**Theresa Ward**

November 24, 2020

IT FDN 110 A

Assignment 06

# https://github.com/ScrappySnacks/IntroToProg-Python-Mod06

# **“To Do” List Script: Functions and Classes**

## Introduction

This assignment continues the modification and improvement of the “to do” list script. Given starter code, we are asked to create more functions to organize the code, which loads data from a \*.txt file into a Python list of dictionary objects. The user is presented a menu of options which is very similar to past assignments. This document will describe the steps taken to create the program and ensure it is working as expected. As with past assignments, PyCharm is used here for script development.

## Script Development

For this assignment, we are to structure the program in PyCharm using the starter code provided, which is organized by separation of concerns. We are to define several functions; three in the Processor class and two in the IO class. In addition, there is code needed in the *while* loop to execute the functions defined in the Processor and IO classes. For this assignment, I have converted function names to snake case per convention. The functions completed for this assignment are described below. Please note: For a reason I haven’t yet determined, the use of the relative file location for “ToDoList.txt” was not working when executing via the command window. The script will only consistently work in PyCharm and command window, if I spell out the entire file location.

### Processor class:

1. AddDataToList(): This function captures a row of data in a dictionary object from user input of task and priority. It then appends this row to existing rows in a list object, lstTable.
2. RemoveDataFromList(): This function removes a task provided by the user. I declare a variable, *i*, which will be used as a flag to alert the user if the task is not found. If the task is contained in the list, the script iterates this variable to equal one. If the task is not found, the variable remains equal to zero and displays the message, “Task is not listed. Try again.” The deletion of the task is accomplished through the remove() method.
3. WriteDataToFile(): This function has an argument, strFileName, that is declared at the very beginning of the python script and points to “ToDoList.txt”. After opening strFileName, this function writes tasks and priorities, contained in lstTable, to file; adding a new line after each entry.

### IO class:

1. InputNewTaskAndPriority(): This function captures user input of task and priority and converts the input to lower case and string data type.
2. InputTaskToRemove(): This function captures user input of task and converts to lower case and string data type.

### Main body:

In the main body of the script, a menu of options is presented within a *while* loop that is set to Boolean True. Two functions print the current contents of the list object, lstTable, and print the menu options. This is followed by requesting user input, which is assigned to strChoice. An *if* statement is used to run the script based on the value of strChoice. Strip() was also added to the remaining *elif* statements.

* If the user enters “1,” the script prompts the user to add a new task and priority per the function InputNewTaskAndPriority(), which unpacks user input and assigns to “task” and “priority.” These are passed to AddDataToList(). This is followed by the InputPressToContinue() function. An example of PyCharm output when “1” is chosen is shown in Figure 1.

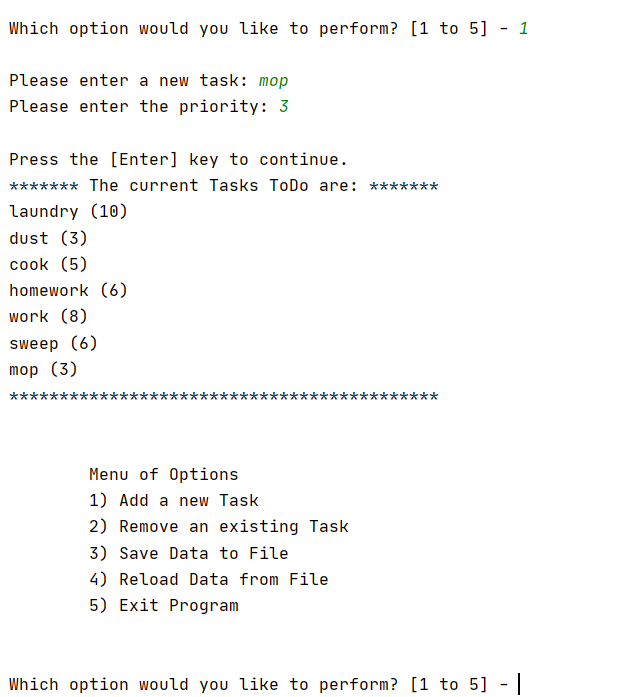


Figure 1. Output in PyCharm: Menu option 1

* If the user enters “2,” the script prompts the user to enter a task for deletion per the function InputTaskToRemove(), which assigns the value to “task.” This is passed to RemoveDataFromList(). This is also followed by the InputPressToContinue() function. An example of PyCharm output when “2” is chosen is shown in Figure 2.

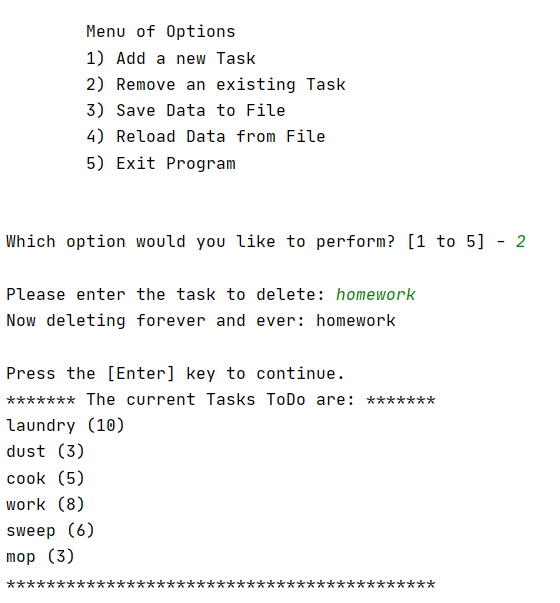


Figure 2. Output in PyCharm: Menu option 2

* If the user enters “3,” the script prompts the user for a “y” or “n” response per the function InputYesNoChoice(). If “y” is chosen, the function WriteDataToFile() is called, followed by InputPressToContinue(). Else, the user is told the save is cancelled. An example of PyCharm output when “3” is chosen is shown in Figure 3. Text file contents are also shown.

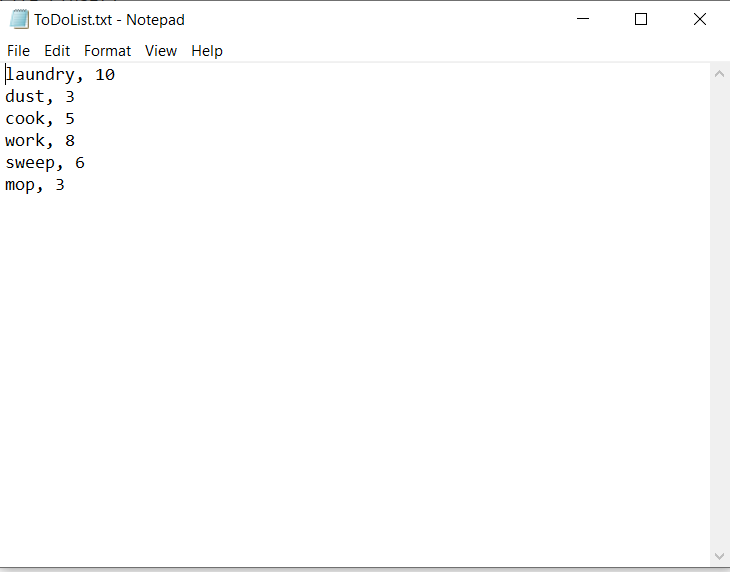
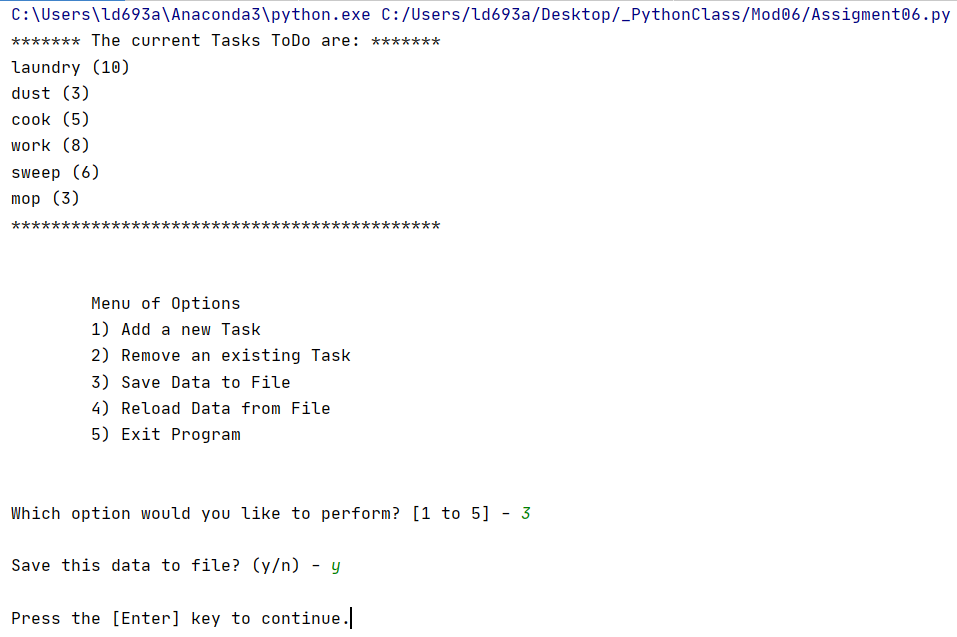


Figure 3. Output in PyCharm when “3” selected

* If the user enters “4,” the script prints a warning and then prompts the user for a “y” or “n” response per the function InputYesNoChoice(). If “y” is chosen, the function ReadDataFromFile() is called, followed by InputPressToContinue(). Else, the user is told the file reload is cancelled. An example of PyCharm output when “4” is chosen is shown in Figure 4.

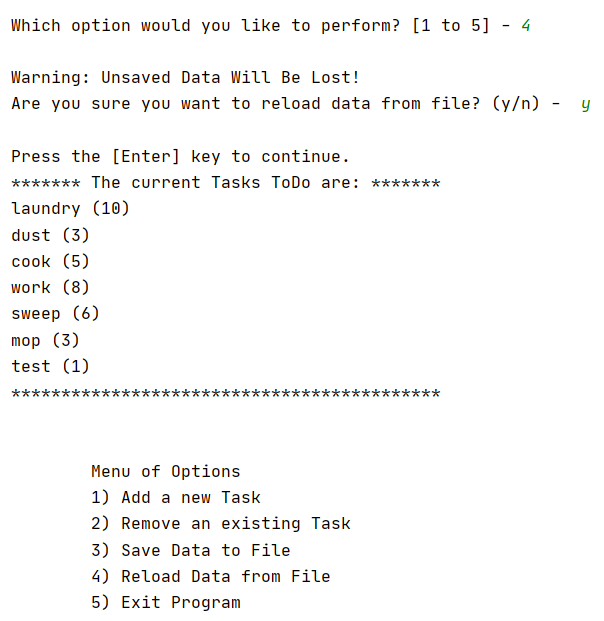


Figure 4. Output in PyCharm when “4” selected. The text file was modified ahead of time to show that it works.

* Lastly if the user enters “5,” the program ends.

It was extremely helpful to test portions of the script outside the main Python script. This allowed for much easier troubleshooting. However some challenging aspects of this assignment included keeping track of portions of the code / interactions between separation of concerns and remembering to call the class along with the function.

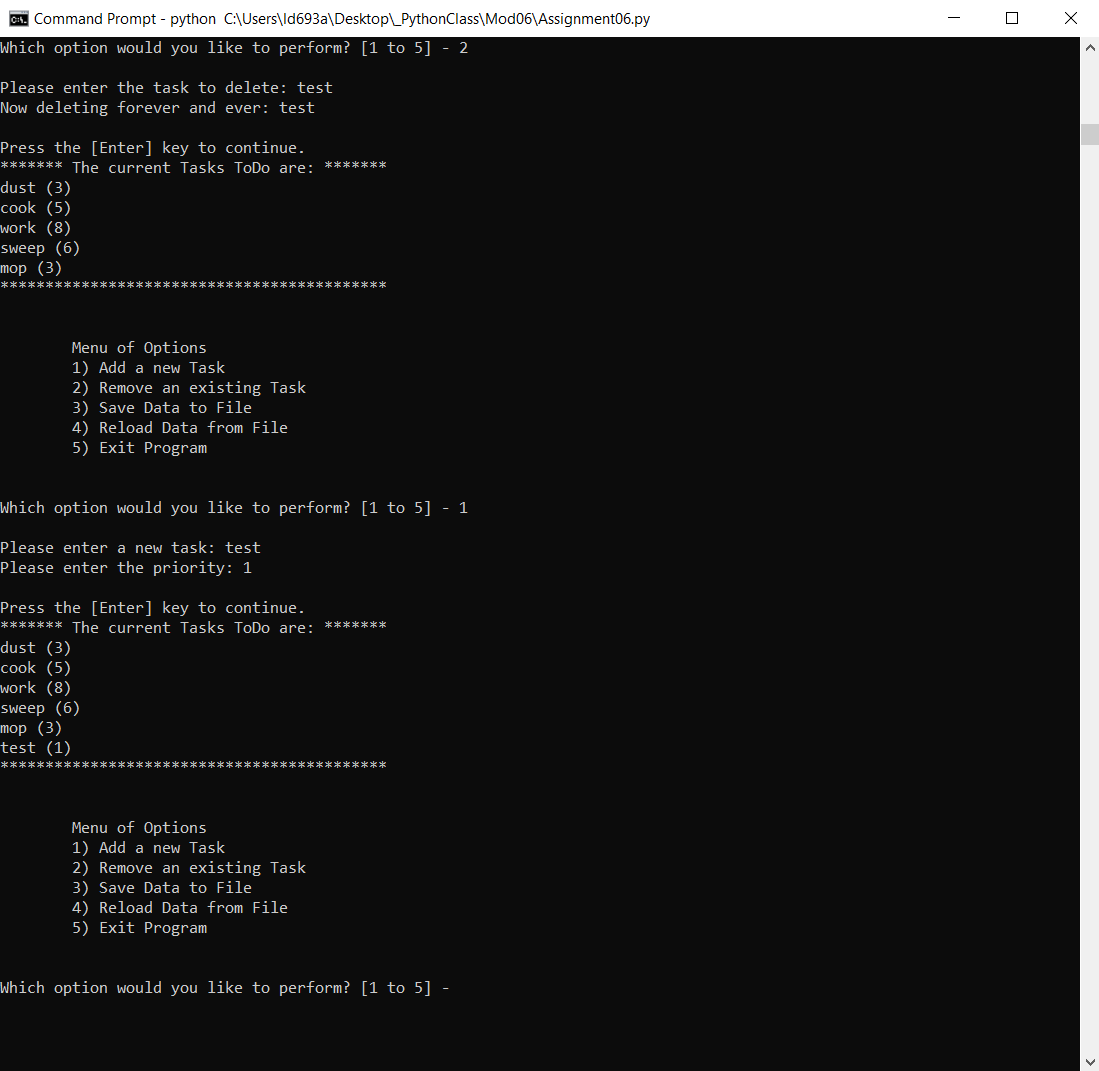


Figure 5. Result of running the Python script via the command window

## Summary

In summary, I was successful in completing a script that meets the requirements of Module 06. The code can add a new task, remove an existing task, save data to file, and reload data from file. Testing portions of the code outside the main script was a very helpful approach in completing development of the assignment. I also need to investigate issues that I am having with accessing “ToDoList.txt“ from the relative file location when executing via the command window.