Name: Victoria O'Laughlin
Date: February 13, 2023
Course: IT FDN 110 A

Assignment 05 - Intro to Programming

Introduction

This document discusses the significance of simple code when writing a program that has many moving parts, including menu options and saving data to a file. My script also practices the separation of concerns to split my script into sections of data, processing, and presentation. It helped to define variables at the top of the script to see all the players at the beginning as well.

Thought process while writing my script:

1. Using the Assignment05_Starter file, I took a good look at the variables that were already declared, the pseudo code, and the loops / if-else statements. I considered what my approach would be with these tools in mind.
2. I was able to use labs 5-2 and 5-1 to get started with my script, and then I used the programming notes to find examples of code starting at step 4, when the script asks the user to add to the table
3. My first attempt at simply starting the script and opening the file became too messy and convoluted. In lab 5-2, the script starts out by writing existing data to a file, so I coded two dicRows with existing data to add to the ToDoList.txt file. This also required the user to enter W, R, or "next". It was doing too much. I took a look at the assignment 05 review video and gathered some tips on simplifying

the code. See Figure 1.0 for how my code looked for step 1 before I simplified it.

```
# Step 1 - When the program starts, load any data you have
# in a text file called ToDoList.txt into a python list of dictionaries rows (like Lab 5-2)

# Get user Input
while(True):
    print("Write or Read file data, then enter 'next' to proceed: ")
    strChoice = input("Choose to [W]rite or [R]ead data, then enter 'next': ")

    # Process the data
    if (strChoice.lower() == 'next'): break
    elif (strChoice.lower() == 'w'):
        # List to File
        objFile = open(strFile, "w")
        dicRow = {"task": "Make appointment", "priority": "medium"}
        objFile.write(dicRow["task"] + ',' + dicRow["priority"] + '\n')
        dicRow = {"task": "Write a script", "priority": "high"}
        objFile.write(dicRow["task"] + ',' + dicRow["priority"] + '\n')
        objFile.close()
    elif (strChoice.lower() == 'r'):
        # File to List
        objFile = open(strFile, "r")
        for row in objFile:
            lstRow = row.split(",") # Returns a list!
            dicRow = {"task": lstRow[0], "priority": lstRow[1].strip()} #strip gets rid of \n
            print(dicRow)
        objFile.close()
    else:
        print('Please enter either W, R, or done!')
        continue
```

Figure 1.0: Code was doing too much!

4. When it was time to ask the user to indicate which row they want to remove, I got stuck because I accidentally left the parentheses out of .strip. So when the user was inputting ‘3’ to indicate they want to a task something from the table, the script was just displaying the menu of options again. See Figure 1.1.
5. After putting the parentheses after .strip(), I reran the code and the correct next steps happened, where the program asks the user to input the task they’d like to remove. See Figure 1.2.
6. As I moved along in my script, I came across other lessons learned:
    a. I tried to change the keys of “task” and “priority” by adding a space before the p, like “ priority”, but it caused errors since this change wasn’t implemented consistently across my code. Consistency is crucial!

b. I got stumped for a while because I noticed that if I input ‘1’ to show current data, after I input ‘2’ to add a new item to the table, then showing current data will not show the data I just inputted because it hadn’t been saved yet. I thought I was doing something wrong until I realized that it’s just displaying the data currently displayed in the file, and not what the user inputted without saving it. The script won’t save on its own!

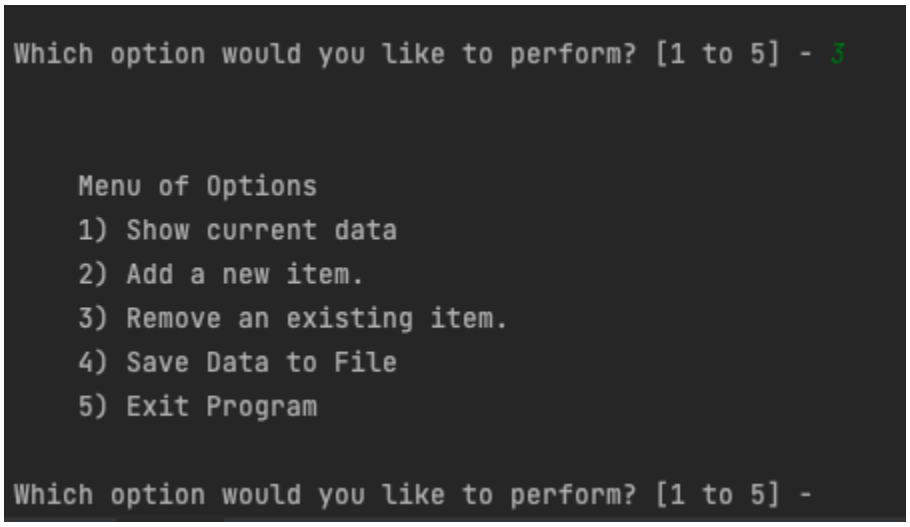

Figure 1.1: a user inputs ‘3’. Code: elif (strChoice.strip == '3')

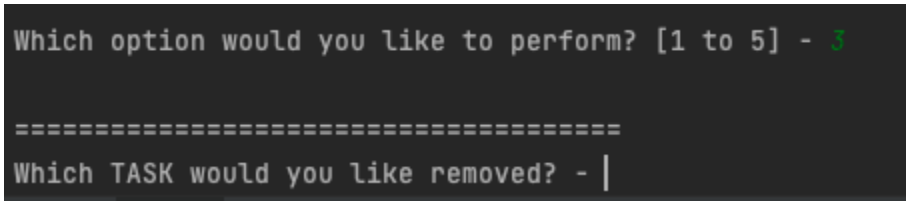

Figure 1.2: a user inputs ‘3’. Code: elif (strChoice.strip() == '3')

Summary

This week’s script had a lot of moving parts involved. This assignment will be a good point of reference moving forward as the use of loops, if-else statements, saving data to a file, and dictionaries all work together. I found this assignment challenging but I learned a lot from my mistakes.