COM S 1270 - 'The Ultimate TODO List!'

Please see the assignment information on Canvas for this assignment's initial due date and final deadline.

Assignment Objective

For this assignment, you will be building "The Ultimate TODO List." This is essentially a task progress tracking application that will help you to be more organized in your studies and work.

The purpose of this assignment is to allow you to become more familiar with the use of 'lists' and 'dictionaries' in the context of lists being the values contained by a dictionary. This 'dictionary of lists' will contain the data for the application. You will be given a pre-coded 'start' file, called `ultimateTODO.py`, to which you will have to add core functionality. This assignment will also require substantial use of logical step-by-step thinking.

Instructions

You should study the file(s) provided, and notice the various 'TODO' comments contained therein. These comments indicate places in the file that you should modify the code to be your own. These 'TODO' comments are the places in the script where code/ student work will be evaluated for the final grade on the assignment. **DO NOT PLACE CODE OUTSIDE OF THESE TODO STATMENT AREAS.**

NOTE: IF YOU DO NOT USE THE PROVIDED 'START FILE,' YOU WILL BE GIVEN A ZERO (0) ON THE ASSIGNMENT. DO NOT ATTEMPT TO PROGRAM YOUR OWN VERSION OF THE APPLICATION!

Your work should be your own, completely original effort.

Other Requirements:

Additionally, you do *not* need to compensate for the crashes that occur when the user enters a value of the incorrect type. (E.g., The program asks for an integer, but the user enters a letter instead.)

It is required for you to write your name, the date you started working on your script, and a short explanation of what your code does at the top of your file. For example:

```
# Matthew Holman             11-25-2024
# Assignment #5
#
# This is a helpful application for tracking the completion of tasks.
```

Do *not* attempt to 'double dip' by placing your 'header' printout information at the top of your file - the header and the initial comment block are different things.

Your work should be your own, completely original effort. Meaning - you should not just copy my code explanation above, but rather you should try to come up with one of your own.

The actual Python script itself can be programmed in any way (excluding cheating ala ChatGPT) so long as the output resembles that below. This includes the use of the initial 'header' when starting the program (i.e. the program title, who it is by, the student's class/ section, etc.). Please note that the 'header' will be strictly enforced with this assignment. Be sure to include the title, your name, and your class/ section.

Example Output

```
The Ultimate TODO List!

By: Matthew Holman
[COM S 127]

-----------------------------------------------------------------
MAIN MENU: [n]ew list, [l]oad list, or [q]uit?: n


backlog: []
```

```
todo: []
in_progress: []
in_review: []
done: []

---------------------------------------------------------------------
APPLICATION MENU: [a]dd to backlog, [m]ove item, [d]elete item, [s]ave list, or [q]uit to main menu?: a

Enter An Item: laundry


backlog: ['laundry']
todo: []
in_progress: []
in_review: []
done: []

---------------------------------------------------------------------
APPLICATION MENU: [a]dd to backlog, [m]ove item, [d]elete item, [s]ave list, or [q]uit to main menu?: a

Enter An Item: dishes


backlog: ['laundry', 'dishes']
todo: []
in_progress: []
in_review: []
done: []

---------------------------------------------------------------------
APPLICATION MENU: [a]dd to backlog, [m]ove item, [d]elete item, [s]ave list, or [q]uit to main menu?: a

Enter An Item: walk cat


backlog: ['laundry', 'dishes', 'walk cat']
todo: []
in_progress: []
in_review: []
done: []

---------------------------------------------------------------------
APPLICATION MENU: [a]dd to backlog, [m]ove item, [d]elete item, [s]ave list, or [q]uit to main menu?: a

Enter An Item: wash car


backlog: ['laundry', 'dishes', 'walk cat', 'wash car']
todo: []
in_progress: []
in_review: []
done: []

---------------------------------------------------------------------
APPLICATION MENU: [a]dd to backlog, [m]ove item, [d]elete item, [s]ave list, or [q]uit to main menu?: m

Enter An Item To Move: dishes
Enter The List To Move dishes To: todo

backlog: ['laundry', 'walk cat', 'wash car']
todo: ['dishes']
in_progress: []
in_review: []
done: []

---------------------------------------------------------------------
APPLICATION MENU: [a]dd to backlog, [m]ove item, [d]elete item, [s]ave list, or [q]uit to main menu?: d

Enter An Item To Delete: wash car

backlog: ['laundry', 'walk cat']
todo: ['dishes']
in_progress: []
in_review: []
```

```
done: []

---------------------------------------------------------------
APPLICATION MENU: [a]dd to backlog, [m]ove item, [d]elete item, [s]ave list, or [q]uit to main menu?: s

Enter List Name (Exclude .lst Extension): asdf
Saving List...


backlog: ['laundry', 'walk cat']
todo: ['dishes']
in_progress: []
in_review: []
done: []

---------------------------------------------------------------
APPLICATION MENU: [a]dd to backlog, [m]ove item, [d]elete item, [s]ave list, or [q]uit to main menu?: q

Returning to MAIN MENU...

---------------------------------------------------------------
MAIN MENU: [n]ew list, [l]oad list, or [q]uit?: l

Enter List Name (Exclude .lst Extension): asdf

backlog: ['laundry', 'walk cat']
todo: ['dishes']
in_progress: []
in_review: []
done: []

---------------------------------------------------------------
APPLICATION MENU: [a]dd to backlog, [m]ove item, [d]elete item, [s]ave list, or [q]uit to main menu?: q

Returning to MAIN MENU...

---------------------------------------------------------------
MAIN MENU: [n]ew list, [l]oad list, or [q]uit?: q

Goodbye!
```

Files Provided

`ultimateTODO.py`

Special Notes

NOTE: This assignment will *slightly less* difficult than the previous assignments. However:

- Completing this assignment may require you to start your work 'before the last minute.' Please plan accordingly.
- Your script **CANNOT** crash under any circumstances except for those circumstances noted below. Any portions of your code where the script crashes will receive a zero (0) for that aspect of the assignment.
    - Understand, when the word 'crash' is used in the instruction above, what this means is 'crashing under expected use cases given the level of knowledge attained in the class.'
        - You do not currently have the ability to enforce user input types (although you will later in the semester). As such, if you ask for an integer as input (e.g., 1), and the user provides a letter (e.g., 'a'), and it crashes, then this is *not* a problem *yet*.

NOTE: You are turning in your CODE - NOT the OUTPUT of your code.

- Your code will be run by the TAs, and the output of those runs, along with the code itself, is what will be evaluated.

NOTE: Assignments turned in in any other format other the specified file types will not be accepted.

- Screenshots of code **will not** be accepted.
- `.sln` files are **not** code files - they contain **no** Python code and **will not** be accepted.
- `.zip`, `.rar`, `.tar.gz`, and other compressed files **will not** be accepted unless otherwise specified.
- If your submission is not in a `.py` file, when so specified, or if the submission is not accompanied by **all** the files needed to

run the submission, the submission will not be graded.
    - **THIS WILL LEAD TO YOU RECEIVING A ZERO (0) ON THE ASSIGNMENT.**
    - **You will *NOT* be allowed to re-submit your work after the final deadline in this case.**

NOTE: You have the ability (and responsibility) to 'double check' that your work/ submission is correct when you turn it in.

- If you accidentally turn in the wrong submission, you will **not** be allowed to resubmit it after the final deadline.
- Please note, you can submit your work as many times as you like *before* the final deadline.