Name: Stephanie Tarczynski

Date: 11/17/2020

Course: IT FDN 110 A

Assignment: Module 05, Assignment 05

How to Store, Manipulate, and Output Data in a Dictionary

# Understanding the Task

Before creating the script, I made sure I understood the task at hand. Assignment 05 requires the program to collect and manipulate user input data to create a To-Do List. The program gives the user the option to display data, add data, delete data, save data, and exit the program. The data is saved to a text file. A starter python file is provided. The assignment requires this code to be run in PyCharm and the OS console window.

# Creating the Script

For this program, I created the code in PyCharm using the Assignment05 starter file provided. I edited the header change log to capture my changes to the code. I used the pseudo-code to guide me through creating the code for the various tasks the program is supposed to perform. The overall objective of this assignment is to create a “To-Do” list in a separate file from Python. The user is given a “Menu of Options” while running this program:

1) Show current data

2) Add a new item.

3) Remove an existing item.

4) Save Data to File

5) Exit Program

The steps to creating the script for these options is explained in the following steps. The “Menu of Options” is displayed within a while loop and is displayed after each choice the user makes. The while loop continues to display these options until the chooses user exits the program.

## Initial Processing

At the start of the program I created a script to load any current data from a file called “ToDoList.txt”. I imported the rows into a list format. I then used the indexes within the list to add each element into dictionary row. The first key is the “Task” and the second key is the “Priority”. The dictionary row is appended to a table. I used the variables declared in the starter code for this portion and throughout the program script. See Figure 1 below for this first step.

*Figure 1. Assignment 05 initial processing step*

*# -- Processing -- #  
# Step 1 - When the program starts, load the any data you have  
# in a text file called ToDoList.txt into a python list of dictionaries rows (like Lab 5-2)*dataFile = open(objFile, **"r"**)  
for row in dataFile:  
 strData = row.split(**":"**)  
 dicRow = {**"Task"**: strData[0].strip(), **"Priority"**: strData[1].strip()} *#unpacking the data to form dictionary rows* lstTable.append(dicRow) *#appending dictionary rows to a table*dataFile.close()

## Option 1: Displaying Current Data

The first option is to display the current data. The data is displayed by looping through the dictionary rows in the list table. The “Task” and “Priority” keys are displayed for each dictionary row. The script for Option 1 is shown in Figure 2.

*Figure 2. Option 1 Script*

if (strChoice.strip() == **'1'**):  
 print(**"Task"** + **" | "** + **"Priority"**) *#header* for dicRow in lstTable:  
 print(dicRow[**"Task"**] + **": "** + dicRow[**"Priority"**]) *#printing the values of the dictionary row keys* continue

## Option 2: Adding New Data

New data is added to the list table by collecting new user inputs for the task and priority. The values are assigned to a dictionary row task and key. The new dictionary row is appended to the list table. The script for Option 2 is shown in Figure 3.

*Figure 3. Option 2 Script*

elif (strChoice.strip() == **'2'**):  
 strTask = input(**"Enter a task: "**)  
 strPriority = input(**"Enter the priority: "**)  
 dicRow = {**"Task"**: strTask, **"Priority"**: strPriority} *#adding new user input values to the dicRow keys* lstTable.append(dicRow)  
 continue

## Option 3: Removing Data

If the user selects option 3, they will be prompted to enter a task to be removed. This input is collected and assigned the strRemove variable. The program loops through each dictionary row in the list table and checks if strRemove matches the value in any of the “Task” keys in each of the dictionary rows. If there is a match the row is deleted and a message to the user is printed that indicates the task was deleted. If the task is not found, the program will print a message that the task is not in the list table and no removals will take place. The script for Option 3 is shown in Figure 4.

*Figure 4. Option 3 Script*

elif (strChoice.strip() == **'3'**):  
 strRemove = input(**"Enter the task you wish to remove: "**).lower().strip()  
 i=0  
 for dicRow in lstTable:  
 if strRemove in dicRow[**"Task"**]:  
 print(**"Task deleted."**)  
 lstTable.remove(dicRow)  
 i+=1  
 elif i == 0:  
 print(**"Task not found."**)  
 continue

## Option 4: Saving Data

The list table data can be saved to the “ToDoList.txt” file by selecting option at any point in the program run. Each of the dictionary rows is written to the file. The task and priority are separated with a semicolon. The user is informed that the data is saved. The script for Option 4 is shown in Figure 5.

*Figure 5. Option 4 Script*

elif (strChoice.strip() == **'4'**):  
 dataFile = open(objFile, **"w"**)  
 for dicRow in lstTable:  
 dataFile.write(dicRow[**"Task"**] + **":"** + dicRow[**"Priority"**] + **"**\n**"**)  
 dataFile.close()  
 print (**"Data saved."**)  
 continue

## Option 5: Exiting the Program

Finally the user can exit the program by choosing Option 5. A final message is printed and the while loop is broken. This option does not write any new data or changes to the “ToDoList.txt” file. Option 4 must be selected to do this. The script for Option 5 is shown in Figure 6.

*Figure 6. Option 5 Script*

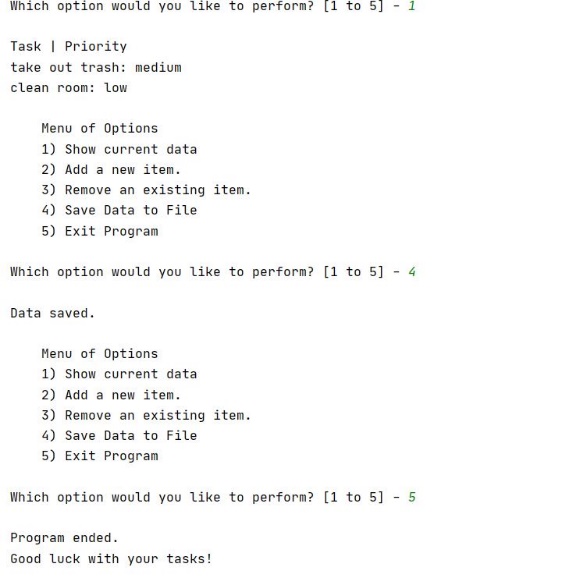
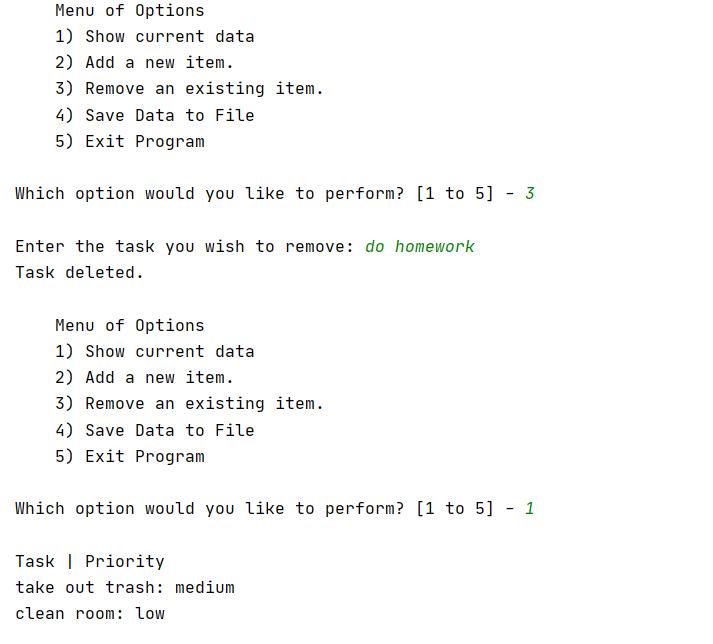
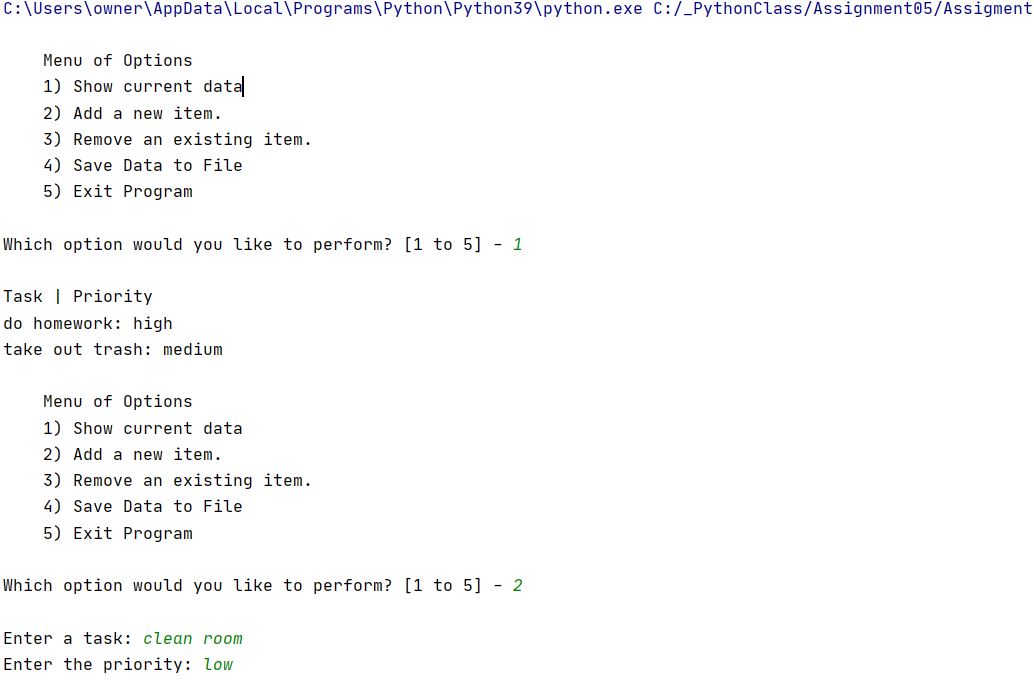
elif (strChoice.strip() == **'5'**):  
 print (**"Program ended."**)  
 print(**"Good luck with your tasks!"**)  
 break *# and Exit the program*

Please reference the Assignment05.py file to see the script in its entirety.

# Running the Code

To run the code, I first tested the code in PyCharm. I ran the program, followed the prompts and verified that my inputs were written to the HomeInventory.txt file. The program worked in PyCharm as shown in Figure 7 (images are labeled 1-3 to show the order of steps taken) and wrote the inputs to the ToDoList.txt file as shown in Figure 8.

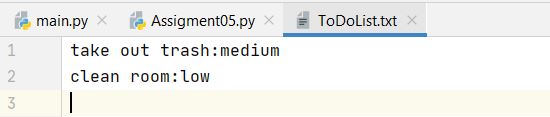
.



1

2

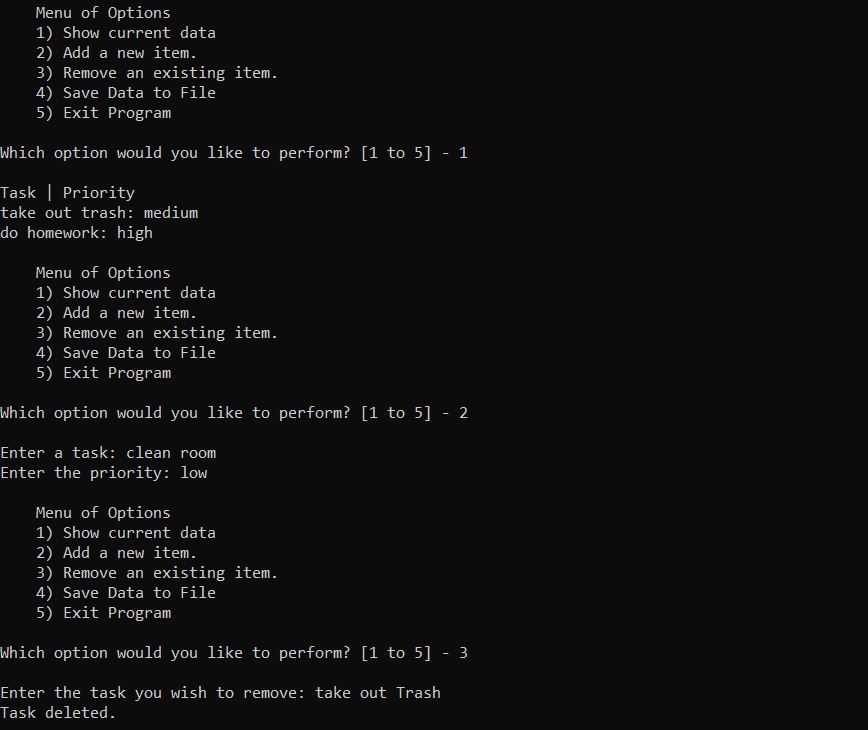
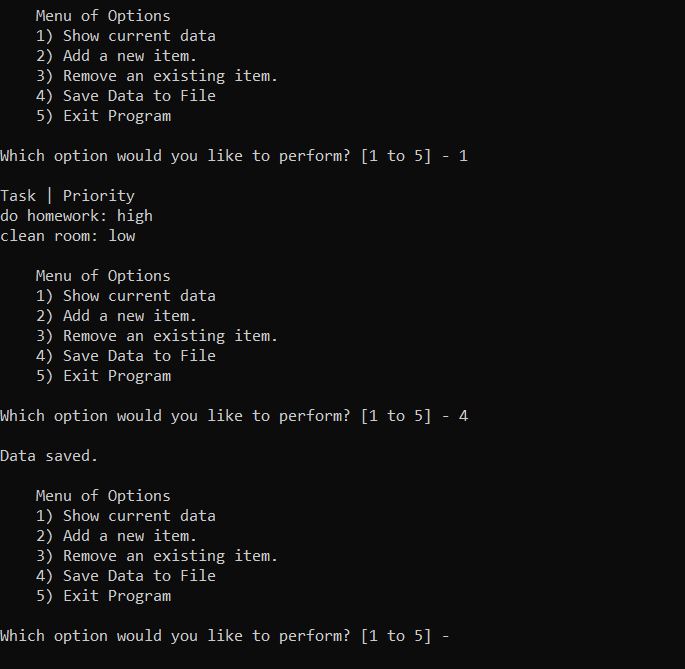
3



*Figure 8. ToDoList.txt output from running Assignment05.Py in PyCharm*

*Figure 7. Assignment 05 running PyCharm*

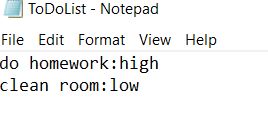
After testing the code in PyCharm, I then ran the same program in the command console. The program can be seen running in Figure 9 (images are labeled 1-3 to show the order of steps taken) and the output can be seen in Figure 10. Since I was running the program in write mode, the new objects and values I entered replaced the ones that I previously entered in PyCharm.



*Figure 9. Assignment 05 running in the OS console window*

2

1



*Figure 10. ToDoList.txt output from running Assignment05.Py in OS console window*

# Summary

To complete this assignment, I used the starter file for Assignment 05 provided and edited the script in PyCharm. I followed the prompts laid out in the pseudo-code and comments to accomplish the assignment tasks. A while loop was used to give the user the options to display, add data to, remove data from, and save data to a file until the user exits the program. The user input data for tasks and their priorities was stored and manipulated in a table of dictionary rows. The program outputs the values of the dictionary rows to a text file. This assignment demonstrated how to use Python to gather, store, and manipulate data using dictionaries and tables; using inputs from a text file and outputting the result to the text file.