

COM S 127 - Assignment #5 Grading Rubric

This assignment was assigned on 10-24-2022

This assignment is due by 11:59 p.m. Friday, November the 4th (11-4-2022). It will be considered 'late' if turned in after this time. However, there will be a two (2) day 'grace period' extended until 11:59 p.m. Sunday, November the 6th (11-6-2022). If the assignment is turned in after 11:59 p.m. Friday, November the 4th, but before 11:59 p.m. Sunday, November the 6th, it will suffer at 10% 'late penalty' to the grading. The assignment will not be accepted after 11:59 p.m. Sunday, November the 6th.

Assignment Objective

The purpose of this assignment is to allow students to become more familiar with the use of 'lists' and 'dictionaries.' This assignment will require the use of a 'dictionary of lists' which will contain the data for the application. The student will be given a pre-coded 'start' file, to which they will have to add core functionality. This assignment will also require substantial use of logical step-by-step thinking.

Instructions

Students should study the file provided, and notice the various 'TODO' comments. These indicate tasks in the file that students should complete by typing in their own original code. These 'TODO' comments indicate items in the script which will be evaluated for the final grade on the assignment. The file the student submits **must** be named **ultimateTODO.py**.

The various 'TODO' comments can all be found inside the **ultimateTODO.py** start file.

Files Provided

Students will have access to the following 'start file,' which will detail all of the TODO tasks: `ultimateTODO.py`

Example Output

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!

Special Notes

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

- Completing this assignment may require the student to start their work 'before the last minute.' Please plan accordingly.
- The student's script **CANNOT** crash under any circumstances. Any portions of the student's code where the script crashes will receive a zero (0) for that aspect of the game.
 - For example, if a student's script crashes during the 'addItem' section, the student will not earn the point for implementing that particular feature.

NOTE: Assignments turned in in any other format other than a .py file 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.
- If a student's submission is not in a .py file, it will not be graded.
 - **THIS WILL LEAD TO THE STUDENT RECEIVING A ZERO (0) ON THE ASSIGNMENT.**
 - **Students will *NOT* be allowed to re-submit their work in this case.**

NOTE: Submitting the assignment on Canvas may result in either an error message or a 'spinning blue circle' on the submission page. This is the normal/ expected behavior.

- So long as the top of the page reads 'Submitted' in green text, there should not be a problem.
- The student can email a Graduate TA/ UGTA to confirm the status of their submission.
- Students may encounter difficulties in submitting their assignment due to issues with the Canvas platform itself. However, this will **not** be an excuse to turn in work past the deadline.
 - **THIS MEANS THAT STUDENTS SHOULD NOT WAIT UNTIL THE 'LAST MINUTE' TO ATTEMPT TO SUBMIT THEIR ASSIGNMENT TO CANVAS. THUS, STUDENTS WILL NEED TO 'PLAN THEIR LIVES' IN REGARD TO THEIR TIMELY SUBMISSION.**
 - **STUDENTS SHOULD ATTEMPT TO SUBMIT THEIR WORK *BEFORE* THE FINAL DEADLINE.**

Grading Items

- **(Due Date AND File Name)** Was the assignment turned in by the due date, **AND** is the file named **ultimateTODO.py**? _____ / 1
- **(Name In Script AND Delete First TODO AND Name In Output)** Has the student typed in their name/ date/ assignment number at the top of the **ultimateTODO.py** file, **AND** did the student delete the first 'TODO' comment in **ultimateTODO.py** such that the first thing in the file is their name/ date, **AND** Has the student added their name/ section number to the initial script output?: _____ / 1
- **(choice == "a" Section Tasks)** Has the student completed the choice == "a" section tasks?: _____ / 1
- **(choice == "m" Section Tasks)** Has the student completed the choice == "m" section tasks?: _____ / 1
- **(choice == "d" Section Tasks)** Has the student completed the choice == "d" section tasks?: _____ / 1
- **(checkItem Section Tasks)** Has the student completed the checkItem section tasks?: _____ / 1
- **(addItem Section Tasks)** Has the student completed the addItem section tasks?: _____ / 1
- **(deleteItem Section Tasks)** Has the student completed the deleteItem section tasks?: _____ / 1
- **(moveItem Section Tasks)** Has the student completed the moveItem section tasks?: _____ / 1
- **(printTODOList Section Tasks)** Has the student completed the printTODOList section tasks?: _____ / 1

TOTAL _____ / 10