Emily Brown

August 11, 2022

IT FDN 110 B Su 22: Foundations of Programming: Python

Assignment05

# **Assignment 05**

**Introduction**

The purpose of Assignment 05 is to modify an existing code that allows a user to input a household task and its priority. Rather than write it to this file each time there is a user input, the data is stored within a dictionary which is stored within a list. I modified this script in PyCharm and utilized PyCharm and Terminal to ensure it runs properly. I am working on a Mac OS.

**Strategy**

In order to modify this code, I utilized the ability to store dictionaries within a list. To accomplish each subtask, I utilized for and while loops when needed. The script allows the user to see what they have included in the list, add or remove data, and save to a text file when finished making modifications. Once the user is ready to save their data, the script saves all list contents to a text file.

**Script**

Below are screenshots from PyCharm of the script for Assignment 05. Note the pseudocode used to organize and label each section of code. The first section of code is the change log, declaration of global variables, and the processing section. The processing section opens the existing text file and adds existing data in that file to the table within the code.

*Graphical user interface, text, application, email

Description automatically generated*

*Figure 1: First section of Python Script for Assignment 05*

In this next section of code, the first portion of the Input/Output section is displayed.

Graphical user interface, text, application, email

Description automatically generated

*Figure 2: First portion of Input/Output section of Python Script for Assignment 05*

Finally, the final section of code contains the rest of the Input/Output section.

**Graphical user interface, text, application, email

Description automatically generated**

*Figure 3: Final section of Input/Output portion of code in Assignment05*

**Results**

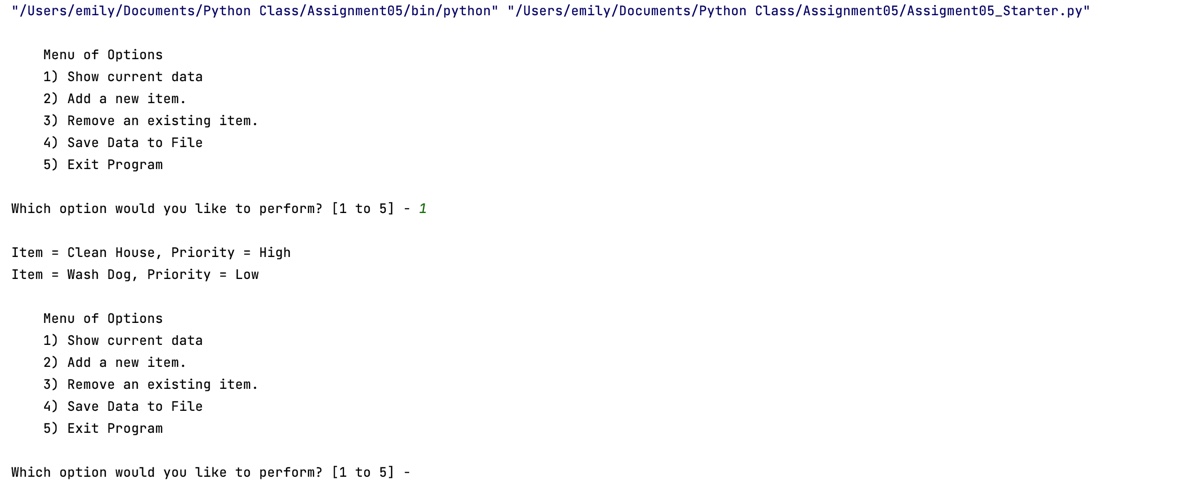
Below are screenshots of how the code presents itself when it is run in PyCharm. The first screenshot illustrates the text that existed in the text file ToDoList.txt before the code was run.

Graphical user interface, text

Description automatically generated

*Figure 4: Text file before running script*

The screenshot below is the first user interface that appears when the code is run. I selected option 1 to illustrate the existing content in the list. Note that it matches the content in the text file above. This is a result of the “Processing” section of the script.

**

*Figure 5: Initial user interface and Option 1*

Next, I selected option 2. The screenshot illustrates the first item that I added. I added two more items after.

Graphical user interface, text, application

Description automatically generated

*Figure 6: Option 2 run successfully*

Since I added three items, I chose Option 1 again to ensure they were added to the list. See below for the list at this point in the process.

A picture containing text

Description automatically generated

*Figure 7: List after Option 2*

I chose Option 3 next. I removed Vacuum from the list. Note that I chose Option 1 again to illustrate that the item had been removed successfully.

Graphical user interface, text

Description automatically generated

*Figure 8: Option 3 and list afterwards*

Finally, I chose Option 4 which saved this data to the text file. Then, I exited the program.

Text

Description automatically generated

*Figure 9: Option 4 and Option 5*

See below for the text file after Option 4 was run.

Graphical user interface, text

Description automatically generated

*Figure 10: Text file after Option 4*

The following screenshot illustrates the user interface when this code is run in Terminal. I added and removed items to illustrate that the changes were made successfully in Terminal.

Text

Description automatically generated

*Figure 11: Script in Terminal – Options 1, 2, and 3*

Below is the screenshot illustrating Options 4 and 5.

*Text

Description automatically generated*

*Figure 12: Script in Terminal – Options 4 and 5*

Finally, see the screenshot below of the text file after these changes.

Graphical user interface, text

Description automatically generated

*Figure 13: Text file after changes made in Terminal*

**Summary**

In summary, the Python script successfully achieves the objective of the assignment. It runs in both PyCharm and Terminal with the same results.