Jason Killmer

2/22/2021

IT FDN 110

Assignment 6

**Assignment 6**

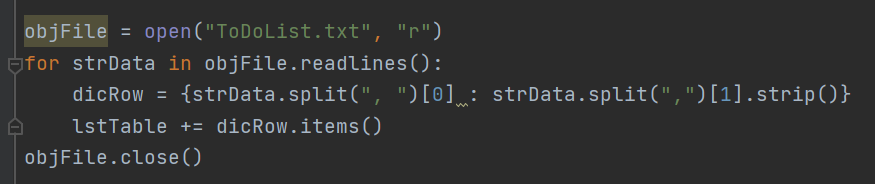
**Introduction**

This assignment combines elements of all previous assignments, and adds the additional use of functions and classes to organize data. Functions can be used to break up sections of a program so that each function handles a specific task. Functions can then be organized into classes, which handle high-level parts of the program.

The goal of the assignment is to both read and write to a file, all while incorporating elements of functions and classes.

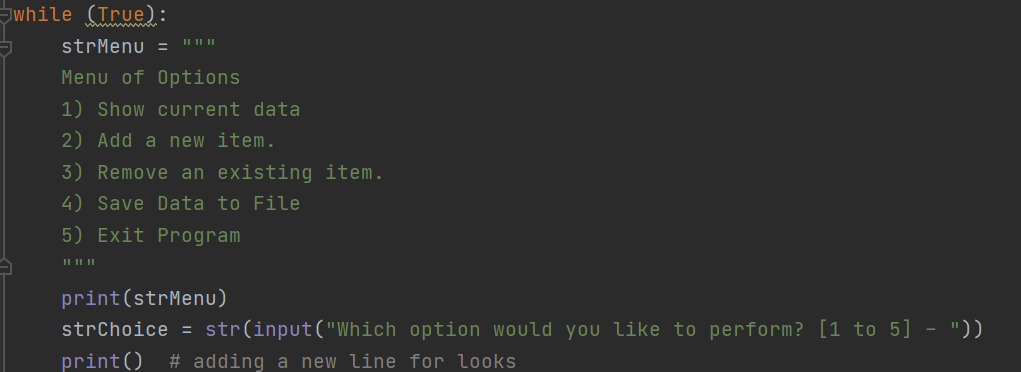
**Coding Steps**

Information from the text file is first read into the program, where the data is fed into a table that consists of rows of dictionaries. This is accomplished using the open command with the second passed string being “r” for “read”. It is important to use the close command after the data is read to ensure data can be written to the file in a later section.



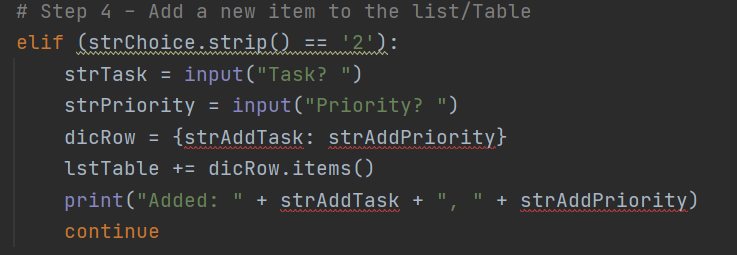
***Figure 1*** *Reading Data into the Program*

The main program structure is built around an initial “menu of options”, which operates under a while loop that will loop infinitely until a “break” command is used. The “break” command exits the loop when used, and is useful under an if/else statement to exit the program under a defined condition.



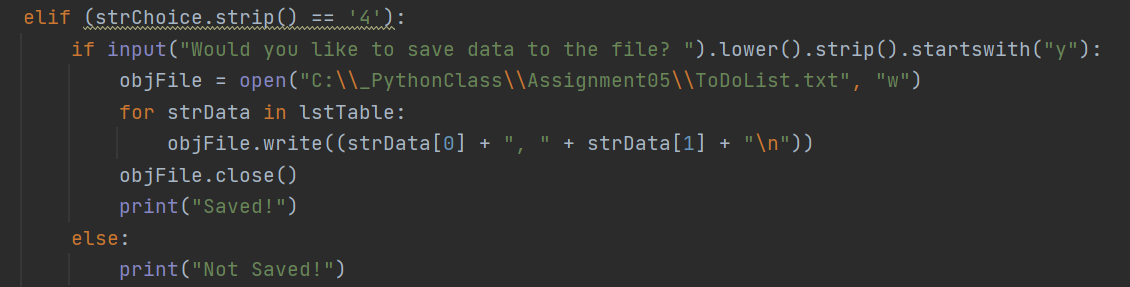
***Figure 2:*** *Infinite while loop with Menu of Options*

The user’s input is read and the program acts according to the if/else conditions, determining which course of action based on the user’s choice. The user’s data is stored in a two- dimensional list (a list of dictionaries of strings), which could be directly translated into a table. The “rows” (dictionaries of strings) can be added to the Table list either via the += operator or the append() function.



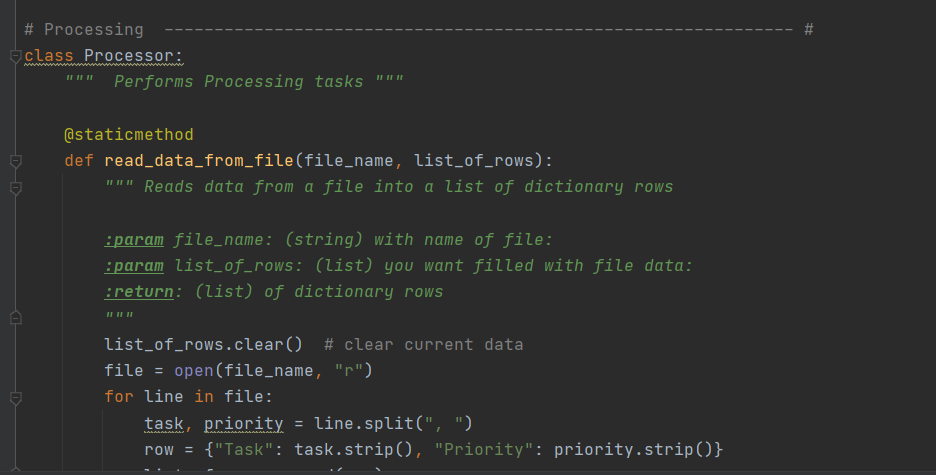
***Figure 3:*** *Adding Dictionaries to Lists*

The last step in the program is to write to the text file, in much the same way this was done in Assignment 3: open(“file”), .write(“string”), and .close(). A file should always be closed when not needed to avoid unintended writing to it, as this command prevents any future write commands during the current program execution.

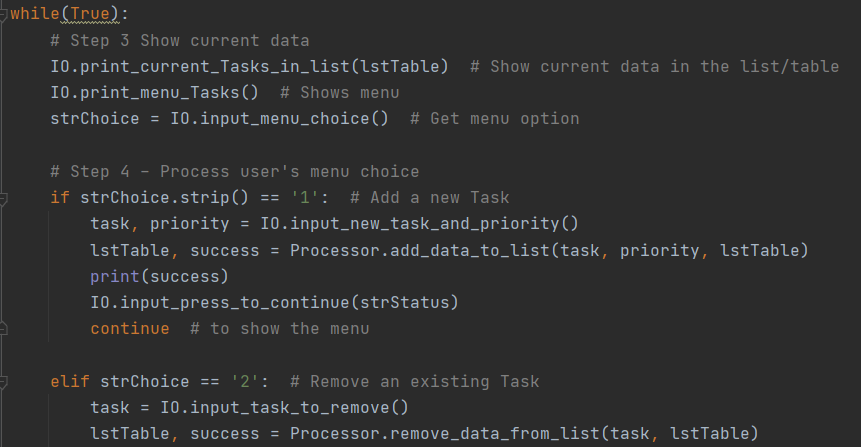
**

***Figure 5:*** *Writing to File*

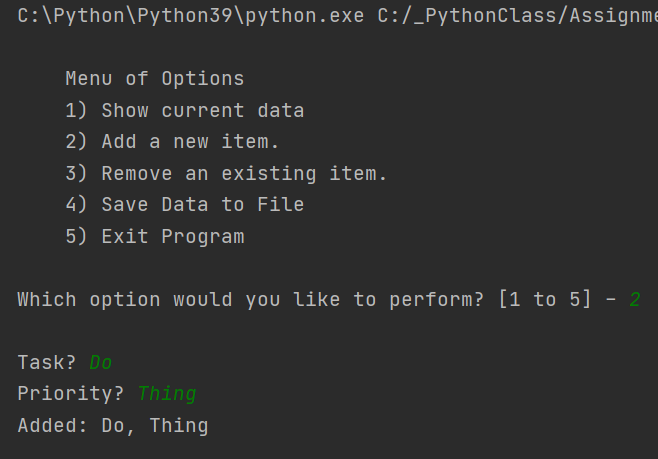
The program is sectioned off into classes: Processor, IO, and the main body. Processor is used to read data from the file, edit the list, and save data back into the file. IO is used to interface with the user, asking questions and taking in user responses. The main body of the program calls sections of these two classes as it moves through the program. No actual IO or processing is done in the main, it is only used to call the other functions.

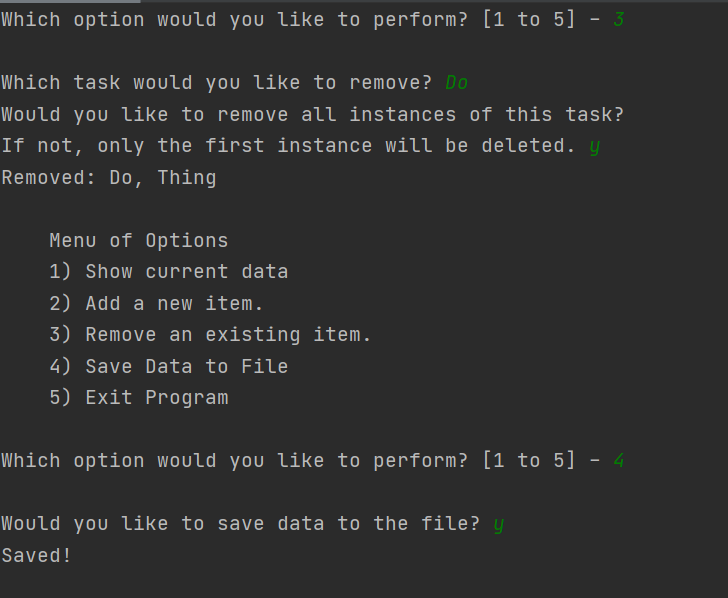
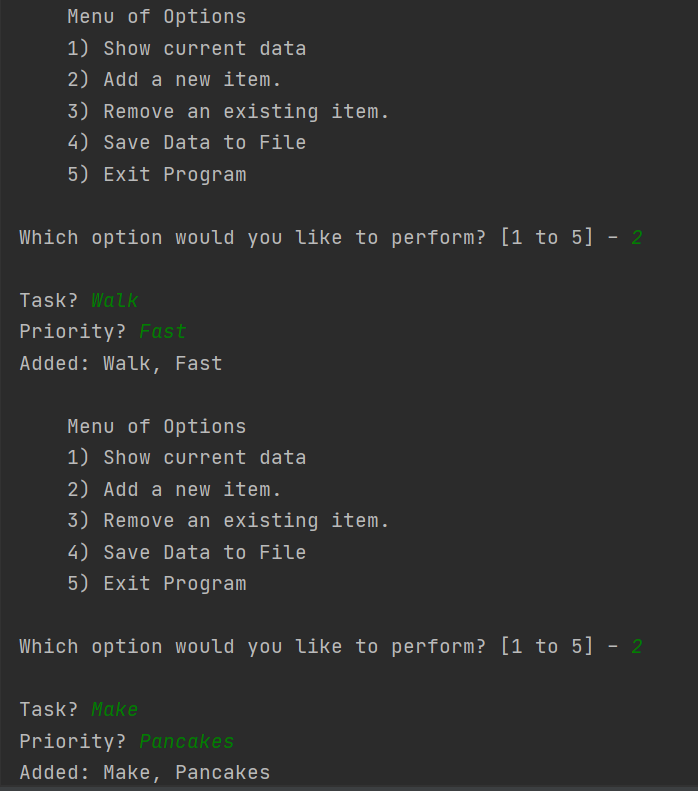
**

***Figure 6:*** *Processor Class*

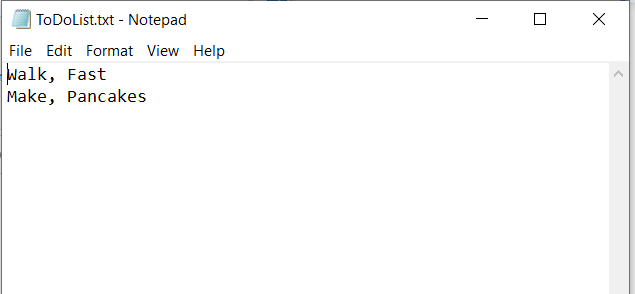
**

***Figure 7:*** *Program Main*

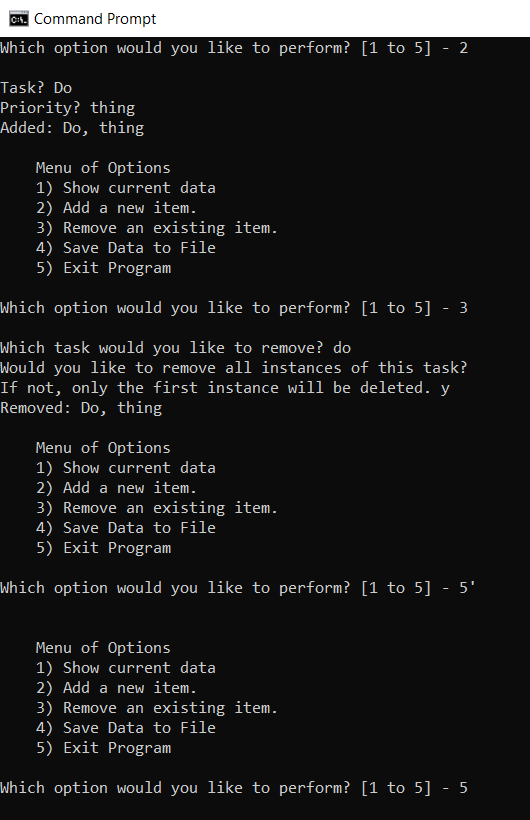




***Figure 6:*** *PyCharm Output*

**

***Figure 7:*** *PyCharm Output to Text*



***Figure 8:*** *Console Output*

**Conclusion**

This program was an exercise in the functions of writing to a file, lists, dictionaries, loops, and if statements. It combines different tools that have been learned on previous assignments into one program and ties them together to show how they can be used in conjunction with one another.