Patricia Coonrad

May 18, 2020

IT FDN 100 A Sp 20: Foundations Of Programming: Python

Assignment 05

ToDo List Script

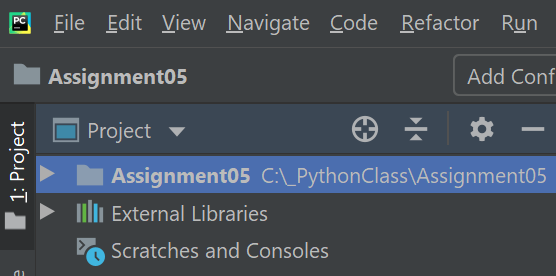
# Introduction

In this assignment I will describe the steps I took to create a Python program called ToDo List. Once the program is started, the system will load the contents of an existing ‘ToDoList.txt‘ file into memory. The ToDo List program will then perform different actions depending on the menu selection made by the user. The user will be able to view the existing tasks and assigned priorities, add new tasks and priorities to the list, remove existing tasks and priorities from the list, and save the changes back to the ‘ToDoList.txt‘ file. The program will give the option to continue performing the actions until the user decides to exit. In this program, I will demonstrate some of the concepts I learned during the course, such as script template, dictionary, key, read data from file into list, read data from a file into a dictionary, ‘Separations of Concerns’, GitHub, to name a few.

# Create the ToDo List Script

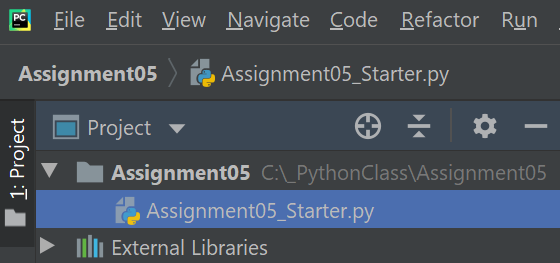
In this exercise, I will describe the steps I took to create the ToDo List program. I will then run the script and capture the image of the script execution using both PyCharm and the Command Window.

1. As per assignment directions, I created a new Project in PyCharm that uses the \_PythonClass\Assignment05 folder as its location (Figure 1).



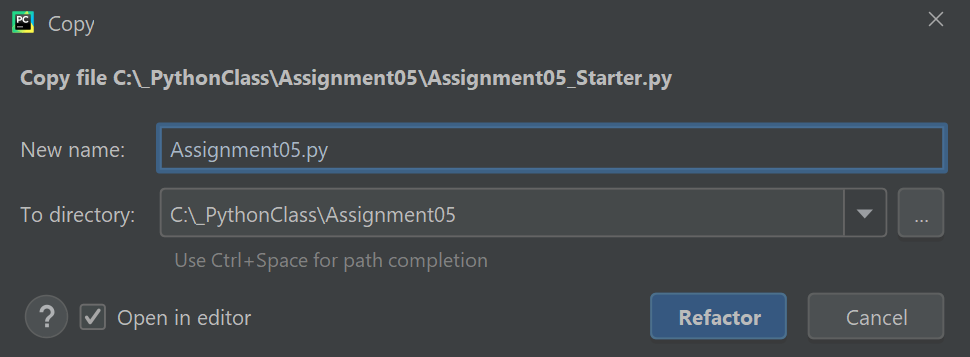
***Figure 1. Assignment05 PyCharm Project***

1. I added the starter file ‘Assignment05\_Starter.py’ to the project. This file was provided by instructor Randy Root and was a starting template I had to modify and use for my program (Figure 2).

****

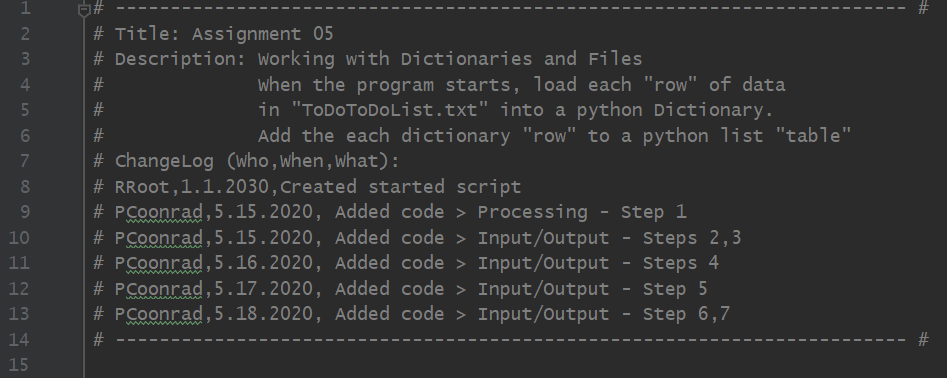
***Figure 2. Adding the ‘Assignment05\_Starter.py’ file to the project***

1. I made a copy of the template file and named it ‘Assignment05.py’ (Figure 3).



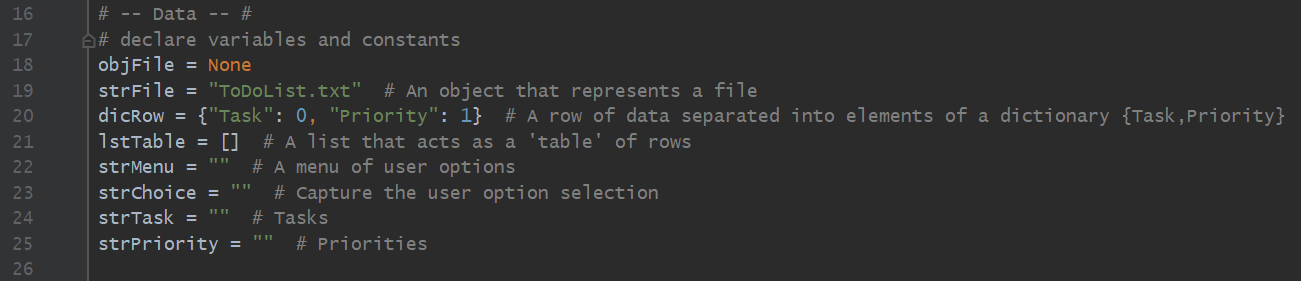
***Figure 3. Copying the starter file and naming ‘Assignment05.py’***

1. Once the ‘Assignment05.py’ file was created, I started modifying the script. I updated the script header change log to reflect the changes I made to the code. The script header is a comment at the beginning of the script. It includes information about the script such as title, brief description of the program, and change log. (Figure 4).



***Figure 4. Assignment05 Script header***

1. I updated the ‘Data’ section with a few additional variables. Even though declaring variables and constants at the top of the code is not considered a Pythonic practice, it becomes very helpful when you have lots of lines of code as it makes it easier to know what each value will be (Figure 5).

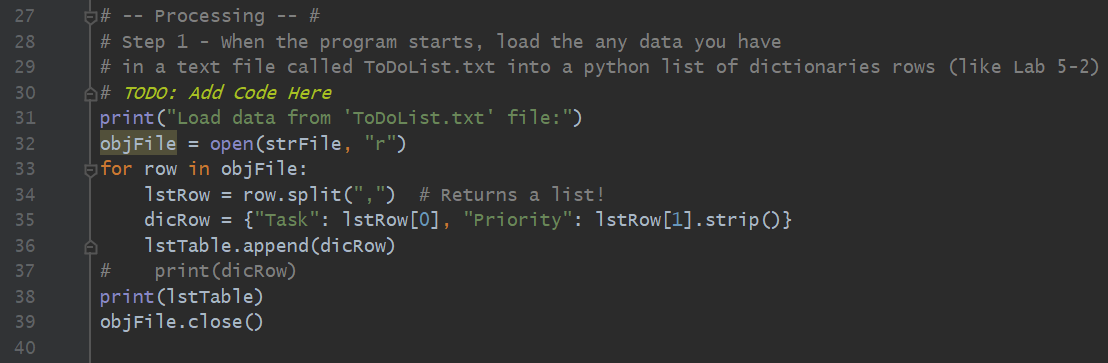


***Figure 5. Declaring variables and constants***

1. When the program starts, the first action is to load any data that is stored in the text file called ‘ToDoList.txt’. This information is loaded in memory and will be available for use through the execution of the program.

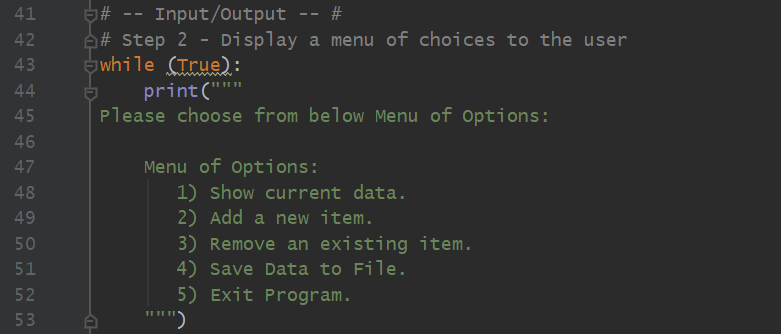
In order to access the information from the file, I used the open() function. I gave the name of the file I wanted to use, in this case ‘ToDoList.txt’, and the action I wanted it to perform once the file was opened. I used option “r” for read: collect data from the file and put it in memory so the program can access.

The ‘ToDoList.txt’ file has two columns of data, "Task" and "Priority". I used the for loop to loop through all rows in the file, read the data and from those columns, store them on a dictionary, add to a list and then display on the screen. Dictionaries allow you to store information in pairs: key and value. Once the action was complete, I used the close() function to close the ‘ToDoList.txt’ file (Figure 6).



***Figure 6. Reading the data from the ‘ToDoList.txt’ file***

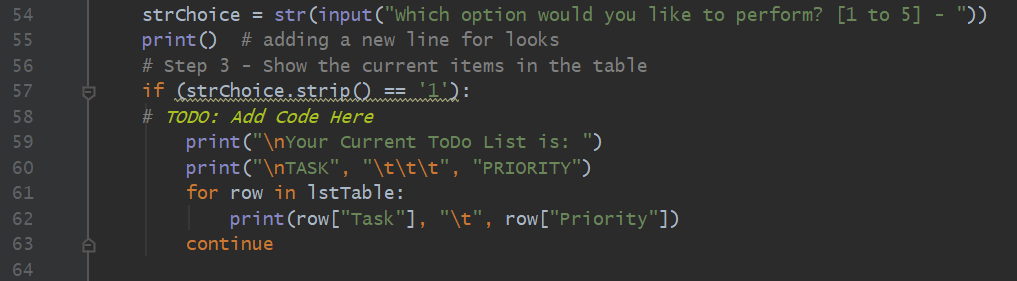
1. The print() function and the while loop were used in this step to display the menu of options to the user during the execution of the program. By using the while loop, the menu will be displayed and the program to continue doing an action until it is instructed to stop when the condition is no longer met (Figure 7).



***Figure 7. Displaying Menu of Options to the user using print() and while loop***

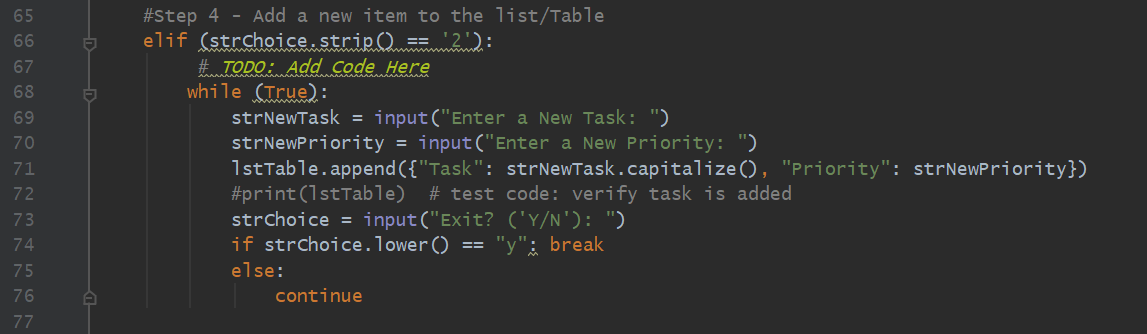
1. The program will ask the user to make a selection from the menu, and depending on the choice made, the code would perform a different action. The if statement was used to have the program branch to a section of the code or skip it, based on the set conditions.

When the user selects option (1) from the menu, the program will display the existing task and its priority to the user. I used a for loop to loop through the table of data and the keys “Task” and “Priority” to access the dictionary values (Figure 8).



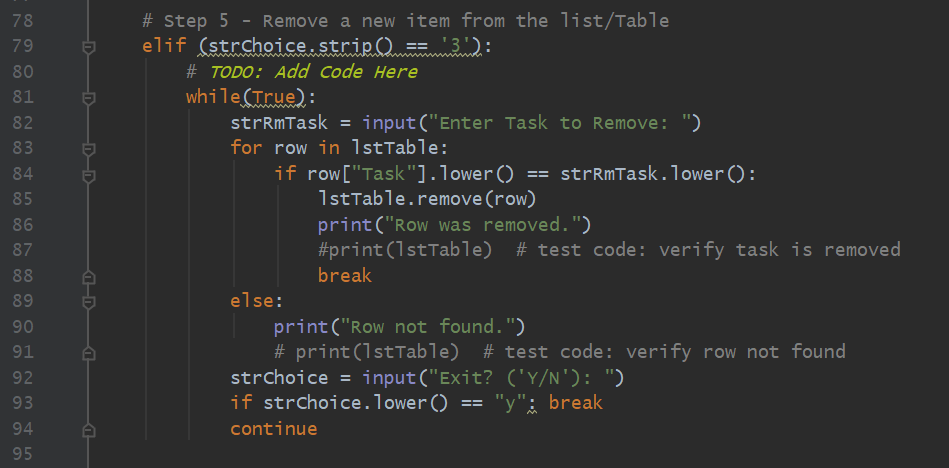
***Figure 8. Access dictionary values using keys***

1. When the user selects (2) from the menu options, the program will prompt the user to enter a new task and its priority. The new values will be appended to the table of data in memory. The program will give the user the option to exit and go back to the main menu or enter more tasks. (Figure 9).



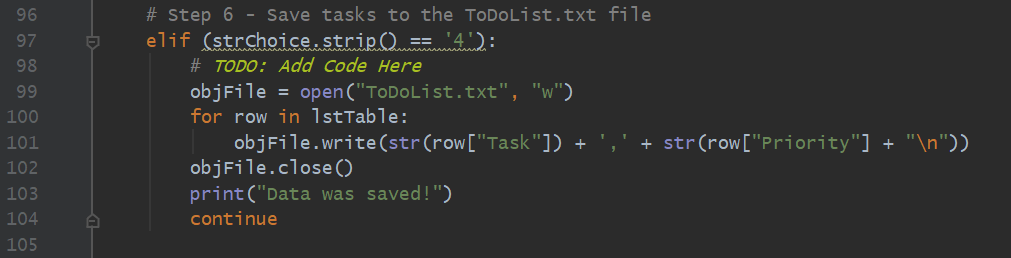
***Figure 9. User input to add data to the table in memory***

1. When the user selects option (3) from the menu, the program will ask if the user would like to remove a task and its priority from the list. It will prompt the user to enter the name of the task to be removed, then it will loop through the list and compare to if the value is in the list. If the value is found, then it is removed. If no value is found, a message is displayed to the user.



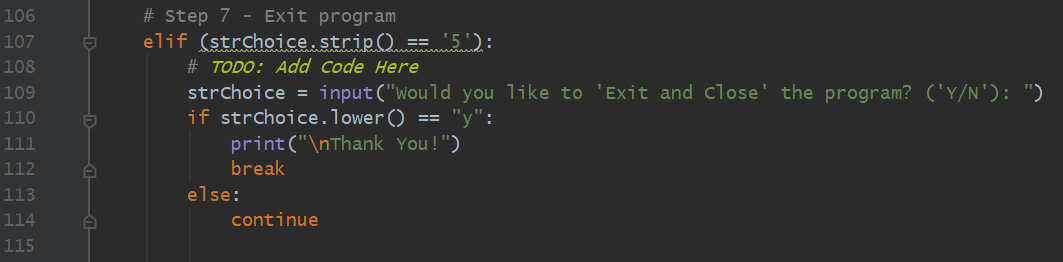
***Figure 10. User input to remove data from the table in memory***

1. All the changes made up to this point were in memory, meaning that no values were added to or removed from the ‘ToDoList.txt’ file. When the user selects option (4) from the menu, the program will then save the changes to the file and they will become permanent. I used the open() function selected the option (“w”) to write to the file, and then used the close() function to close the file (Figure 11).



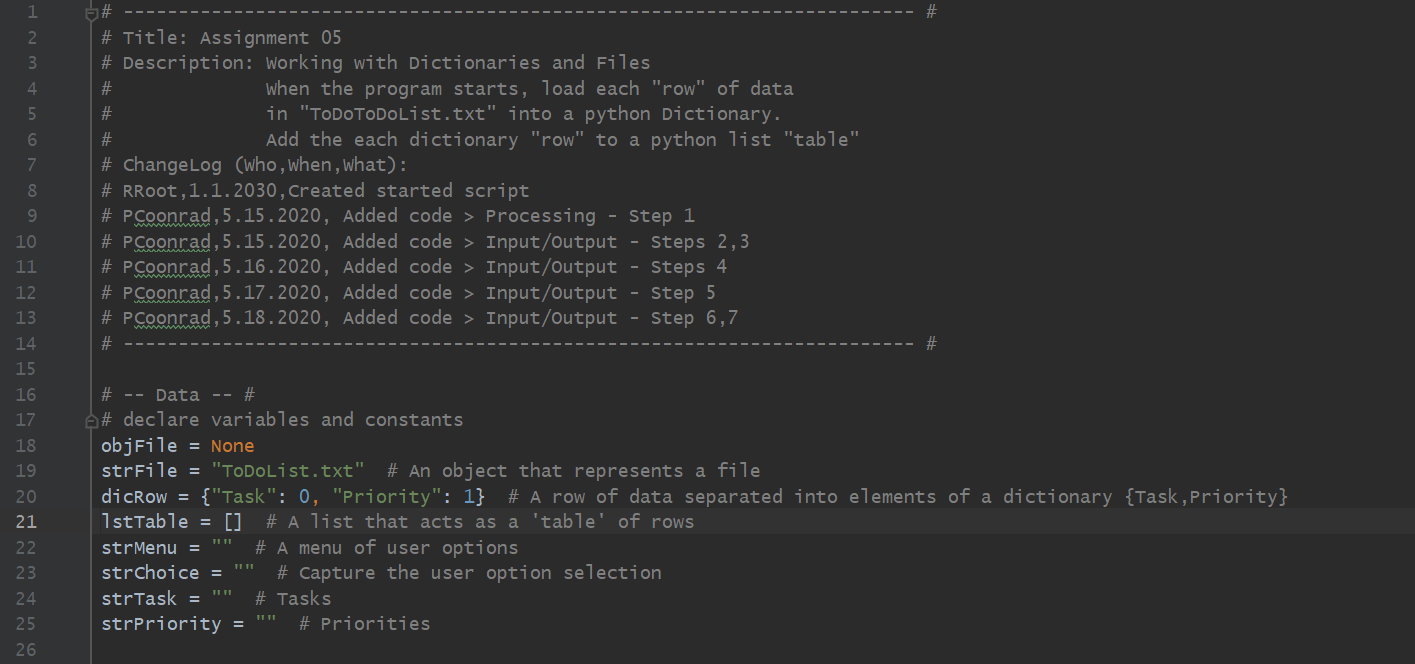
***Figure 11. Saving changes to the ‘ToDoList.txt’ file***

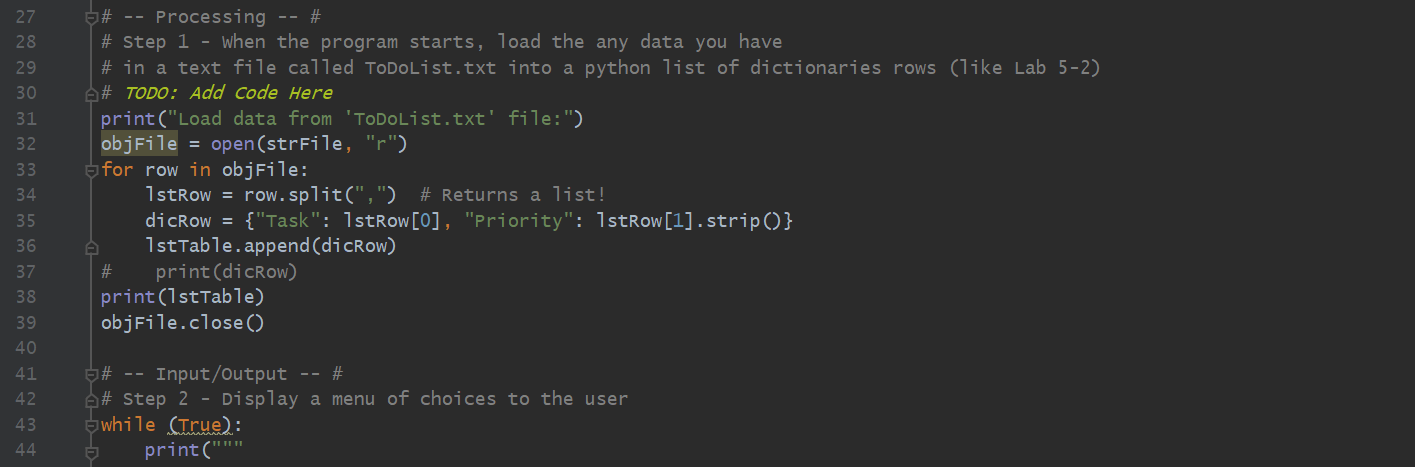
1. When the user selects option (5) from the menu, the program will prompt to confirm if the user want to exist and close the program or go back to the main menu (Figure 12).

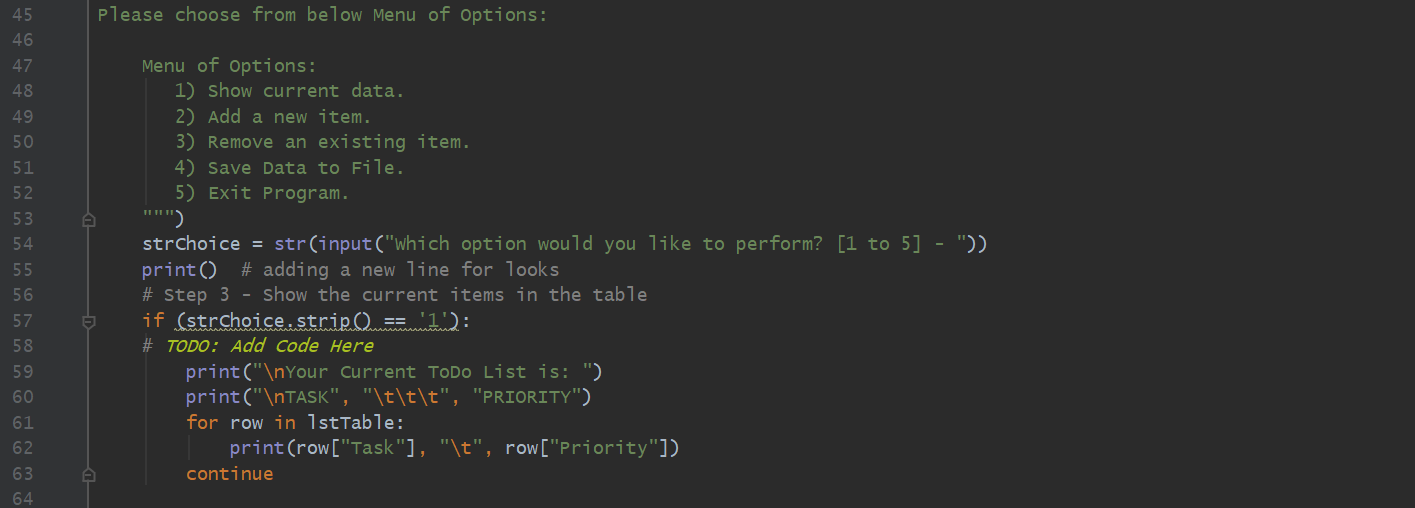


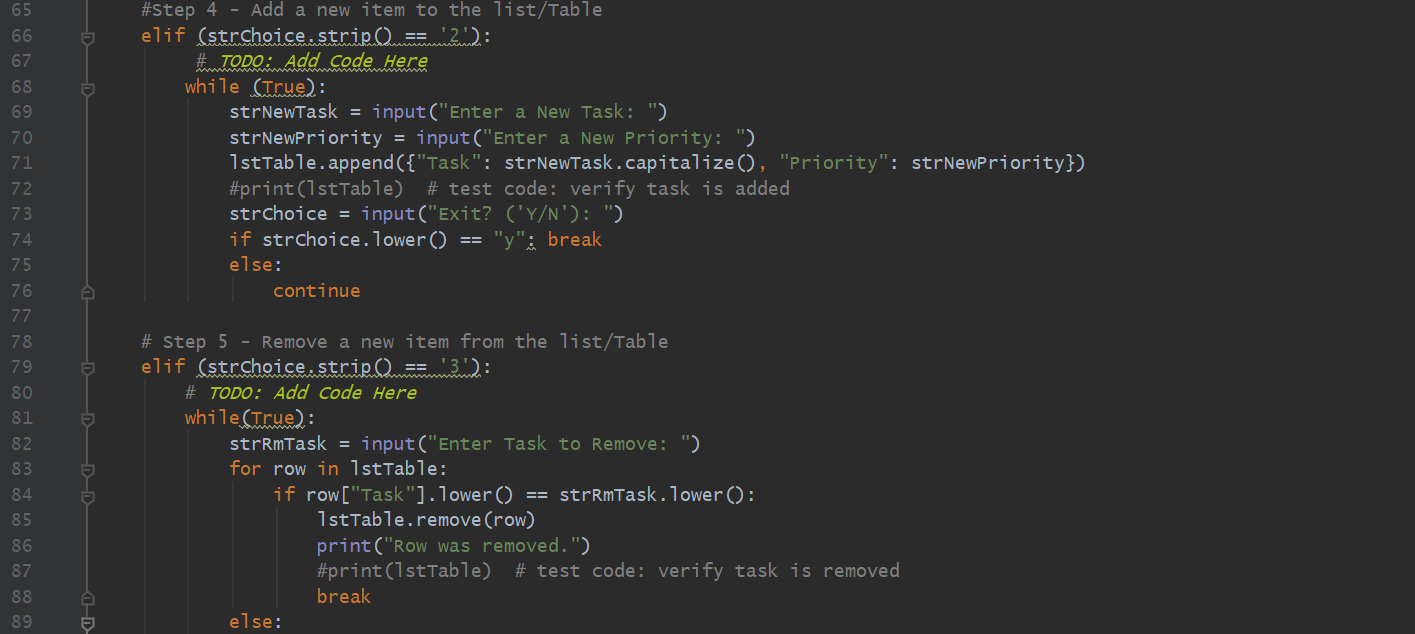
***Figure 12. Exiting the program***

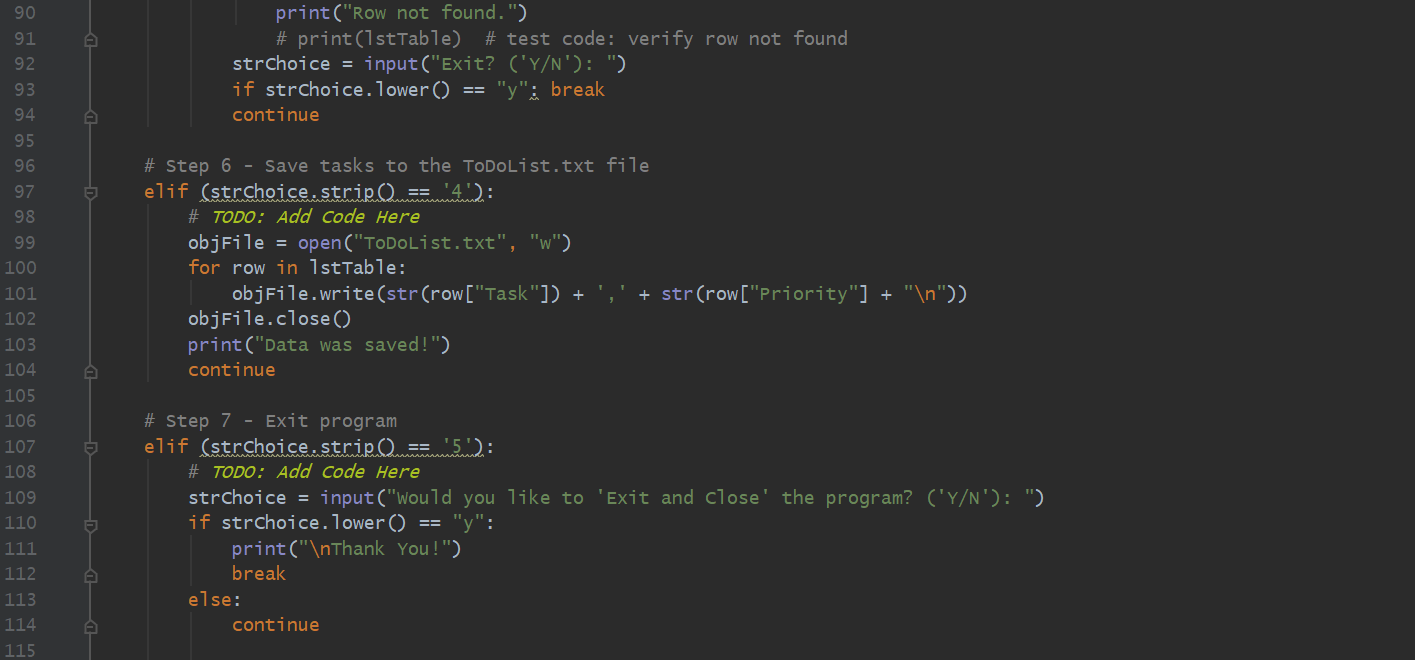
1. Below is a screenshot if the entire script in PyCharm (Figure 13).







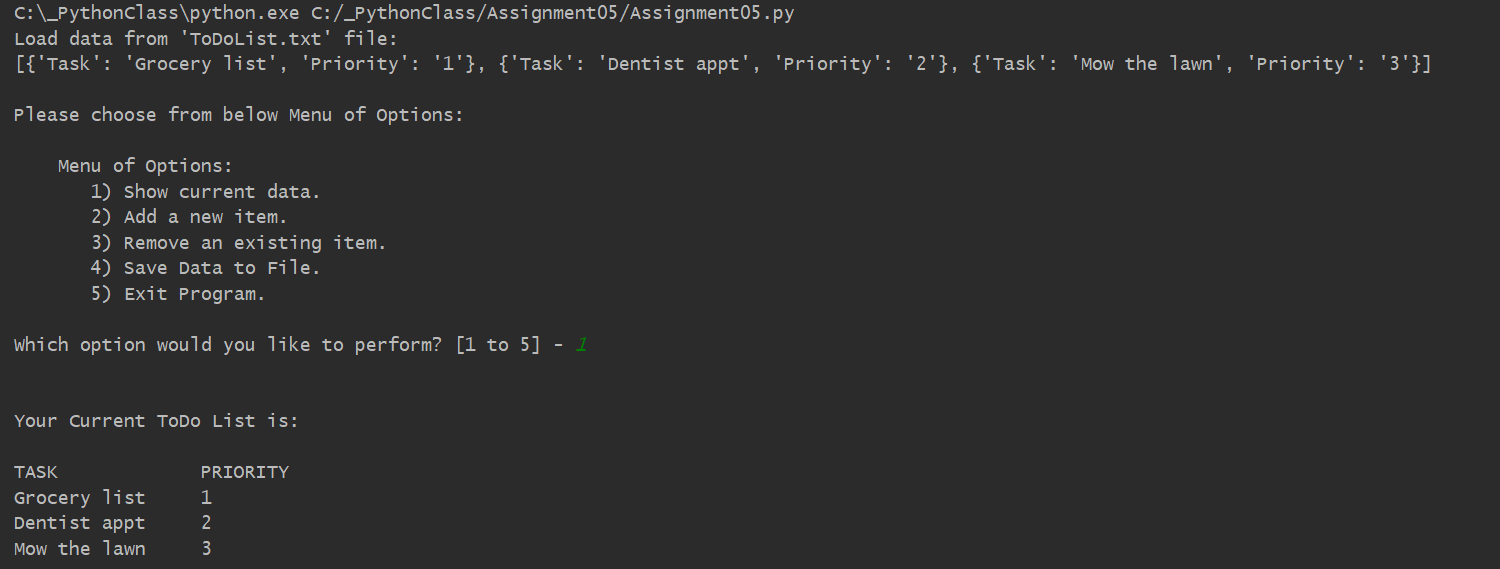


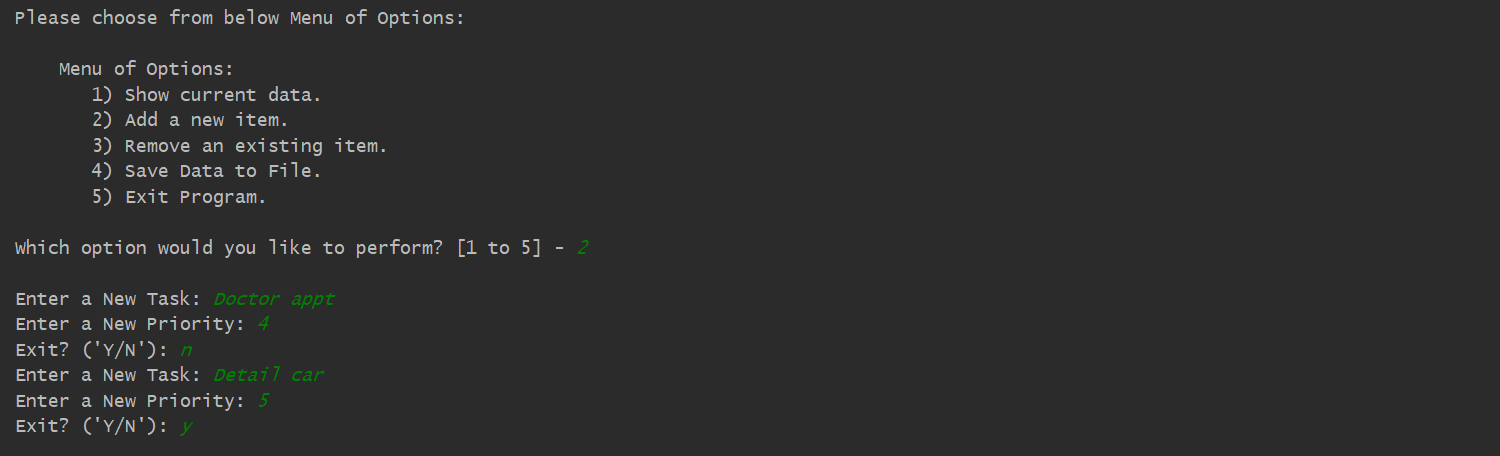


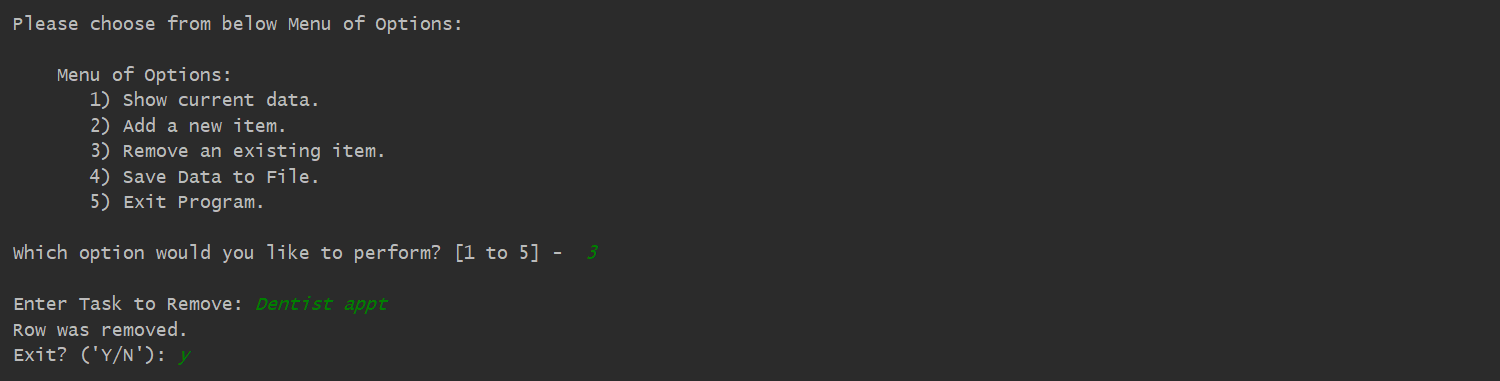
***Figure 13. Screenshot of the script***

# Running the ToDo List Program Using PyCharm

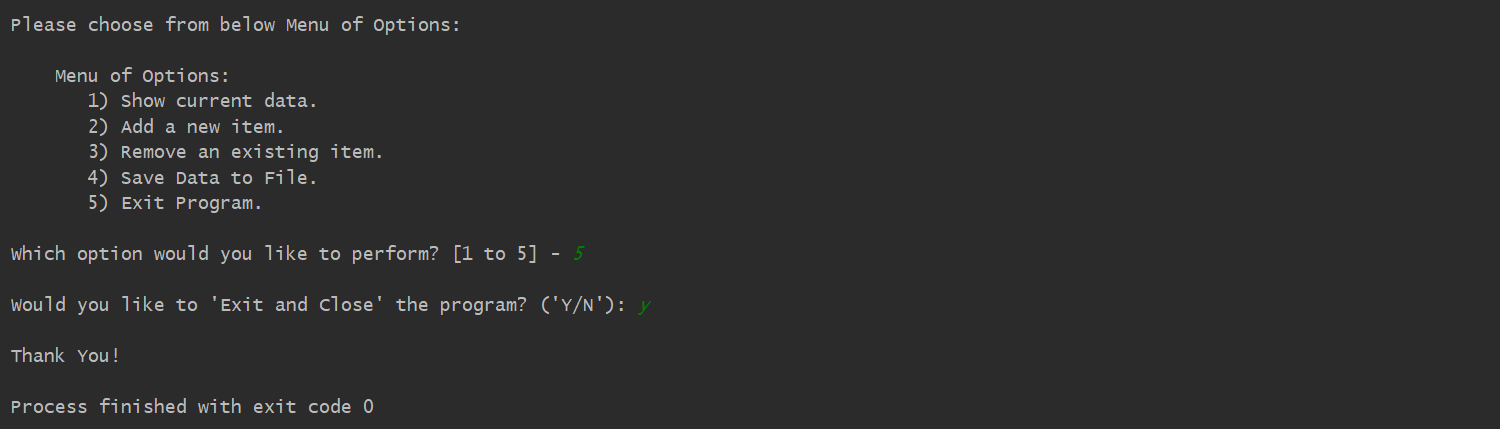
After the code was complete, I successfully ran the program using PyCharm (Figure 14).







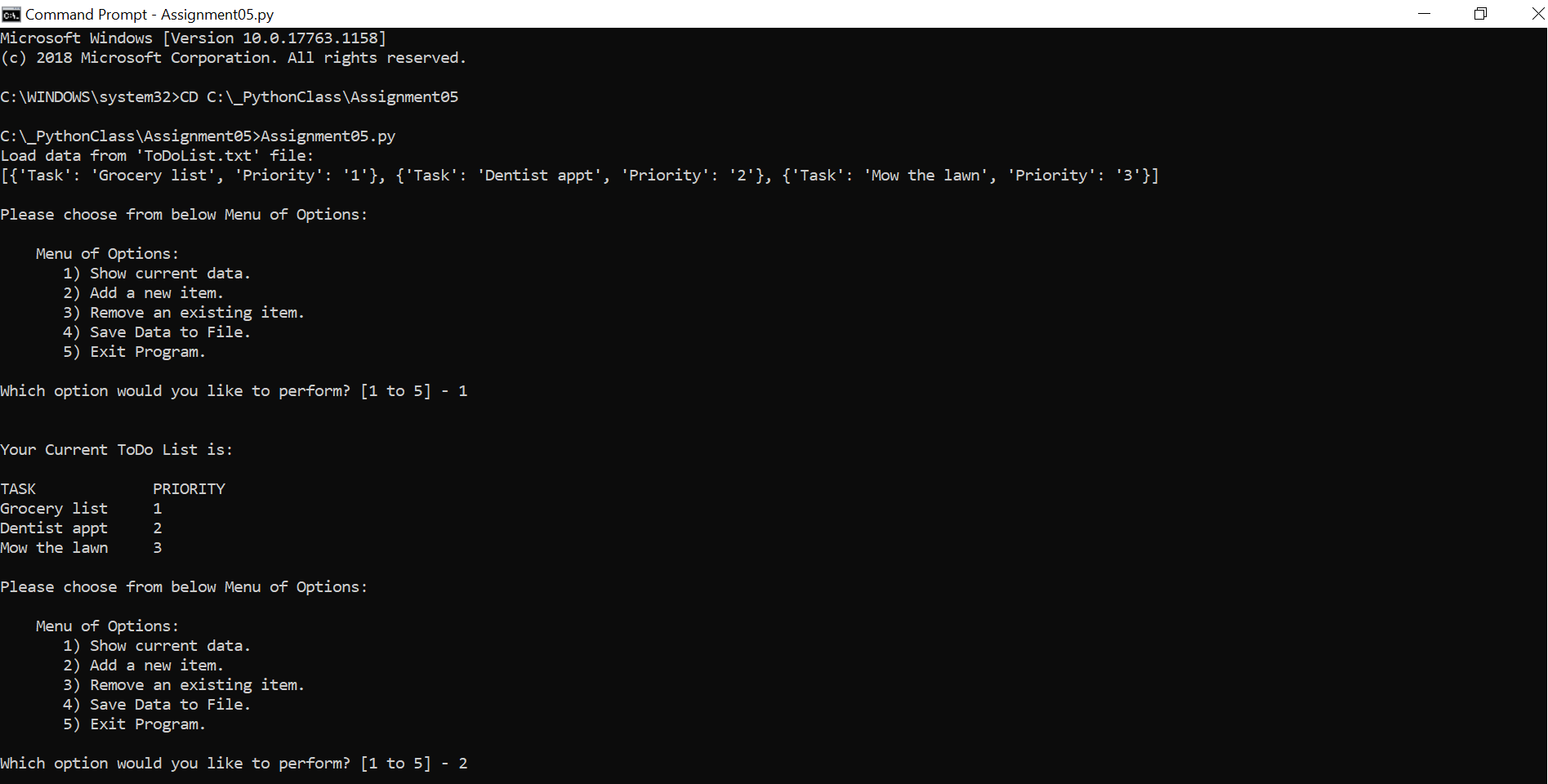


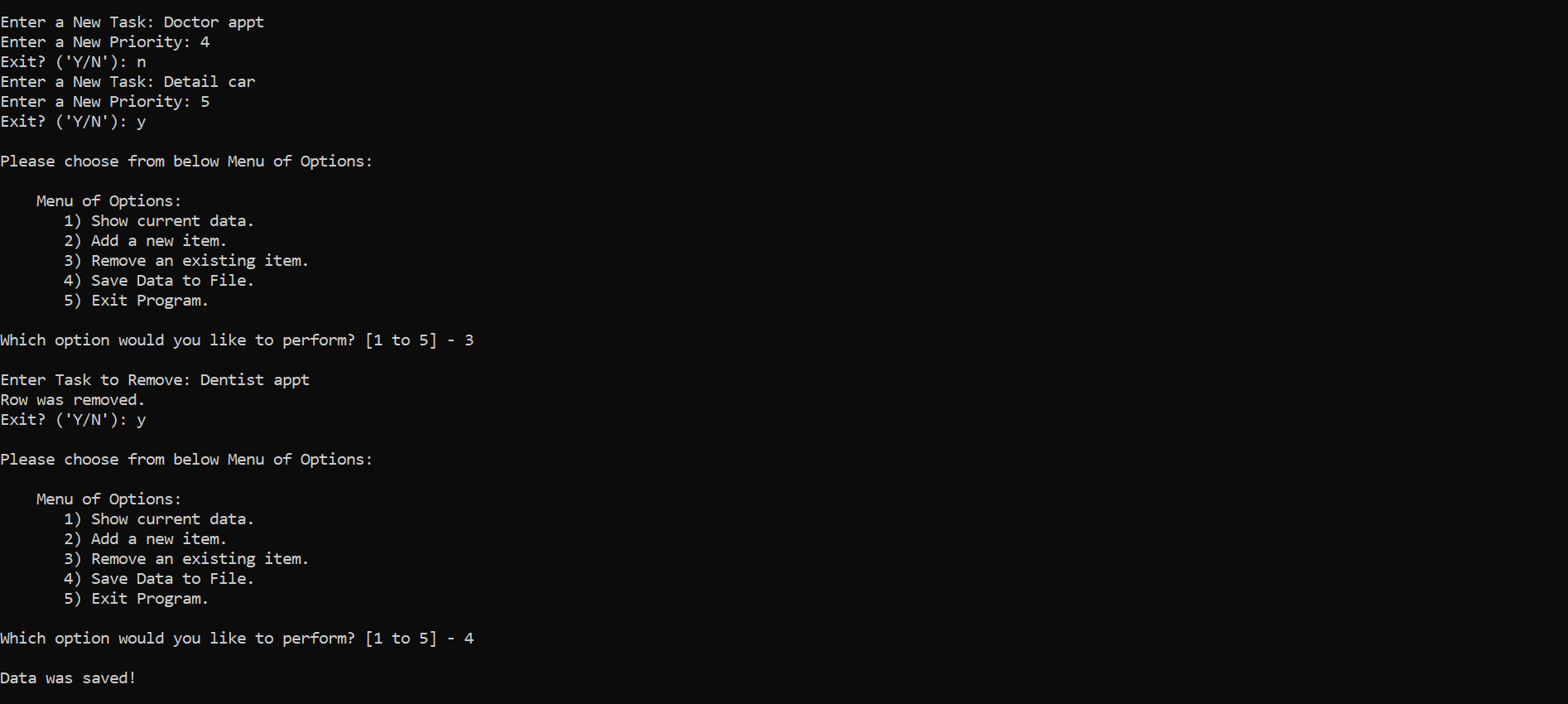


***Figure 14. Screenshot of the script running in PyCharm***

# Running the ToDo List Program Using the Command Window

And I also successfully ran the program using the Command Window (Figure 15).



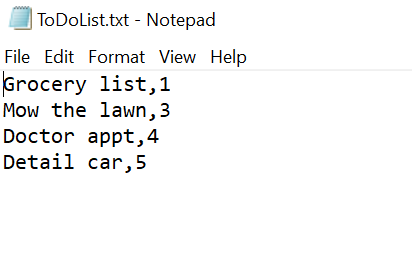




***Figure 15. Screenshot of the script running in a Command Window***

# Verify that it Worked

After I ran the program, I located the text file and opened it in a text editor to verify the file had data and it was correct (Figure 16).



***Figure 16. Verifying that the file has the correct data***

Summary

In the Module 05, I have created the ToDo List program and have demonstrated the use of different concepts I learned during the course such as script template, dictionary, key, read data from file into list, read data from a file into a dictionary, ‘Separations of Concerns’, GitHub, to name a few. By watching the instructional videos, reading the textbook and consulting additional documentation, I was able to successfully create the ToDo List program and ran the program using PyCharm and Command Window.