa8x.py`**
BE SURE it ***lines up*** nicely as you see it below.

```
# Author:          Type-Your-Name
# CLID:            Type-Your-CLID
# Course/Section:  CMPS 150 – Lecture Section # ___
# Assignment:      pa8x
# Date Assigned:   Monday, April 10, 2017
# Date/Time Due:   Tuesday, April 25, 2017 -- 11:55 pm
#
# Description:     This program processes bank transactions using classes/objects.
#
# Certification of Authenticity:
# I certify that this assignment is entirely my own work.
```

2) Program Description

Write a program to processes a file of bank transactions. There is an unknown number of sets of bank transactions in an input file. Ask the user for the name of the input file and use this string variable in your open statement.
For example:

```
filename = input("What is the name of your input file? ")
infile = open(filename,"r")
```

There is a sample input file ("trans.py") on the class Moodle site. Each read of data will consist of three(3) pieces/lines of data. The first line is a bank account number. The second line is a transaction type (single letter, either 'W', 'D' or 'B') for withdrawal, deposit or balance inquiry. The third line is the amount of the transaction. For balance inquiries, the amount will always be zero.

Stop reading from the file (stop processing) when you read a bank account number of 'X'.

Before processing data in the file, get the beginning balance and bank account number for two(2) bank accounts from the keyboard.

3) Class/Ojbect Requirements

Change the initializer from PA8, part1 to have four(4) data fields. All must be private. They are:
`__acctNum, __balance, __totalDeposits, __totalWithdrawals`

Also change the initializer to be passed only two(2) parameters: They are: **`acctNumIn`** and **`balanceIn`**

These parameters will set the **`acctNum`** and **`balance`** data fields. The **`totalDeposits`** and **`totalWithdrawals`** fields will always be initialized to zero(0).

Write "Get" functions for all data fields.

Finally, write Deposit, Withdrawal and Balance methods.

The Deposit method is passed one parameter, which is the amount to be deposited. It should increase both the balance data field and the totalDeposits data field by this amount. Print an appropriate transaction line. No return value.

The Withdrawal method is passed one parameter, which is the amount to be withdrawn. If the balance field is equal to or larger than the parameter passed in, this method should decrease the balance data field and increase the totalWithdrawals data field by this amount. Print an appropriate transaction line. No return value.

The Balance method is passed no parameters. It simply prints a "balance" information line. No return value.

With the information read from the keyboard, create two(2) bank account objects with the given balance and bank account numbers. The creation of an object looks similar to the following code (see main given on Moodle site):

```
account1 = Account(acctNum1,balance1)
```

When processing the file of transaction information, you must determine which bank account object must be "affected" by the transaction and then "apply" the appropriate class method.

sample input file

```
ABC123
W
48.75
DEF456
D
45.00
ABC123
W
20.00
ABC123
B
0
DEF456
W
80.50
DEF456
W
900.00
ABC123
D
125.00
DEF456
D
100
ABC123
W
550.5
DEF456
W
135.90
ABC123
B
0
ABC123
W
72.80
DEF456
D
75.00
X
-99
-99
```

4) Sample Run

```
Enter bank account ID #1: ABC123
Enter balance for bank account #1: 200
Enter bank account ID #2: DEF456
Enter balance for bank account #2: 500

What is the name of your input file:  trans.py


Acct          Trans Type        Amount       Balance
-------------------------------------------------------
ABC123         Withdrawal          48.75        151.25
DEF456         Deposit             45.00        545.00
ABC123         Withdrawal          20.00        131.25
ABC123         Balance                          131.25
DEF456         Withdrawal          80.50        464.50
DEF456         Withdrawal         900.00      <denied>
ABC123         Deposit            125.00        256.25
DEF456         Deposit            100.00        564.50
ABC123         Withdrawal         550.50      <denied>
DEF456         Withdrawal         135.90        428.60
ABC123         Balance                          256.25
ABC123         Withdrawal          72.80        183.45
DEF456         Deposit             75.00        503.60
-------------------------------------------------------

ABC123, Deposits:               125.00
ABC123, Withdrawals:            141.55
ABC123, Ending Balance:         183.45

DEF456, Deposits:               220.00
DEF456, Withdrawals:            216.40
DEF456, Ending Balance:         503.60
```

5) Upload to Moodle

Get in a browser and login to Moodle.
Instead of going to the Lecture Section, go to your UPLOAD section on the Moodle site.
Click on the link for Programming Assignment #8, Part2.
Select to "Add a Submission" then "Upload a File"
Select to "Choose a File" and go about the process of browsing/finding "**`pa8x.py`**" on the computer.
Select to "Upload this File"**When returned to the Upload screen, MAKE SURE to click on the "Save Changes" button.**
You will be returned to the "Programming Assignment #8, Part2" screen.
This time you should see your source code file listed on it.

6) Logout of Moodle

You can turn in programs
up to 24 hours late for a maximum of 75% credit
or up to 48 hours late for a maximum of 50% credit