Tomas Armenta

April 16, 2023

IT FDN 110 A

Assignment 05

To Do List Script

Introduction

This paper will illustrate the steps taken to complete assignment 05. Beginning with module 05, I will describe the use of lists, dictionaries, and how they are utilized. I will also provide a brief discussion of labs performed in module 05 and then explain how what I learned was used to create the "To Do List" script for completion of assignment 05. Finally, I will conclude assignment 05 with a summary of lessons learned and final thoughts.

Lists

Lists are a great way of storing collections of data and learning how to utilize lists by storing information into a text file to be loaded again by a script, is one of the tools learned to complete assignment05. Essentially, lists can hold various types of data, but when saving and loading information to a text file, the data can be readily accessed by various programming languages by saving the list as a comma separated string, saved to a text file. As explained in the next sections lists will prove useful in completing the "To Do List." Figure 1 shows the scripts used to form the basis for saving and loading to a text file.

```
# Process the data
objFile = open(strFile, "w")
lstRow = ["1", "Bob Smith", "BSmith@Hotmail.com"]
objFile.write(lstRow[0] + ',' + lstRow[1] + ',' + lstRow[2] + '\n')
objFile.close()
```

```
# Process the data
objFile = open(strFile, "r")
for row in objFile:
    lstRow = row.split(",")
    print(lstRow)
    print(lstRow[0] + '|' + lstRow[1] + '|' + lstRow[2].strip())
objFile.close()
```

Fig.1, *Top*: showing code for saving data to a file, taken from module 5. Fig.1, *Bottom:* showing code for loading data from a file, taken from module 5.

Dictionaries

Similar to other sequences, learned in previous modules, dictionaries can store various forms of data while using key(character) subscripts. Dictionaries are very useful at saving data in pairs as we'll see incorporated into the assignment 05 script. Likewise, combining lists and dictionaries is a great way to create a list/table of dictionary objects. As shown in Figure 2, a "lstTable" is used to create a table of dictionaries.

```
# Create the table
dicRow = {"id":"1","name":"Bob Smith", "email":"BSmith@Hotmail.com"}
lstTable.append(dicRow)

# Process the data
for objRow in lstTable:
    print(objRow)
```

Fig.2, *Taken from module 05*: showing code for creating a dictionary and adding it to a table, by saving the information as a list of dictionaries, then processing the data and reading code line by line.

Labs

Lab 5-1, was an example of writing and reading a home inventory list, by taking string input values, creating a list, and saving the information to a text file. Lab 5-2, was an extension on Lab 5-1, collectively using lists and dictionaries to work with a table of dictionaries. Similarly, the labs were meant to support and create the "To Do List" script for assignment 05. See Figure 3 & 4 for more lab details.

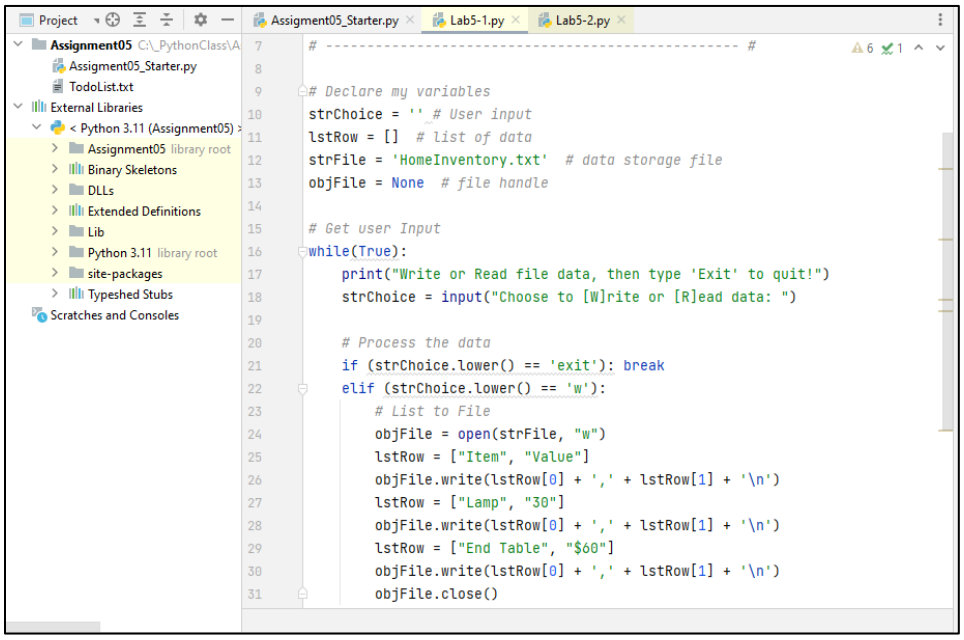

Fig.3.1, *Left*: showing Lab 5-1 PyCharm script for writing to a text file.

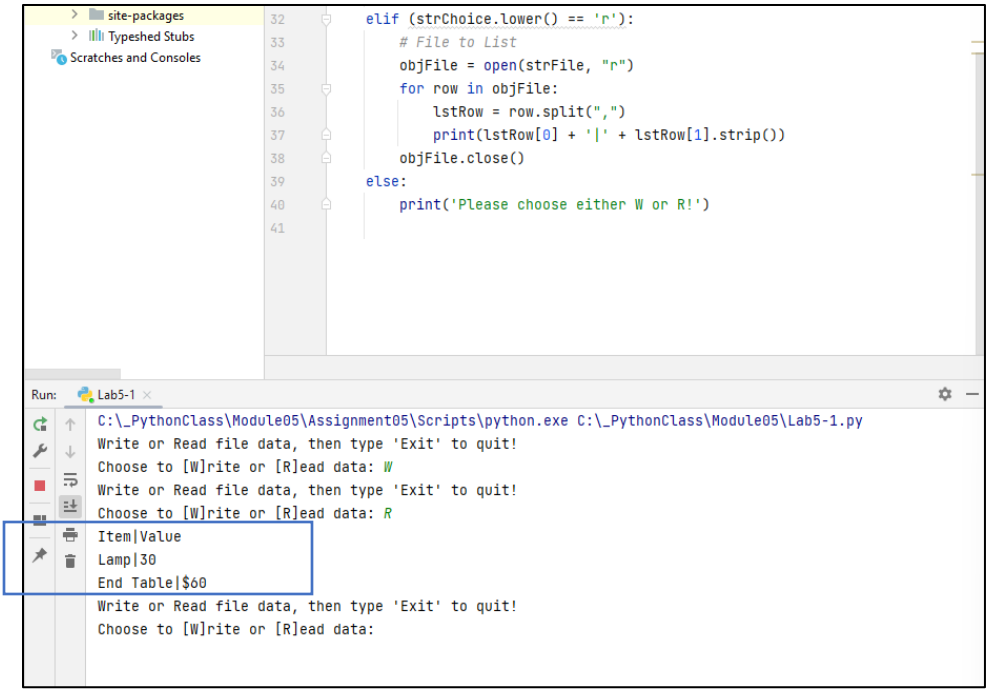

Fig.3.2, *Left*: showing Lab 5-1 PyCharm script for writing/reading from a text file, resulting output shown bottom-left blue square.

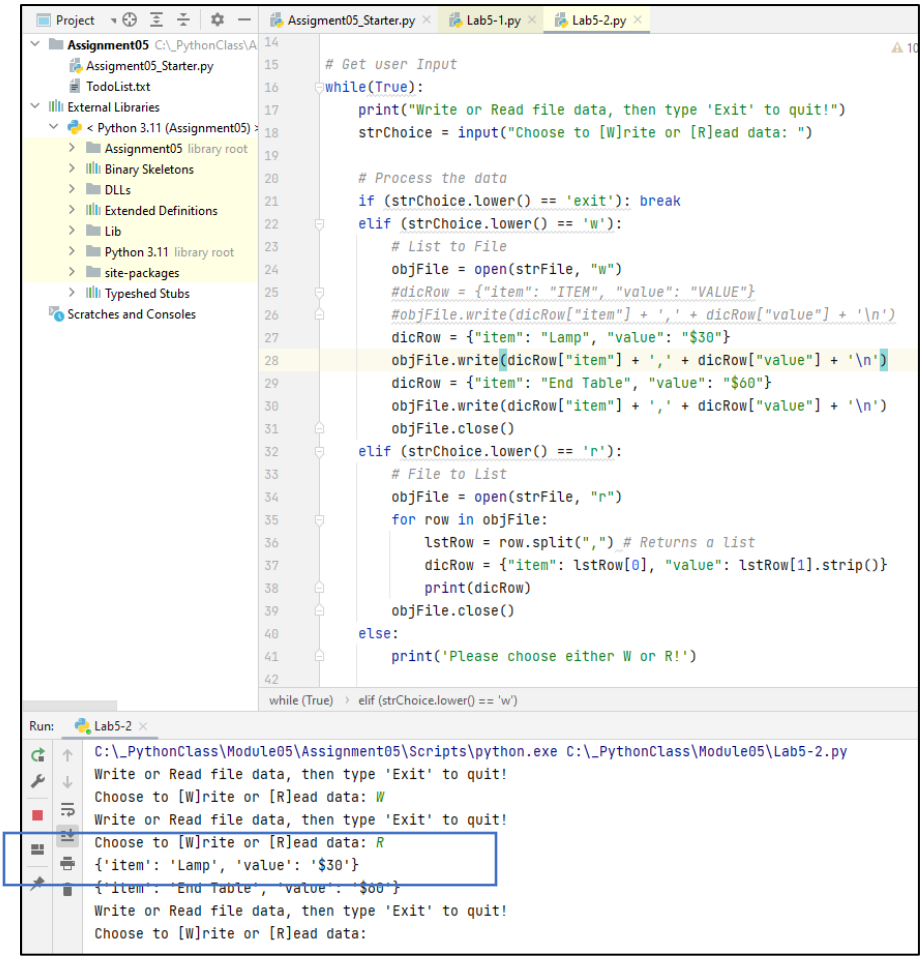

Fig.4, *Left*: showing Lab 5-2 PyCharm script for writing/reading from a text file similar to Lab 5-1, but saving as a list/table of dictionaries, resulting output shown bottom-left blue square.

To Do List Script

Applying the knowledge gained from module 5 and labs, I was able to successfully create a script that can take user input to create a “To Do List”. The information is taken from an already created file and/or created by the script. The data is then put in a list to create a table of dictionaries that form “Task” and “Priority” keys with values input by user. The data is then saved to a text file, as a comma separated string. Items can be added/removed at will. Altogether, the module 05 course video, reading the book chapter, reviewing the web pages, and viewing the support videos, were all used to create the script (Fig.5) and successfully run the script file in PyCharm and the Windows command window as shown in figure 6 & 7.

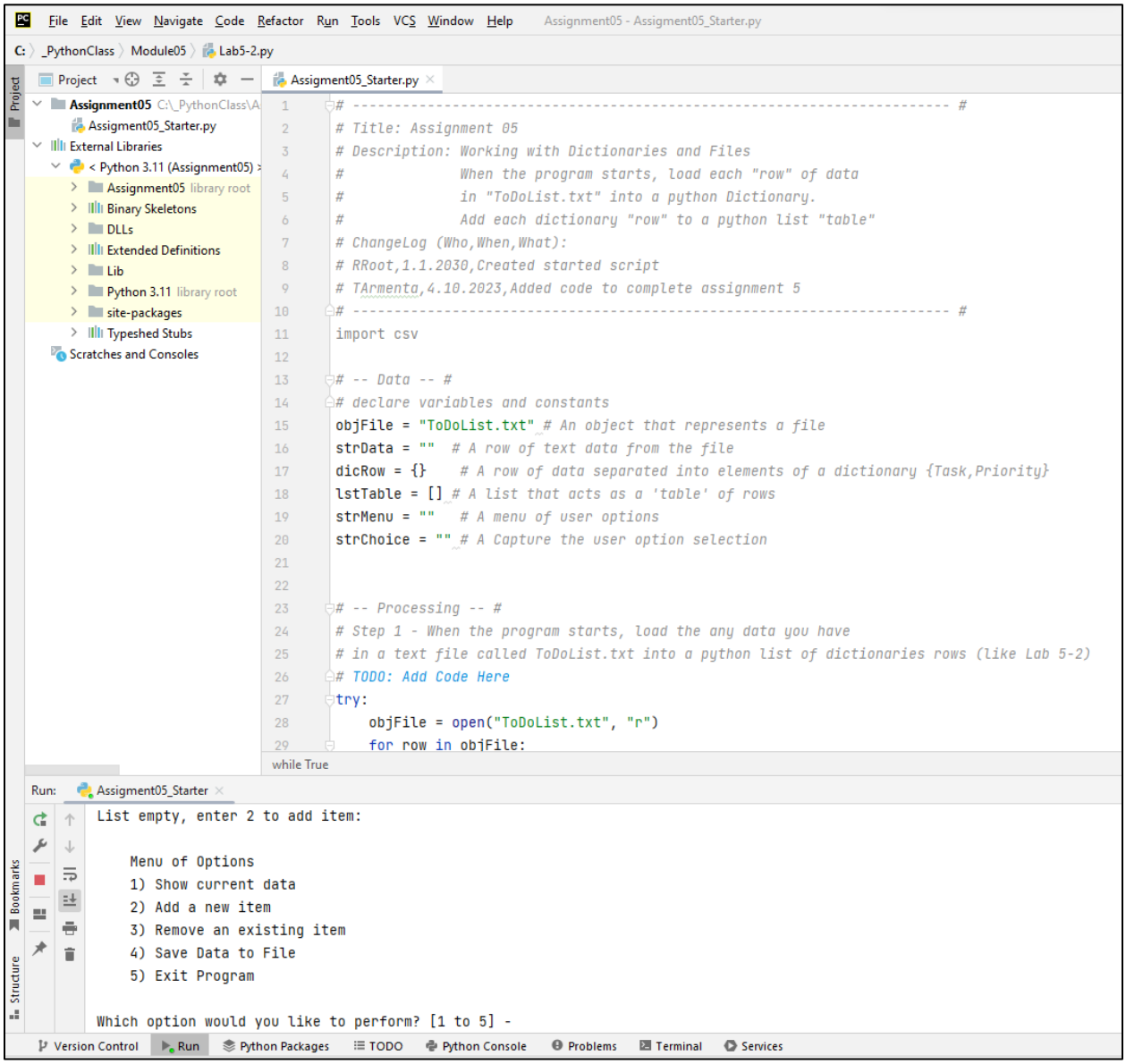

Fig.5.1, Showing assignment05 PyCharm script and running script with no text file created yet.

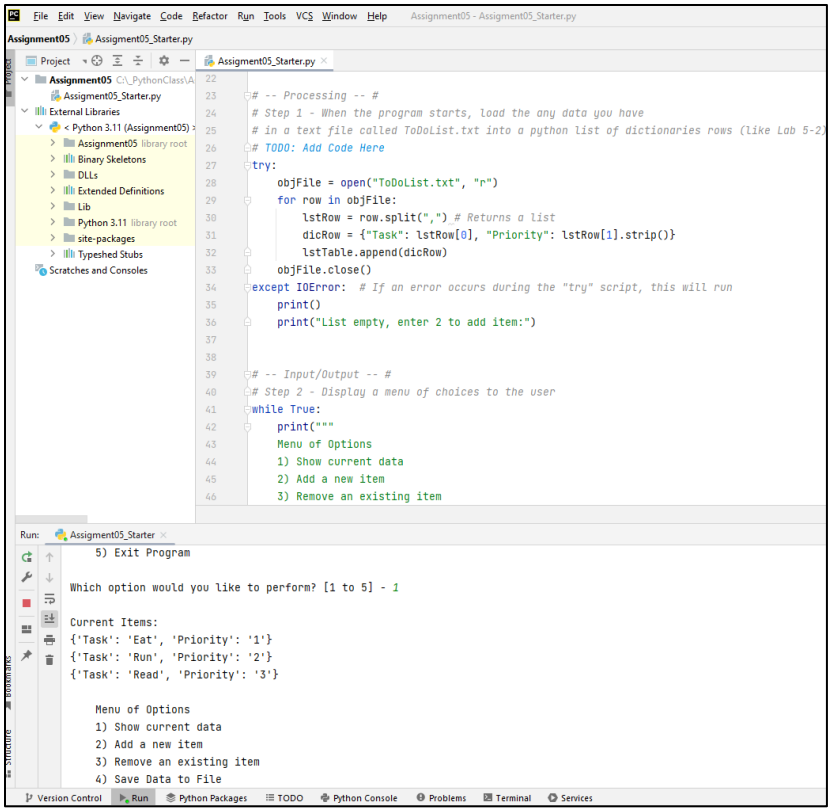

Fig.5.2, *Left*: showing assignment05 PyCharm script with opening processing code, menu, and showing list of dictionaries taken from user input.

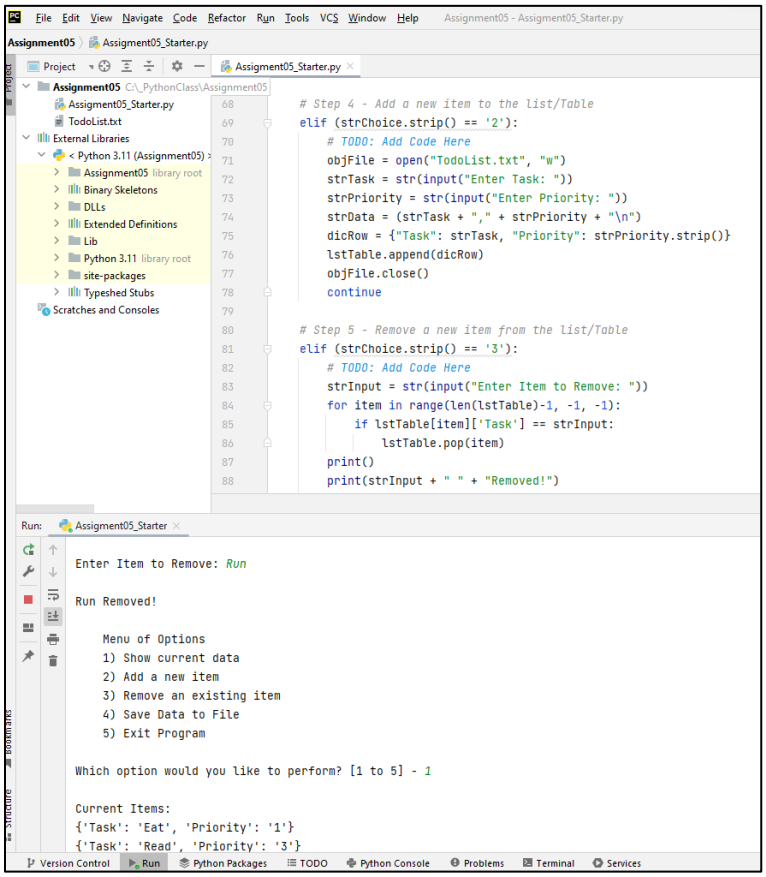

Fig.5.3, *Left*: showing assignment05 PyCharm script with add new items script, remove script, and resulting list of dictionaries left.

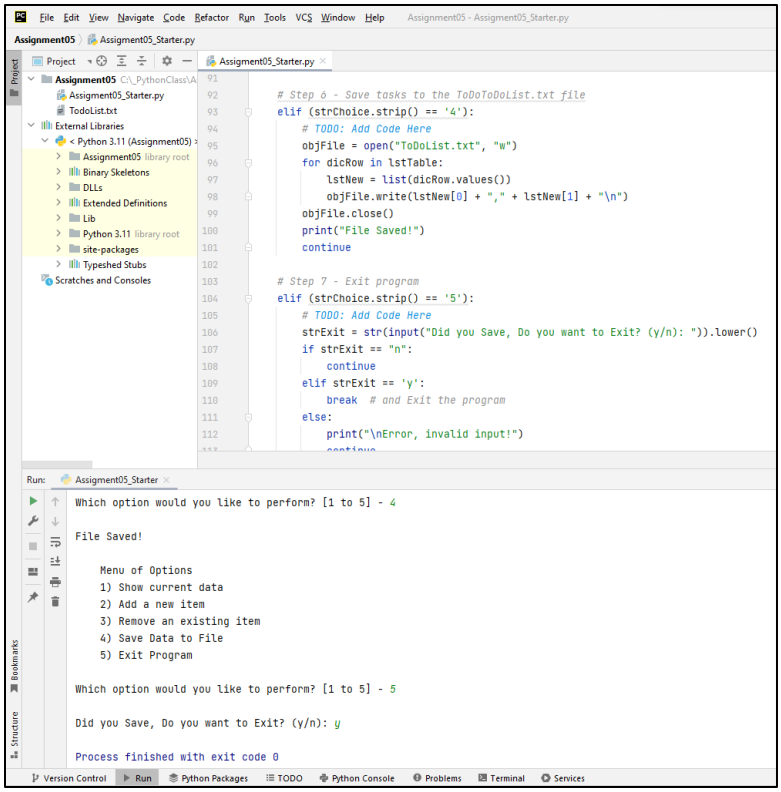

Fig.5.4, *Left*: showing assignment05 PyCharm script with save to text file script and exit program.

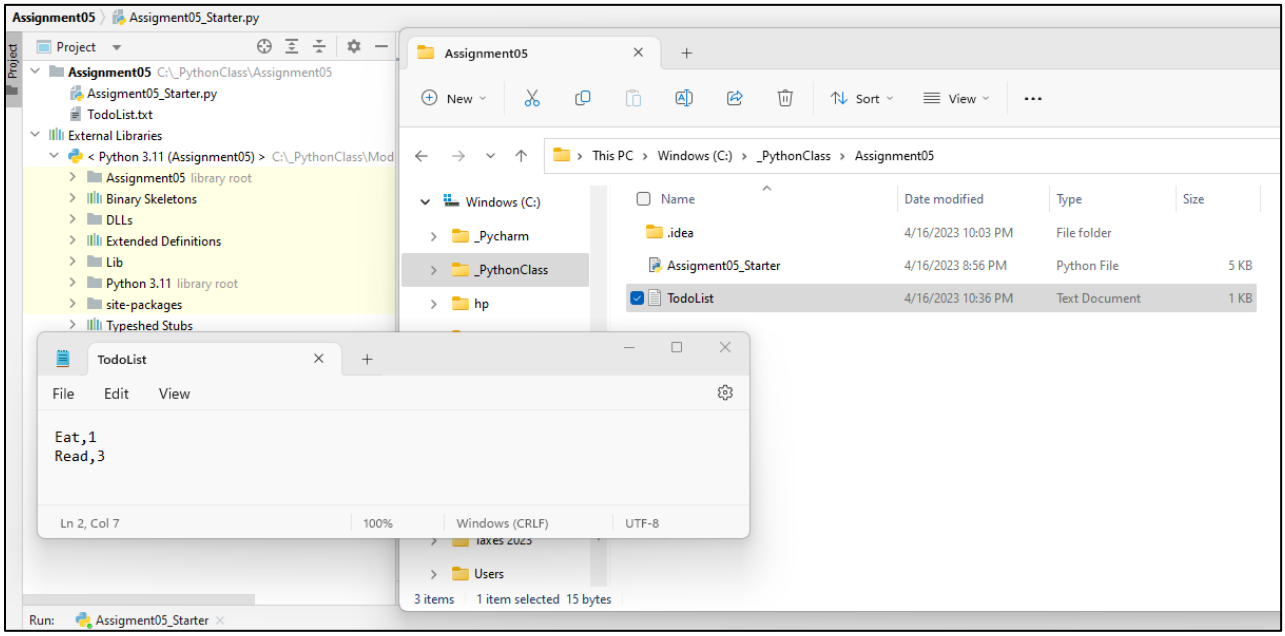

Fig.5.5, Showing assignment05 PyCharm/computer location of "ToDoList.txt" file, with resulting string list created.

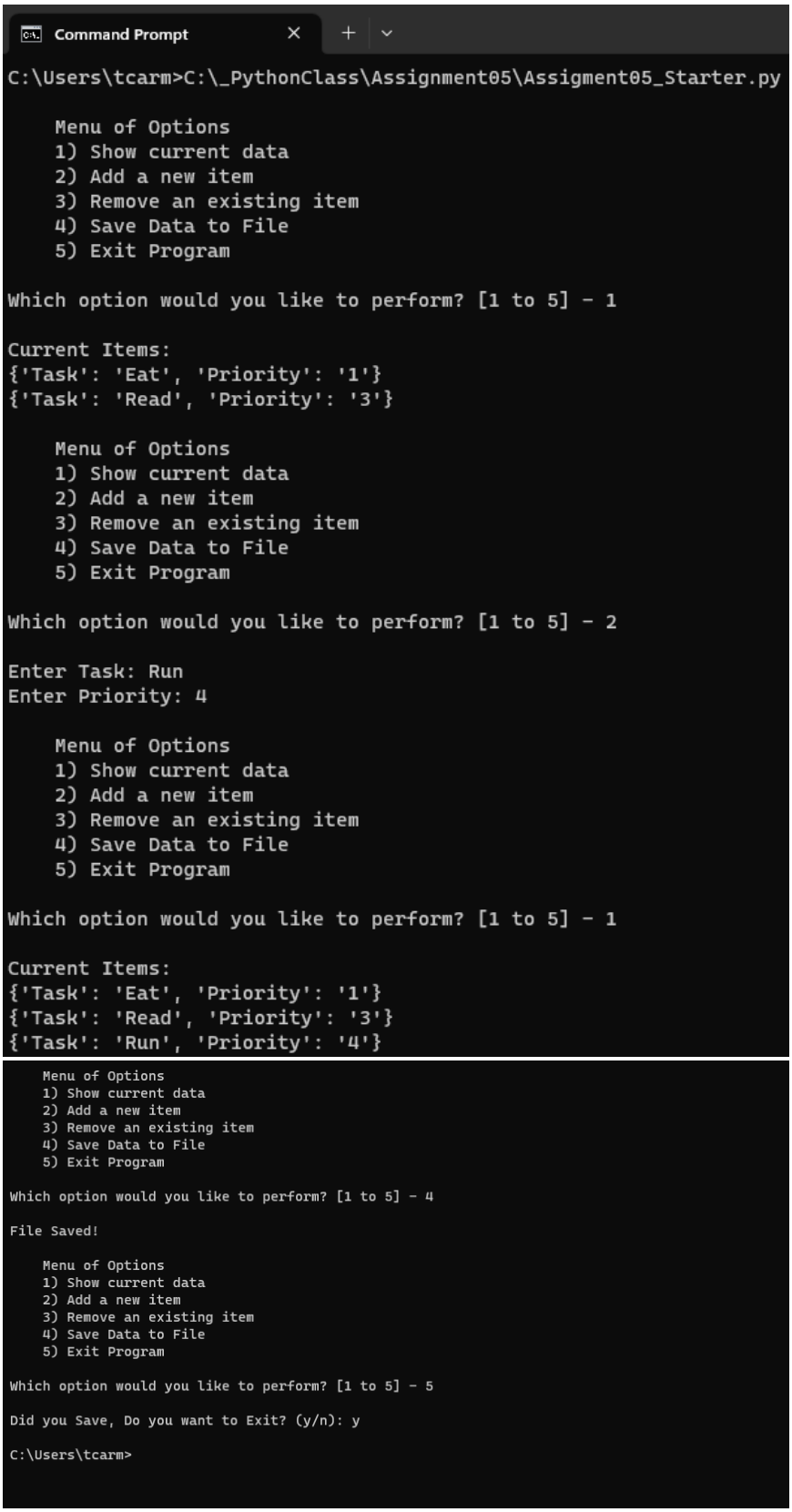

Fig.6, *Left*: showing assignment05 Windows Command prompt script running, using list created from PyCharm, adding to the list, saving, and showing results.

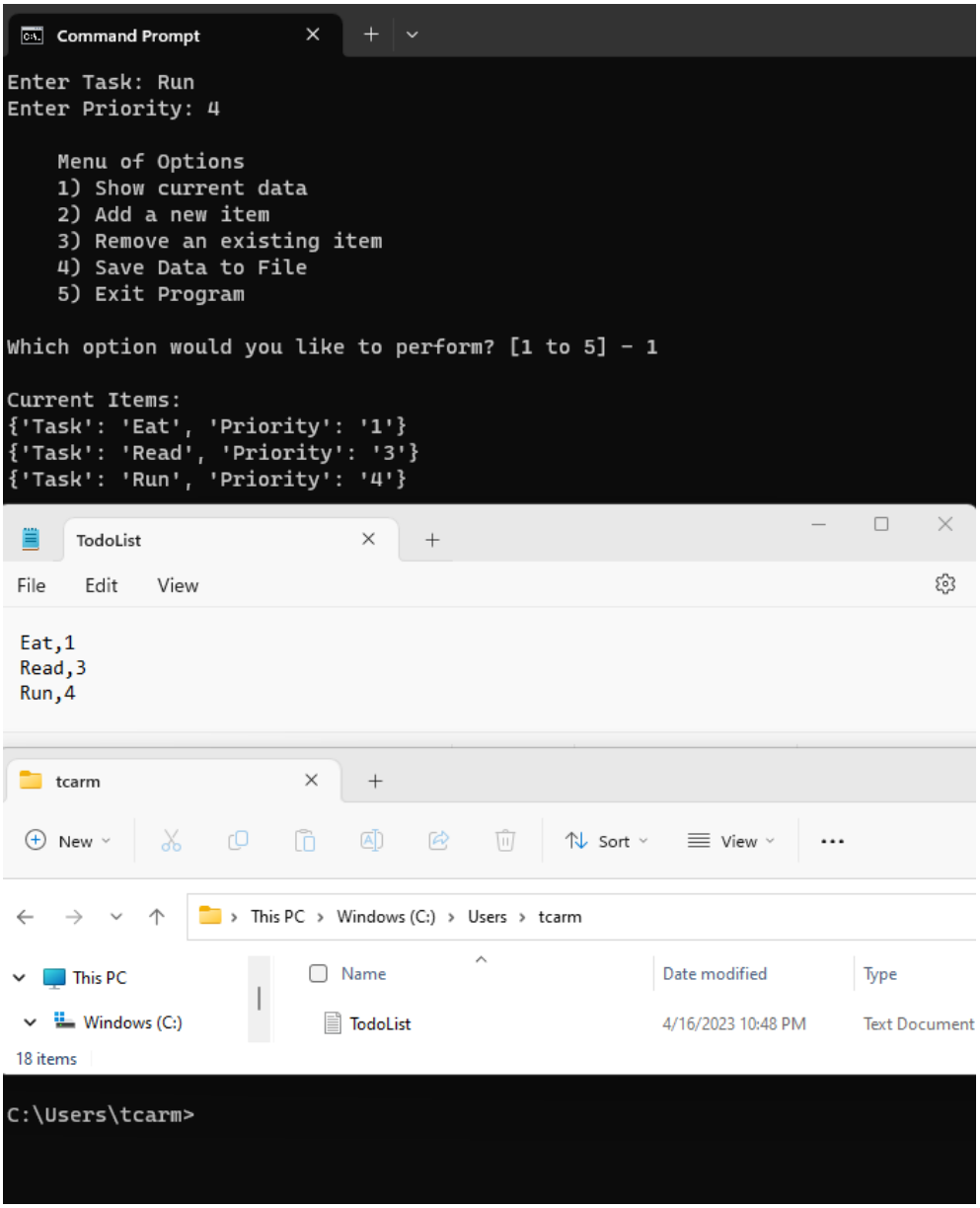

Fig.7, Showing assignment05 Command prompt script and results, after adding additional data, showing "ToDoList.txt" and saved file location.

Summary

Assignment 05 was challenging, but I was able to gather from module 05 and external website research to complete a script suitable for assignment 05. The labs were useful in showing the basis of writing/reading lists and dictionaries. It took some time, but I finally created a script that can read from a text file, edit, and save back to file. This assignment taught me how to create a list/table of dictionaries with the ability to add/remove rows of information. I added a few additions to help bolster my code and like always there can be more improvements, but I ended up happy with the results.