CS150A & CS150B Introduction to Computer Science (Zaring)
Fall 2018
Assignment 6
Due by 4:30pm on Wednesday, Nov. 21
NO LATE SUBMISSIONS ACCEPTED FOR THIS ASSIGNMENT

Description:
Programming in Wing 101 and using your class notes, class lecture materials, and the material in Chapters 1-10 of *M&R*, write a simple image-library program.

Initially, the program starts with an empty library. The program then repeatedly prompts the user to enter commands through which to manipulate the library. Your program must handle the following commands:

- `quit`
  Stops the program.
- `list`
  Displays on the screen the numbers, file names, and titles of all the images currently in the library.
- `showall`
  Displays all the images currently in the library, one after another, moving on to the next image after the currently-displayed image is clicked on
- `show`
  Displays one of the images currently in the library after prompting the user to enter the number of the image they'd like to show. If the number the user enters doesn't correspond to an image currently in the library, an error message is printed out.
- `add`
  Adds a new image to the end of the library after prompting the user to enter name of the image-file holding the image and a title for the image.
- `remove`
  Removes one of the images currently in the library after prompting the user to enter the number of the image they'd like to remove. If the number the user enters doesn't correspond to an image currently in the library, an error message is printed out.

If the user ever enters an invalid command (i.e., something other than `quit`, ... , `remove`), an error message is printed out.

Save your completed program in the file `assign06.py`.

A Sample Solution to Run on Your Own Machine:
Download the files `solution06.pyc` and `runSolution06.py` to the same place (i.e., both to the same folder or both to the desktop). The file `solution06.pyc` contains a sample solution to this assignment. (The file isn't human-readable, so don't bother trying to open it with an editor, a word processor, Wing 101, etc.)

To run the sample solution (so that you can see how a complete solution should behave), use Wing 101 to open and execute the file `runSolution06.py`.

A Sample Solution to Run on the Olin 202 Machines:

Download the files `solution06Olin202.pyc` and `runSolution06Olin202.py` to the same place (i.e., both to the same folder or both to the desktop). The file `solution06Olin202.pyc` contains a sample solution to this assignment. (The file isn't human-readable, so don't bother trying to open it with an editor, a word processor, Wing 101, etc.)

To run the sample solution (so that you can see how a complete solution should behave), use Wing 101 to open and execute the file `runSolution06Olin202.py`.

Strategy:

Use the sorts of programming strategies you've seen in Chapters 1-10 of *M&R*, the lectures, and the labs. You'll need functions, loops, if-statements, images, strings, and lists.

Don't try to write the entire program at once: do it in stages and get one step more-or-less completely worked before you move on to the next step. Further, don't try to solve the program by writing just one big main program: break the program up into functions. I suggest you work on things in roughly the following manner and order:

(1) Complete the main program so that it reads in commands, decides what kind of command it has, and prints out an error message for invalid commands. For each of the commands (except for the `quit` command), have your main program simply call functions that take care of all the processing for the various commands. To start with, these functions won't do anything at all.
(2) Test your main program as it now stands to see if it's performing the basics of command input and command processing correctly.
(3) Complete the function for processing the `list` command.
(4) Complete the function for processing `add` command. This function will have to read the necessary command parameters (i.e., the image-file name and the title) from the user and then deal with them.
(5) Complete the function for processing the `showall` command.
(6) Test your program as it now stands to see if it's handling sequences of `list`, `add`, and `showall` commands correctly.
(7) Complete the function for processing the `show` command. This function will have to read the necessary command parameters from the user and then deal with them.
(8) Complete the function for processing the `remove` command. This function will have to read the necessary command parameters from the user and then deal with them.

The preceptors and I are happy to give you help. When/if you come to us for help, please bring with you whatever work you've done (pictures, notes, current code listings, files, your laptop, and whatever). The earlier you start on the assignment, the earlier you'll discover where your issues might be, and the earlier you'll be able to seek appropriate help.

Make your code not just correct, but also beautiful and comprehensible to other people. Be sure to supply a comment containing your name (at the very top of your program) along with other comments dispersed throughout the program as you see fit.

NOTE: Programs that clearly indicate no serious effort at breaking the program into functions

will be rejected out-of-hand and will receive a score of zero.

NOTE: Programs that clearly indicate no serious effort at producing an adequate amount of meaningful comments will be rejected out-of-hand and will receive a score of zero.

Be sure your code is neatly formatted and uses well-chosen variable names (some rather poor choices for variable names clearly include – but aren't limited to – cryptic abbreviations of words, random words that have little to do with the program, and content-free words like "variable1", "temp", "b", "h", "fn", "pic", "img", "win", etc.). If you borrow code from an example presented in lecture/lab or in the relevant chapters of *M&R*, be sure to acknowledge the source of the code (in a comment). (Please recall that searching the web looking for solutions is never acceptable for this course.) Your score will depend on the style, form, and correctness of your program.

What to Hand in:

- Your file `assign06.py`, submitted using the `Assignment 6` item on the `Assignments` page of the CS150 Katie course
- A readable printed listing of your file `assign06.py`, a listing that avoids awkward line-wrapping and so on – If necessary, consider printing from a word processor so that you can choose smaller font sizes, two-up printing, and so on. (Windows users: Listings printed using the Notepad application aren't acceptable.)