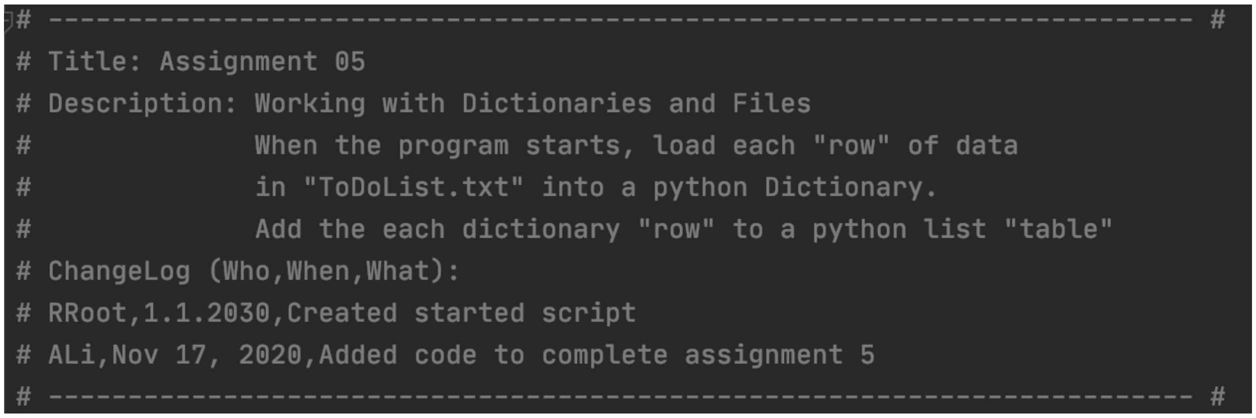
Andrew Li
Nov. 17, 2020
Foundations of Programming: Python
Assignment05
https://github.com/idrew4u/IntroToProg-Python

To Do List Python Script

Introduction
This round, we were tasked to do something a little different. We were asked to complete a started python script. The originator already outlined how they wanted the script to perform using pseudocode and comments. This gave the next programmer an idea on how to create the rest of the script to accomplish the task of gathering data from the user and saving the information into a text file. The objective of the program was to load up a previous task list (if available) and add additional items to the list, data including the task and the priority of the task. Afterwards, we want to save the updated information in the text file.

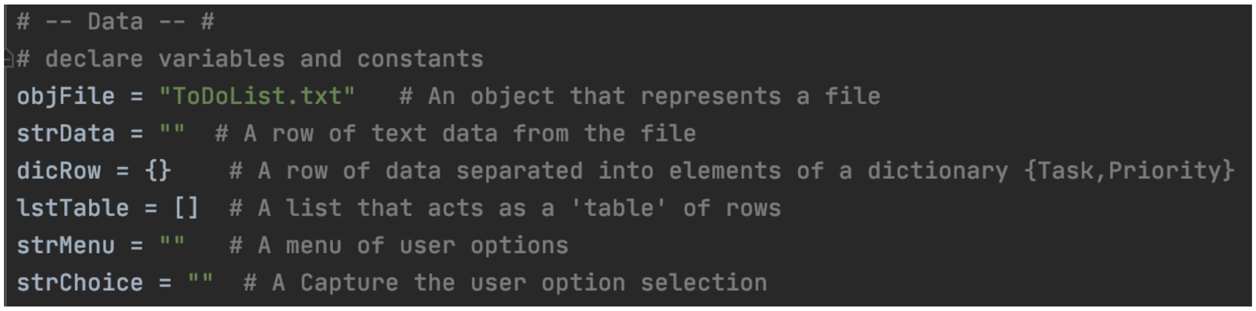
Creating the Script
Script Header
The first thing we should do is write a header for the script. The header is the first thing you see and gives us a basic idea what the script is trying to accomplish. It asks for the title of the program, a description of what it does, and logs changes to the script (Figure 1). The logs should include who made the change, the date it was changed, and what was changed in the script. In this situation, the originator already started the script header, we just had to update the information to allow other viewers we made updates and changes to the script.



```
# ---------------------------------------------------------------------- #
# Title: Assignment 05
# Description: Working with Dictionaries and Files
#              When the program starts, load each "row" of data
#              in "ToDoList.txt" into a python Dictionary.
#              Add the each dictionary "row" to a python list "table"
# ChangeLog (Who,When,What):
# RRoot,1.1.2030,Created started script
# ALi,Nov 17, 2020,Added code to complete assignment 5
# ---------------------------------------------------------------------- #
```

Figure 1: The header includes the title, description, and log changes.

Data
The next portion of the code outlines the data we will be using and storing (Figure 2). As the pseudocode states, it declares the variable and constants we plan on using within the script. It then goes a little more into detail what those variables and constants would be used for. The originator had already had them laid out, we just have to incorporate them to accomplish our end goal.

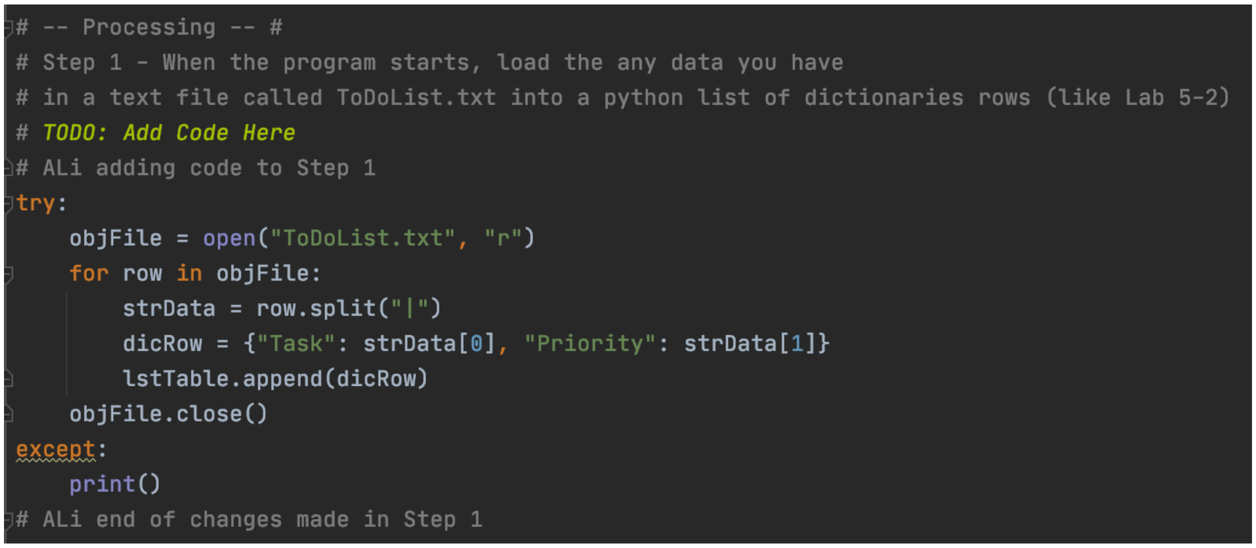


```
# -- Data -- #
# declare variables and constants
objFile = "ToDoList.txt"   # An object that represents a file
strData = ""  # A row of text data from the file
dicRow = {}    # A row of data separated into elements of a dictionary {Task,Priority}
lstTable = []  # A list that acts as a 'table' of rows
strMenu = ""   # A menu of user options
strChoice = ""  # A Capture the user option selection
```

Figure 2: Listing the data that will be references within the script.

Loading List Data from a File

The originator had already outlined the next few steps for us. Step 1 wanted us to load the data from the "ToDoList.txt" into a python list of dictionary rows when the program starts. In the following code (Figure 3), we are able to open the "ToDoList.txt" file, and load them back into the memory.. We used the variables and constants from the previous section and defined them to process the data.

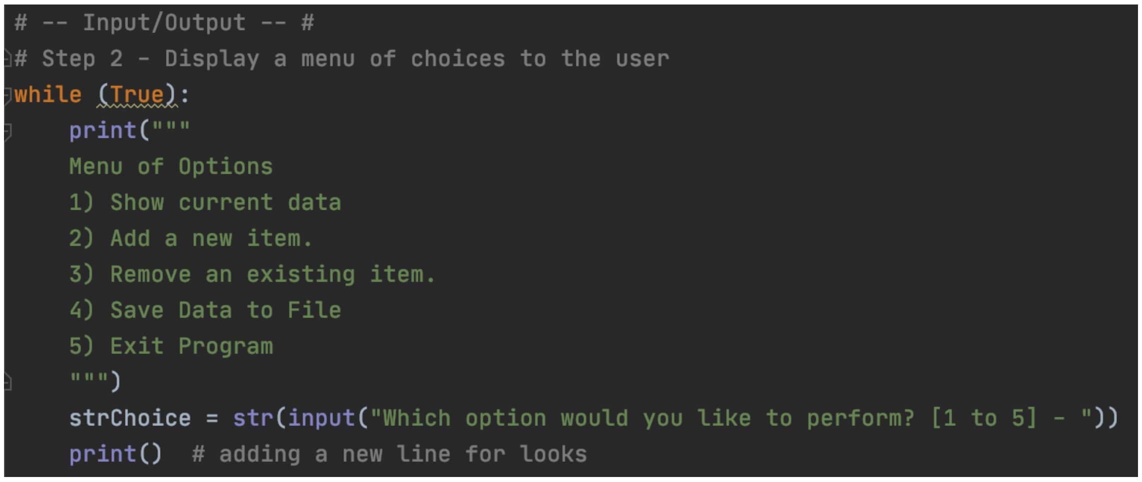


```
# -- Processing -- #
# Step 1 - When the program starts, load the any data you have
# in a text file called ToDoList.txt into a python list of dictionaries rows (like Lab 5-2)
# TODO: Add Code Here
# ALi adding code to Step 1
try:
    objFile = open("ToDoList.txt", "r")
    for row in objFile:
        strData = row.split("|")
        dicRow = {"Task": strData[0], "Priority": strData[1]}
        lstTable.append(dicRow)
    objFile.close()
except:
    print()
# ALi end of changes made in Step 1
```

Figure 3: Opening and loading the "ToDoList.txt" back into memory.

Menu

As outlined, Step 2 displays a menu of choices to the user. By using the while loop, the menu will repeat until the user is ready to exit the program (Choice 5). The menu prints out the 5 options for the user to choose from, then gathers an input in order to proceed with that option (Figure 4).

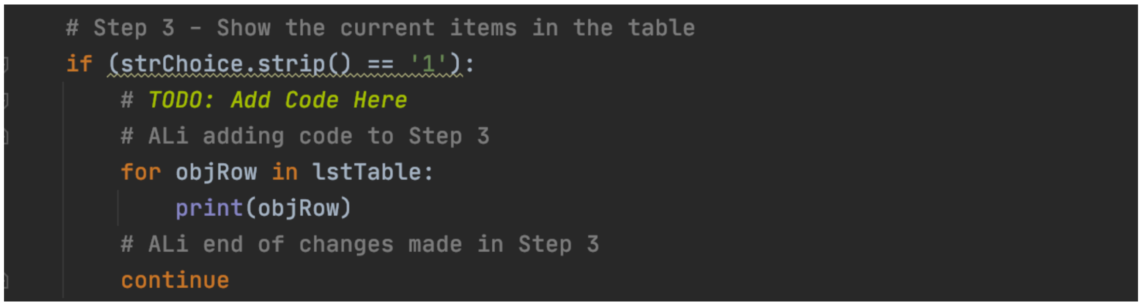


```
# -- Input/Output -- #
# Step 2 - Display a menu of choices to the user
while (True):
    print("""
    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program
    """)
    strChoice = str(input("Which option would you like to perform? [1 to 5] - "))
    print()  # adding a new line for looks
```

Figure 4: Displays menu of choices and asks for user input of choice

Show current items in table

If the user chooses choice 1, it bring you to step 3 in our code. By choosing option 1, it simply prints out the data already collected in available (Figure 5). At the end of this section, it then loops us back to the menu of choices.

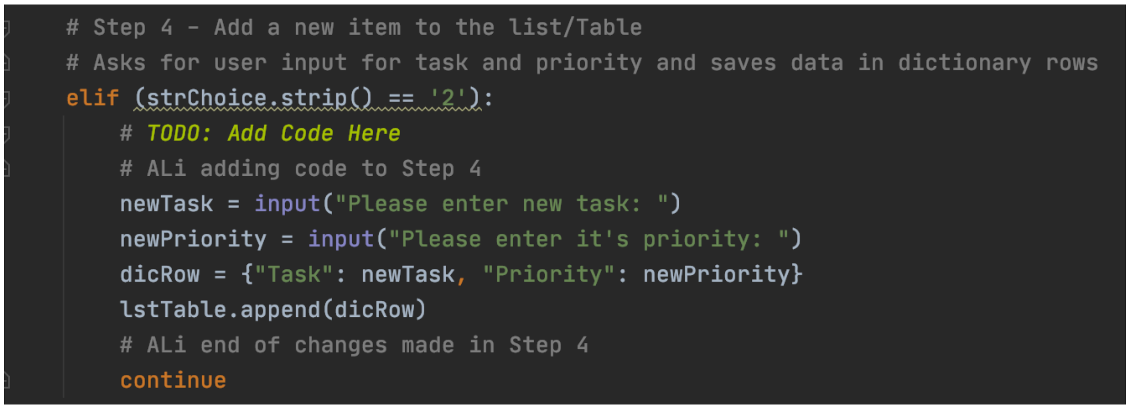


```
    # Step 3 - Show the current items in the table
    if (strChoice.strip() == '1'):
        # TODO: Add Code Here
        # ALi adding code to Step 3
        for objRow in lstTable:
            print(objRow)
        # ALi end of changes made in Step 3
        continue
```

Figure 5: Displays data in table back to user.

Adding New Item to List/Table

If the user does not choose option 1, the program then checks to see if they chose option 2. Option 2 (Step 4), allows the user to add a new item to the list/table. Once the user chooses option 2, it prompts the user to enter a new task, followed by the priority level. We then create a dictionary entry via the data collected from the user, and adds it to our table. When all the above is done, it then returns back to the menu.

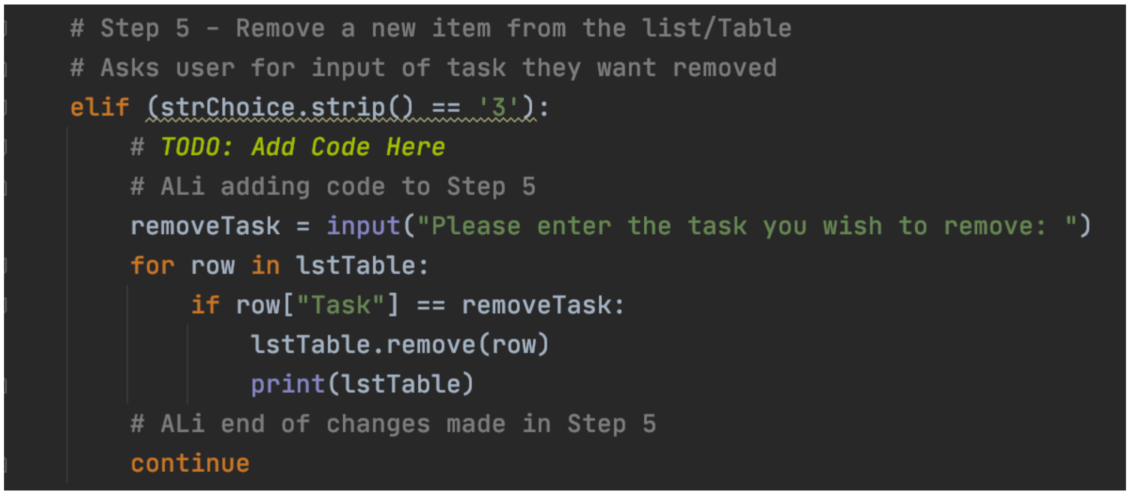


```python
    # Step 4 - Add a new item to the list/Table
    # Asks for user input for task and priority and saves data in dictionary rows
    elif (strChoice.strip() == '2'):
        # TODO: Add Code Here
        # ALi adding code to Step 4
        newTask = input("Please enter new task: ")
        newPriority = input("Please enter it's priority: ")
        dicRow = {"Task": newTask, "Priority": newPriority}
        lstTable.append(dicRow)
        # ALi end of changes made in Step 4
        continue
```

Figure 6: Asks user to input the new task and it’s priority.

Remove New Item from List/Table

Next is option 3 (our Step 5). Option 3 allows us to remove an item we had previously entered. If the user’s input that matches a task in our table, it will remove that task and priority from our set of data (Figure 7). I then wanted to display the table back to the user to show that the item has been removed.

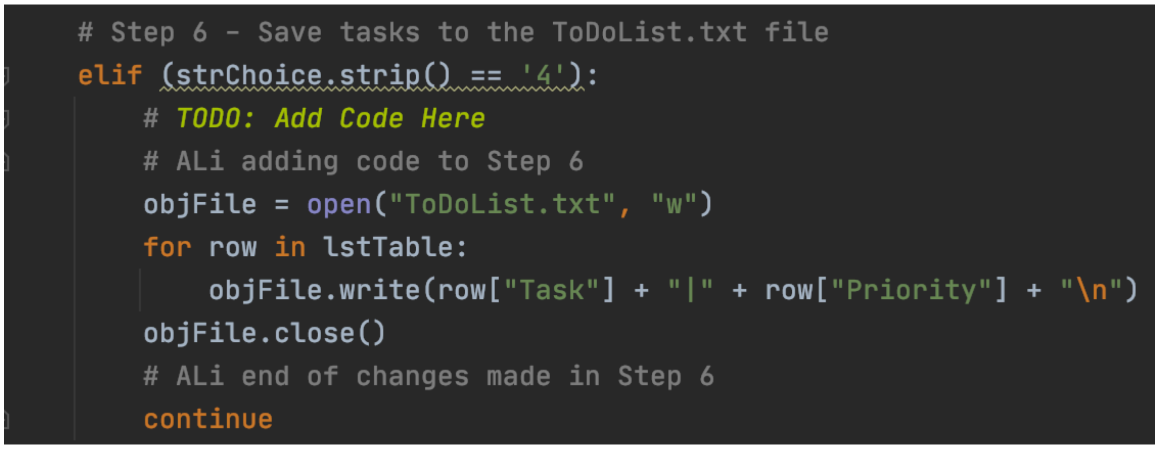


```python
    # Step 5 - Remove a new item from the list/Table
    # Asks user for input of task they want removed
    elif (strChoice.strip() == '3'):
        # TODO: Add Code Here
        # ALi adding code to Step 5
        removeTask = input("Please enter the task you wish to remove: ")
        for row in lstTable:
            if row["Task"] == removeTask:
                lstTable.remove(row)
                print(lstTable)
        # ALi end of changes made in Step 5
        continue
```

Figure 7: Removes the item from table, then prints the remaining items within the table.

Saving Tasks to File and Closing the Program

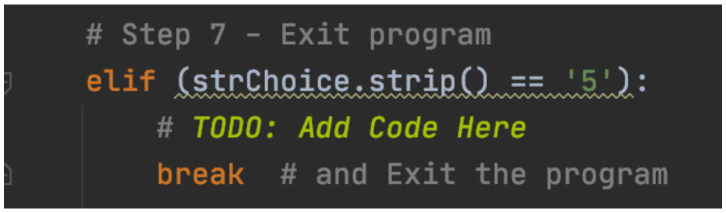
Lastly, we come to our last two choices given to the user: saving the file and exiting the program. If the user selects choice 4 (our Step 6), it opens up the “ToDoList.txt” and writes the newly added tasks to that text file (Figure 8).



```
# Step 6 - Save tasks to the ToDoList.txt file
elif (strChoice.strip() == '4'):
    # TODO: Add Code Here
    # ALi adding code to Step 6
    objFile = open("ToDoList.txt", "w")
    for row in lstTable:
        objFile.write(row["Task"] + "|" + row["Priority"] + "\n")
    objFile.close()
    # ALi end of changes made in Step 6
    continue
```

Figure 8: Saves the entries to "ToDoList.txt" file.

If the user decides to choose option 5 (Step 7) to exit the program, it will break the loop and end the program (Figure 9).

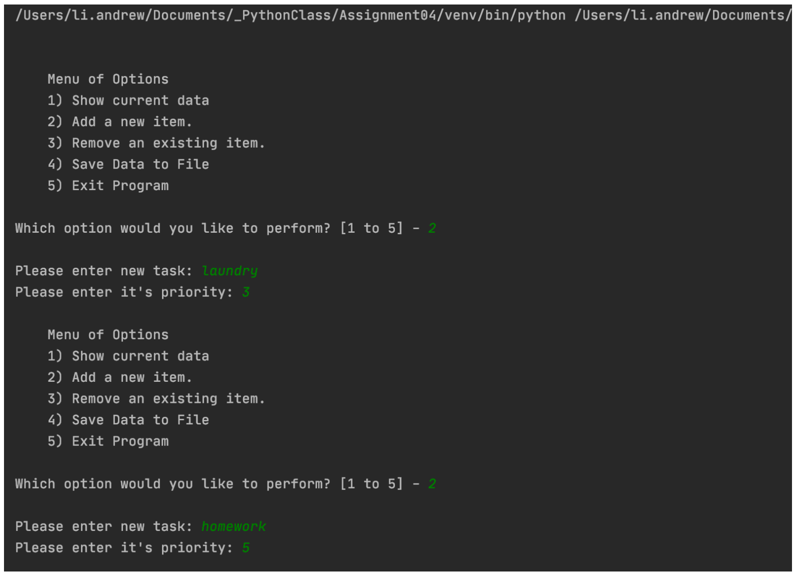


```
# Step 7 - Exit program
elif (strChoice.strip() == '5'):
    # TODO: Add Code Here
    break  # and Exit the program
```

Figure 9: Breaks loop and ends program.

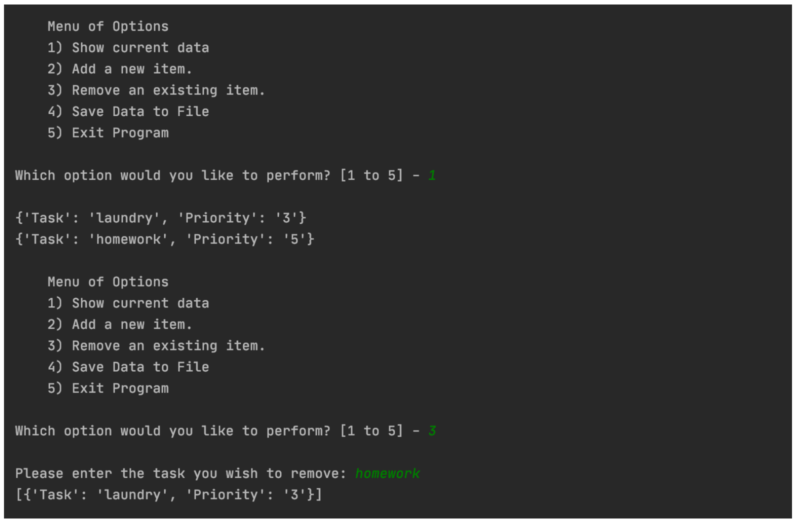
Running the Program

After creating the script, we verified the program is working properly by running it in PyCharm (Figure 10a-c) and the terminal command window on macOS (Figure 11a-b).



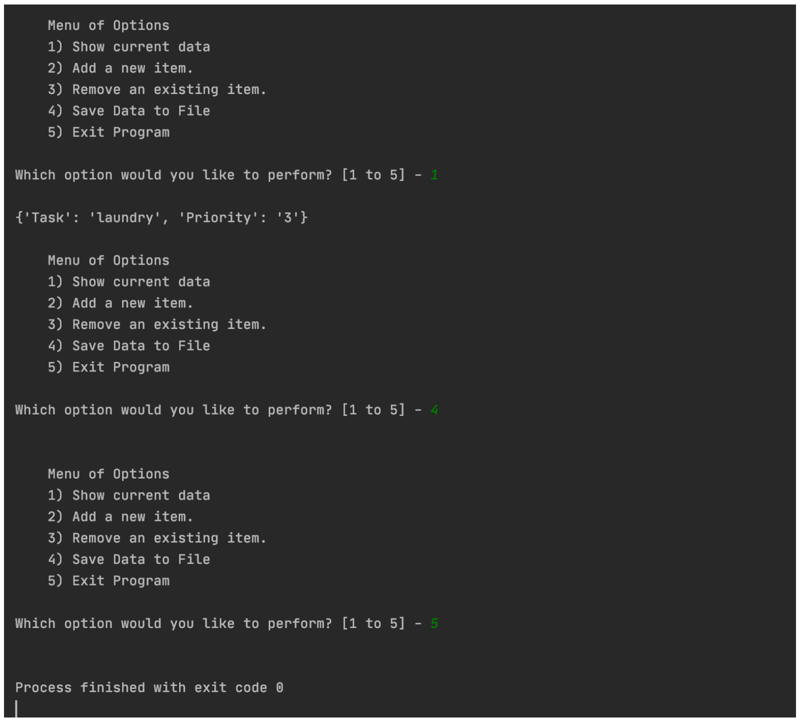
```
/Users/li.andrew/Documents/_PythonClass/Assignment04/venv/bin/python /Users/li.andrew/Documents/


    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 2

Please enter new task: laundry
Please enter it's priority: 3

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 2

Please enter new task: homework
Please enter it's priority: 5
```

Figure 10a: The results after running our program in the PyCharm IDE (1 of 3).



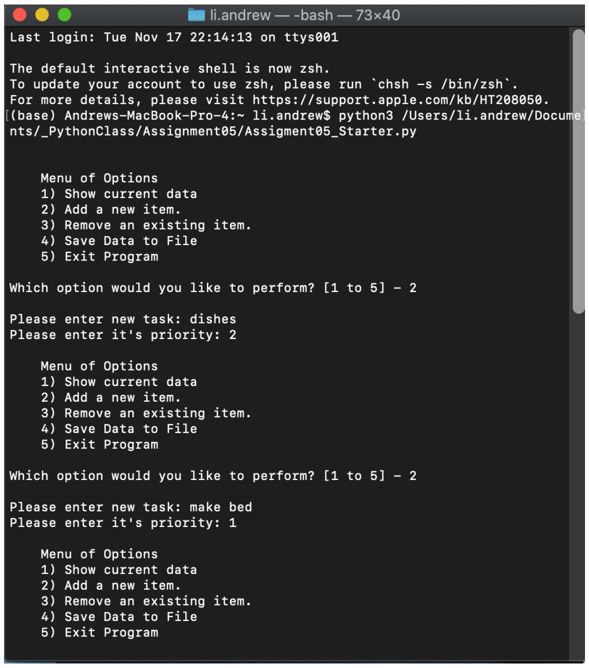
```
    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 1

{'Task': 'laundry', 'Priority': '3'}
{'Task': 'homework', 'Priority': '5'}

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 3

Please enter the task you wish to remove: homework
[{'Task': 'laundry', 'Priority': '3'}]
```

Figure 10b: The results after running our program in the PyCharm IDE (2 of 3).



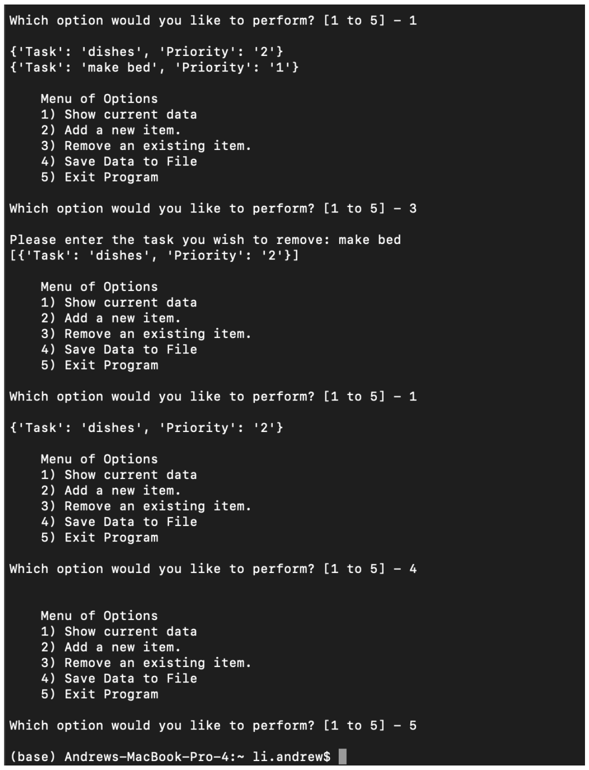
```
    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 1

{'Task': 'laundry', 'Priority': '3'}

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 4


    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 5


Process finished with exit code 0
```

Figure 10c: The results after running our program in the PyCharm IDE (3 of 3).



```
Last login: Tue Nov 17 22:14:13 on ttys001

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
(base) Andrews-MacBook-Pro-4:~ li.andrew$ python3 /Users/li.andrew/Docume
nts/_PythonClass/Assignment05/Assigment05_Starter.py


    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 2

Please enter new task: dishes
Please enter it's priority: 2

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 2

Please enter new task: make bed
Please enter it's priority: 1

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program
```

Figure 11a: The results after running our program in the terminal command shell (1 of 2) .



```
Which option would you like to perform? [1 to 5] - 1

{'Task': 'dishes', 'Priority': '2'}
{'Task': 'make bed', 'Priority': '1'}

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 3

Please enter the task you wish to remove: make bed
[{'Task': 'dishes', 'Priority': '2'}]

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 1

{'Task': 'dishes', 'Priority': '2'}

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 4


    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 5

(base) Andrews-MacBook-Pro-4:~ li.andrew$ 
```

Figure 11b: The results after running our program in the terminal command shell (2 of 2) .

Summary

We were able to complete the script that the originator had started. The originator had already outlined the steps with what they wanted to be accomplished within the script. We defined the variables and constants early, which we used throughout the program and loaded the “ToDoList.txt” back in the memory. We were able to then incorporate the menu and comments provided to us by the originator to set up how each choice in the menu would respond and process it. Once the program had finish each choice (other that choice 5, exiting the program), it would loop back to the menu for the user. Each choice gave the user the option to view, add, remove, or save the data before exiting the program.