Jacob Bentzen

May 17st, 2022

Foundations Of Programming: Python

Assignment 5

Adding to a Code Template : Files, Lists, and Dictionaries

**Introduction**

In assignment 5, I was tasked with using a code template to create a program that allows the user to add, delete, and store a list of tasks using Python code. I will need to utilize lists and dictionaries to store and present the input data, as well as use my knowledge on writing and reading to and from txt files.

**Understanding the Template**

Finding what is already completed:

First, I will need to run through the code template and find out what I need to complete. As fig. 1 shows, much of the header and the data section has been completed.

Text

Description automatically generated

*Fig. 1: Header and data section of the template.*

It looks like there are plenty of variables already declared that are ready to be used in the rest of the code.

Speaking of the rest of the code, fig. 2 shows what is expected to be completed.

Text

Description automatically generated

*Fig. 2: Shows the to be completed portion of the code template.*

Basically, the template is requesting each step of the program to be filled out individually under each number, as well as an initial request to load the data from an existing text file.

Loading the Existing Data from the File:

As you can see in Fig. 3, I utilized a try except block to check if the file was created and then used a for loop to run through the file and process the data.

Text

Description automatically generated

*Fig. 3: Shows the code used to process the existing data file into a list for the program*

The code checks to see if the file is created, and if not tells the user that it is not created. It also takes the data and changes the format to put it into a dictionary.

Viewing and Writing Data:

Step two and three allow the user to see the data and to add items to the data list (Fig 4).

Text

Description automatically generated

*Fig. 4: Viewing the data using a dictionary and adding items to the table using .append*

The already written code presents the user with the choice menu and uses their input to select the choice. Choice 1 allows the user to view the current data in the table. I simply did a print function of each row in the main list so that the user could see all the tasks/priorities in the data.

Choice 2 allows the user to enter their own task and priority using the input command. This input is converted to the dictionary format and then added to the list.

Removing Data:

Choice 3 allows the user to remove a task of their choice from the list of tasks (Fig. 5).

Text

Description automatically generated

*Fig. 5: Removing the task using .remove*

The code uses .remove if the input matches a row in the list, and has an else clause for if the row is not found. This prints out the current task list so the user can see why their task was not on the list.

Saving the Data to File and Exiting:

Choice 4 allow the user to write the current data to a txt file and choice 5 allow the user to exit the program (Fig. 6).

Text

Description automatically generated

*Fig. 6: Saving the list by writing to the txt file, and exiting the program using break*

The list is saved by writing the lstTable to the file via a for loop. I made sure that the format matches what is necessary to re-read the file when the program is opened again (I.E. separating the data using a comma).

The exit from the program is a simple break from the while loop.

**Testing the Script**

Using PyCharm:

Since the program was written in PyCharm, it is straightforward to run and test the completed program (Fig. 7).

Text

Description automatically generated

*Fig. 7: Testing the program in the PyCharm run window.*

Using the Command Window:

After running the script in PyCharm, I made sure to test it using the Command Window on my computer (Fig. 8).

Shape, rectangle

Description automatically generated

*Fig. 8: Command Window with the Home Inventory Program completed.*

**Summary**

I was able to complete this program by utilizing my understanding of dictionaries, lists, and writing data to files. Working on this assignment has been very helpful in strengthening my foundational knowledge of writing to files and utilizing lists and dictionaries for data movement and storage.