Tim Shore
November 16, 2020
Foundations of Programming: Python
Assignment 05
<GitHub URL>

Creating a Task/Priority File

Introduction

This assignment has been the most complex so far this class. We needed to script a program that stored dictionary elements, called up those elements, added more dictionary elements, deleted dictionary elements by the user's choice, and saved all of those elements in a text file.

Scripting using Separation of Concerns (SoC)

This assignment was an import of a Python script file with sections to be completed that were commented as #TODO. The file was formatted using the design principle – separation of concerns. *"In computer science, separation of concerns (SoC) is a design principle for separating a computer program into distinct sections such that each section addresses a separate concern. A concern is a set of information that affects the code of a computer program."* https://en.wikipedia.org/wiki/Separation_of_conserns, 2020

For this Python file, the program script was divided into three sections:

- ✗ Data (declaration of variables and constants)
- ✗ Processing (performing tasks on data
- ✗ Presentation (Input/Output between user and program)

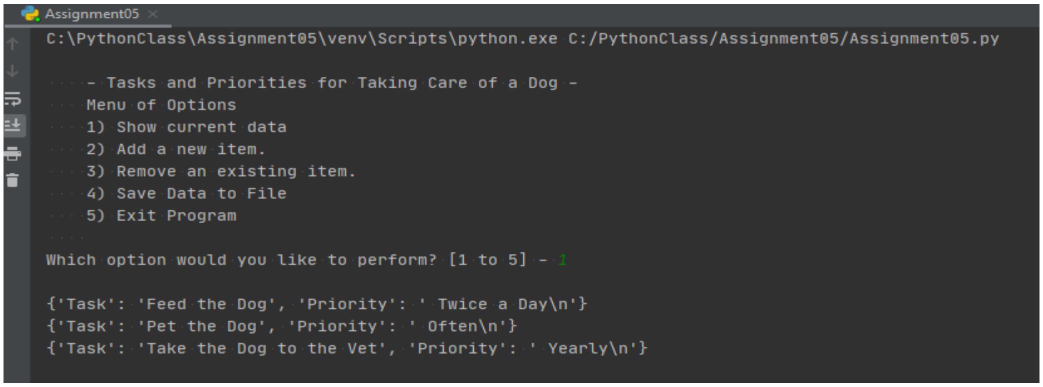
Using Dictionaries

The focus of this assignment, was to use dictionaries in the code. *"It's a lot like an actual dictionary where each entry is a pair: a word and its definition. When you look up a word, you get its definition. Python dictionaries work the same way: you look up a key and get its value."* Michael Dawson, Python Programming for Absolute Beginners, 2010. For the assignment, our keys were "Task" and "Priority" and the values were for us to decide.

Scripting the Assignment

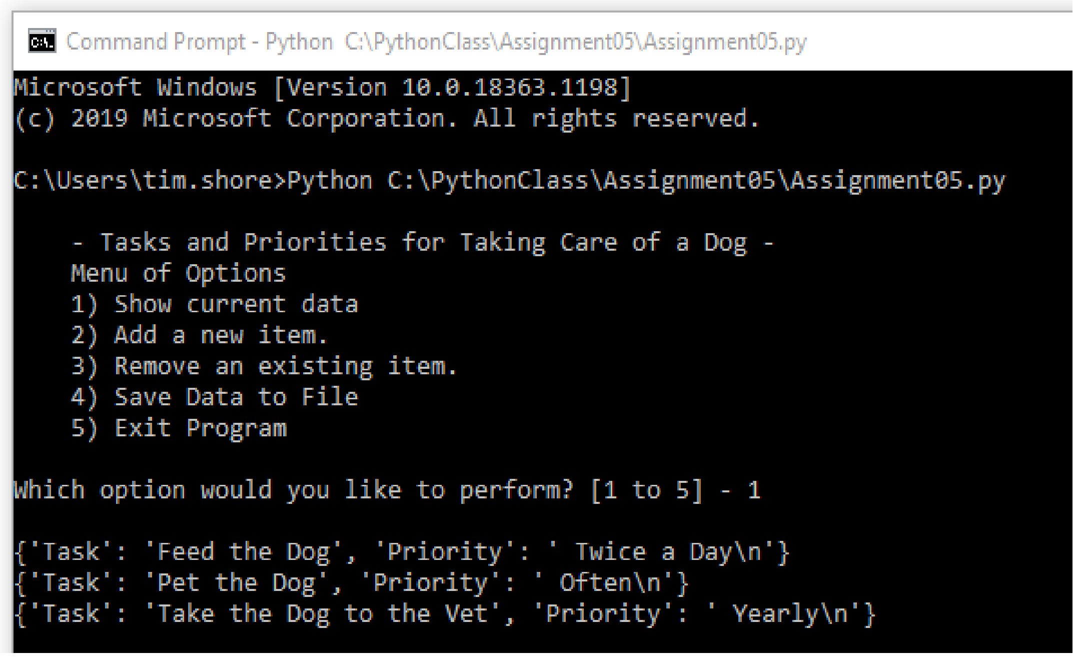
Beginning with a Text File

I decided to go with the important tasks that must be done when caring for a dog. With the data task, I started the program by scripting and loading a text file with three tasks and priorities in the processing section of my program. This way when the user chose the first option on the menu (Show current data), there would be data for them to see.



```
C:\PythonClass\Assignment05\venv\Scripts\python.exe C:/PythonClass/Assignment05/Assignment05.py

    - Tasks and Priorities for Taking Care of a Dog -
    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': 'Feed the Dog', 'Priority': ' Twice a Day\n'}
{'Task': 'Pet the Dog', 'Priority': ' Often\n'}
{'Task': 'Take the Dog to the Vet', 'Priority': ' Yearly\n'}
```

Figure 5.1: Loading of Tasks and Priorities in PyCharm



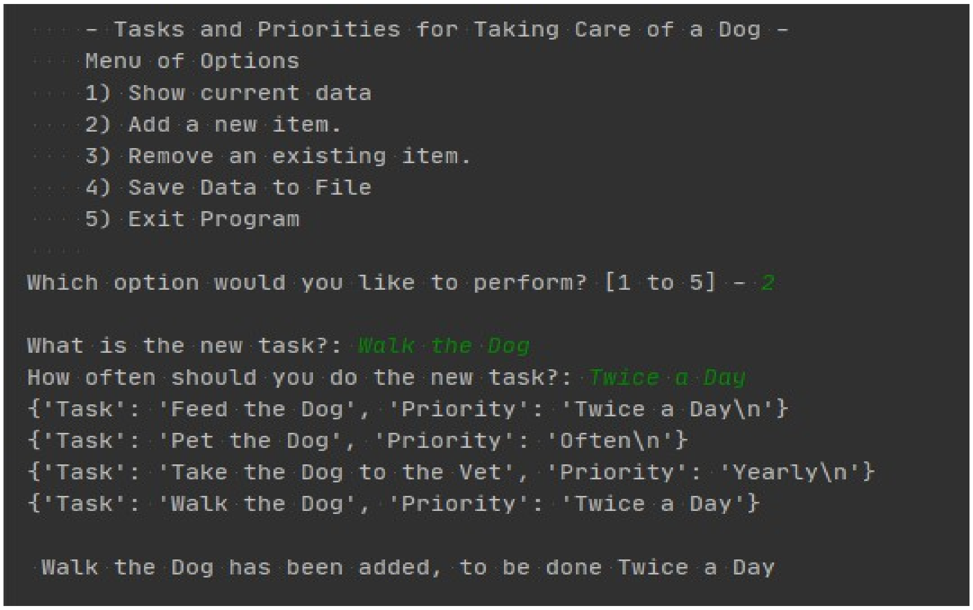
Command Prompt - Python C:\PythonClass\Assignment05\Assignment05.py

```
Microsoft Windows [Version 10.0.18363.1198]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\tim.shore>Python C:\PythonClass\Assignment05\Assignment05.py

    - Tasks and Priorities for Taking Care of a Dog -
    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': 'Feed the Dog', 'Priority': ' Twice a Day\n'}
{'Task': 'Pet the Dog', 'Priority': ' Often\n'}
{'Task': 'Take the Dog to the Vet', 'Priority': ' Yearly\n'}
```

Figure 5.2: Loading of Tasks and Priorities in Command Line

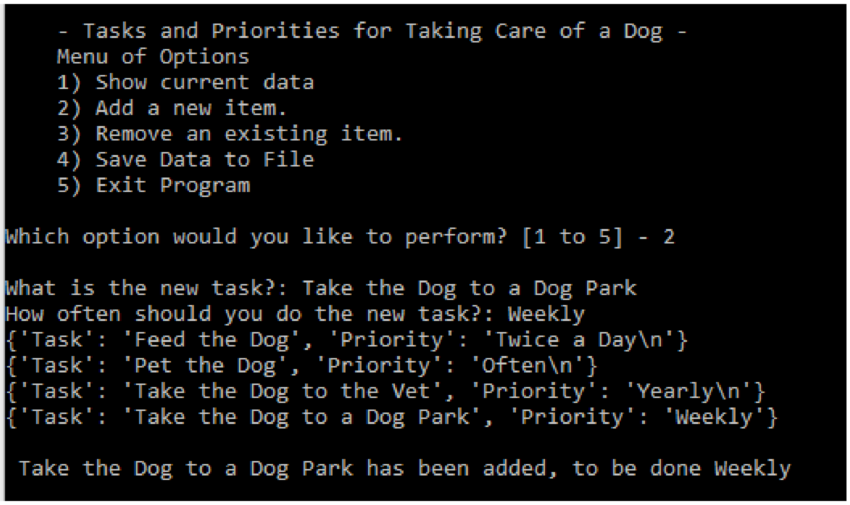
Adding an Item to the List

The next user option was to add a task and priority to the list. I used the variables “task” and “priority” to my values (declared them in my first SoC), had the user input the items, and saved them as an additional row as dictionary pairs. The code used the modifier “append” to the rest of the list stored in the variable “lstTable”.



```
    - Tasks and Priorities for Taking Care of a Dog -
    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

What is the new task?: Walk the Dog
How often should you do the new task?: Twice a Day
{'Task': 'Feed the Dog', 'Priority': 'Twice a Day\n'}
{'Task': 'Pet the Dog', 'Priority': 'Often\n'}
{'Task': 'Take the Dog to the Vet', 'Priority': 'Yearly\n'}
{'Task': 'Walk the Dog', 'Priority': 'Twice a Day'}

 Walk the Dog has been added, to be done Twice a Day
```

Figure 5.3: Adding a new task and priority to the data in PyCharm

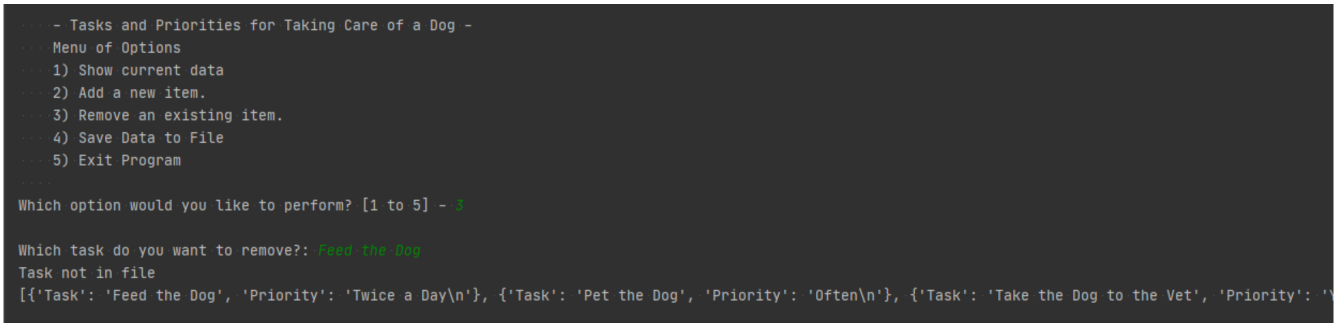


```
    - Tasks and Priorities for Taking Care of a Dog -
    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

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

What is the new task?: Take the Dog to a Dog Park
How often should you do the new task?: Weekly
{'Task': 'Feed the Dog', 'Priority': 'Twice a Day\n'}
{'Task': 'Pet the Dog', 'Priority': 'Often\n'}
{'Task': 'Take the Dog to the Vet', 'Priority': 'Yearly\n'}
{'Task': 'Take the Dog to a Dog Park', 'Priority': 'Weekly'}

 Take the Dog to a Dog Park has been added, to be done Weekly
```

Figure 5.4: Adding a new task and priority to the data in Command Line

Removing an Item from the List

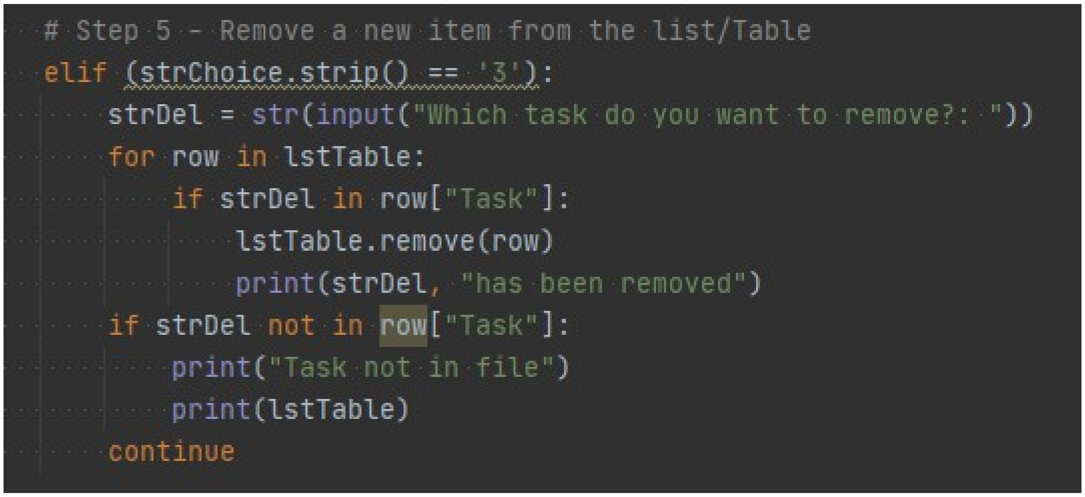
I've had the most difficulty with this part of the assignment. Sometimes I have a tendency to overthink the coding and find myself down rabbit holes I had no intention of going. I did manage to get the code to run without a syntax error, but it couldn't find the task that the user wanted to remove.



```
    - Tasks and Priorities for Taking Care of a Dog -
    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

Which task do you want to remove?: Feed the Dog
Task not in file
[{'Task': 'Feed the Dog', 'Priority': 'Twice a Day\n'}, {'Task': 'Pet the Dog', 'Priority': 'Often\n'}, {'Task': 'Take the Dog to the Vet', 'Priority': '
```

Figure 5.5: Code could not find task in PyCharm

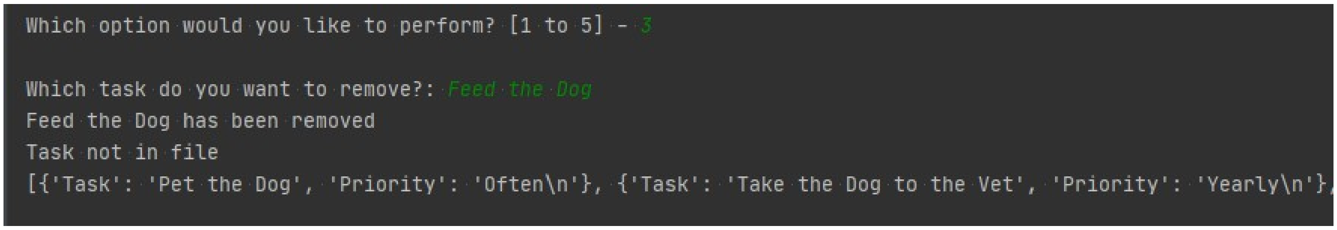
After much trial and error, I realized that I was trying to do more work for the program in coding than was necessary. I needed to focus on the elements I was trying to call and let the computer do the work that it needed to do.



```
    # Step 5 - Remove a new item from the list/Table
    elif (strChoice.strip() == '3'):
        strDel = str(input("Which task do you want to remove?: "))
        for row in lstTable:
            if strDel in row["Task"]:
                lstTable.remove(row)
                print(strDel, "has been removed")
        if strDel not in row["Task"]:
            print("Task not in file")
            print(lstTable)
        continue
```

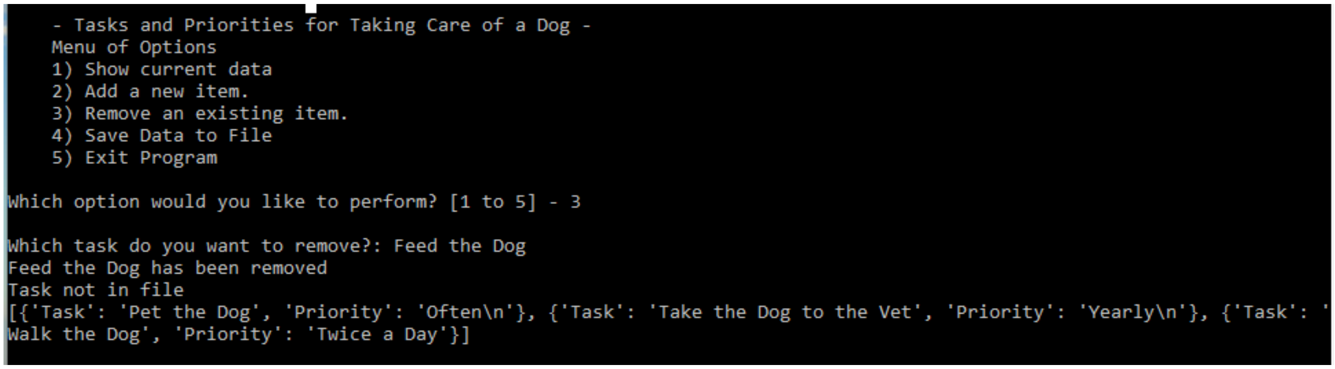
Figure 5.6: Correcting the Scripting in PyCharm

Now when the program runs, not only does it remove the task and priority from the list, but it also reaffirms that it's no longer there by not finding it when it runs the second 'if' statement. It's a little clunky, but it does the trick!



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

Which task do you want to remove?: Feed the Dog
Feed the Dog has been removed
Task not in file
[{'Task': 'Pet the Dog', 'Priority': 'Often\n'}, {'Task': 'Take the Dog to the Vet', 'Priority': 'Yearly\n'},
```

Figure 5.7: Task removed from program in PyCharm

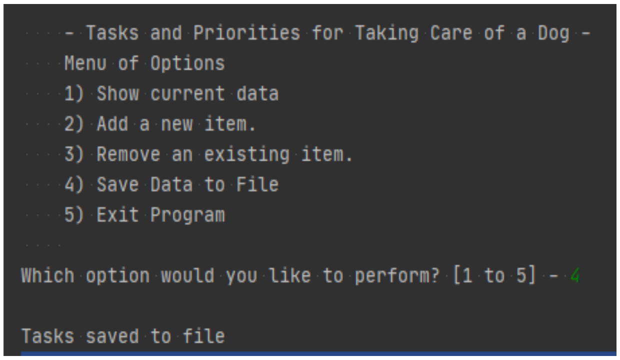


```
        - Tasks and Priorities for Taking Care of a Dog -
        Menu of Options
        1) Show current data
        2) Add a new item.
        3) Remove an existing item.
        4) Save Data to File
        5) Exit Program

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

Which task do you want to remove?: Feed the Dog
Feed the Dog has been removed
Task not in file
[{'Task': 'Pet the Dog', 'Priority': 'Often\n'}, {'Task': 'Take the Dog to the Vet', 'Priority': 'Yearly\n'}, {'Task': '
Walk the Dog', 'Priority': 'Twice a Day'}]
```

Figure 5.8: Task removed from program in Command Line

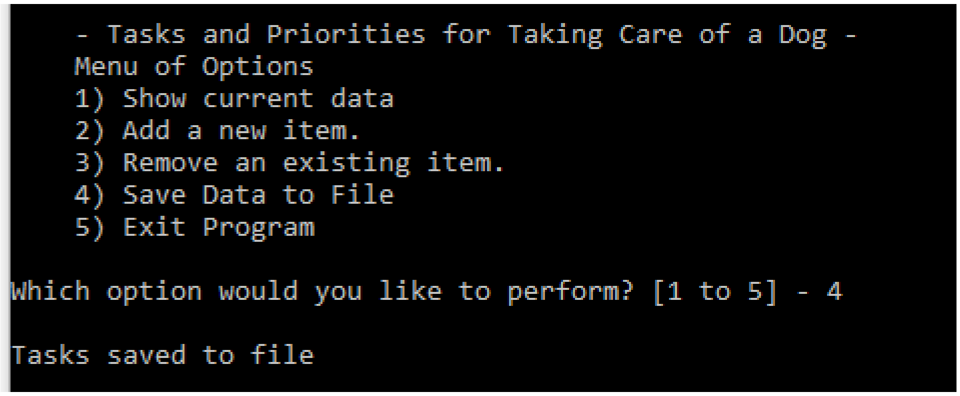
Saving to the Text File

For the final “TODO” for the assignment is to save the data to the text file. I had tried to use the “a” (append) operator when opening the text file in the code, but it never seemed to work. So by just having the program “w” (write) the entire file again and use the append module to add the additional data, it all worked just fine.



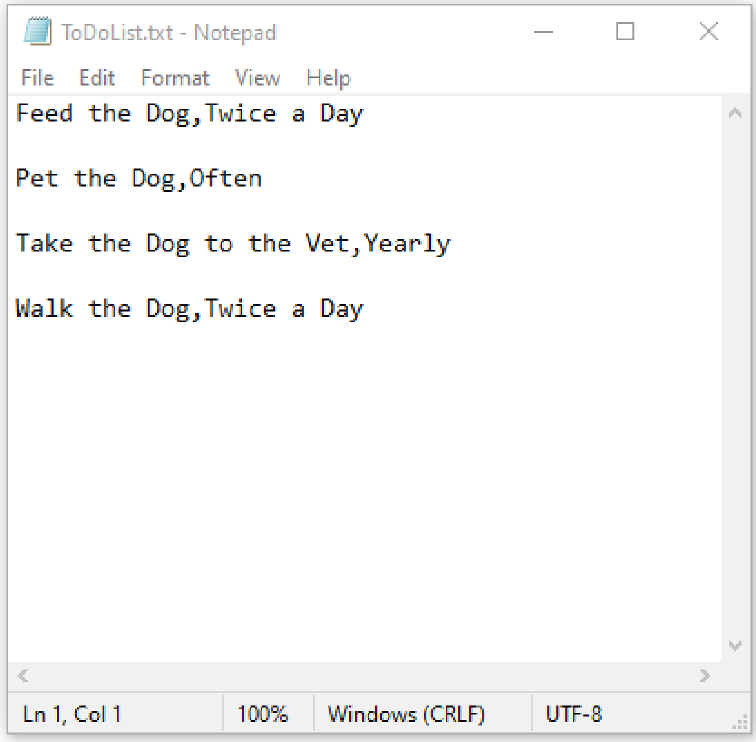
```
    - Tasks and Priorities for Taking Care of a Dog -
    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program
    
Which option would you like to perform? [1 to 5] - 4

Tasks saved to file
```

Figure 5.9: Task saved text file in PyCharm



```
    - Tasks and Priorities for Taking Care of a Dog -
    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

Tasks saved to file
```

Figure 5:10: Task saved to text file in Command Line



```
Feed the Dog,Twice a Day

Pet the Dog,Often

Take the Dog to the Vet,Yearly

Walk the Dog,Twice a Day
```

Figure 5:11: Data added to Text File

Summary

It's been a long road, but I believe I've completed the assignment successfully. I was able to create all the data as dictionaries and manipulate it not only in the program, but in the text file as well. Upward and onward with Python programming!