CS160 Computer Science I
Program 7

Objectives
Work with files
Work with functions

Assignment
This program will display a very basic handling of transactions for a checking account. This program will read in a data file with all the past transactions. Use `FileUtil.selectOpenFile` or `input()` to ask for a file name from the user. The format of the file will be very basic: a single letter which indicates the type of transaction ('B' for beginning balance, 'C' for a check, 'D' for a deposit, or 'W' for a withdrawal), a date (as a string in the format of MM/DD), and an amount (floating point value). The transaction type may be upper or lower case. Using the appropriate string methods which make this a non-issue. The first line of the data MUST BE the beginning balance. The remaining lines may be for any type of transaction other than specifying the beginning balance. All lines will have three values, separated by commas, and in the order specified. A sample data file might be:

```
B,03/01,1000.00
C 03/02,100.00
C,03/03,9.00
W,03/04,20.00
C,03/05,50.00
```

You do not need to write a program to create the data file, you can use any text editor. You can write a program to create the file if you want, but you don't need to turn it, or more to the point, don't turn it in.

After reading each line, break it up into its individual pieces of information. The `split()` method would likely be the easiest way to do so, but you can use the `find()` or `index()` methods as well.

Write the information from the data file to the display in a neat, column-aligned format with an updated balance on each line. Expand the single letter for the transaction type into complete text, as demonstrated below. Calculate the new balance from the information provided in each line, adding the amount to the balance for a deposit and subtracting the amount from the balance for checks and withdrawals. Make sure each dollar amount has *exactly* 2 places after the decimal point. You do not need to do anything with the date read from the file, just print it out as it was read from the file. Also include today's date at the top of the table. We accessed todays' date in class in the CalcAge example.

This is a very limited program, so there are limited opportunities to write functions, but one function that must be written is:
`def transactionType (transLetter):`

This function accepts as an argument the transaction letter (“B”, “C”, “D”, or “W”) and returns the full transaction type (“Beginning Balance”, “Check”, “Deposit”, or “Withdrawal”).

All of the code for the main program must be a function named `main()`. The only line of executable code that should not be in a function is to call `main()` as the last line of the program.

A sample of the output, using the data from above, might be:

```
Checkbook balance as of 03/06

03/01  Beginning Balance                        1000.00
03/02  Check                          100.00     900.00
03/03  Check                            9.00     891.00
03/04  Withdrawal                      20.00     871.00
03/05  Check                           50.00     821.00
```