Janelle Bernales

May 17, 2023

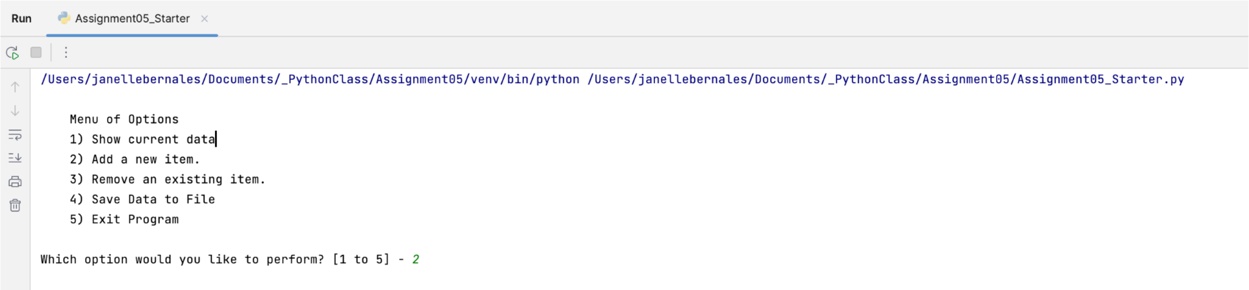
Intro to Programming (Python)

Assignment 05

# Intro to Programming: Adding and deleting user input, saving to a file, and using GitHub

## Introduction

In Module 05 I learned how to edit another programmer's code to create a simple text-based program that records, displays, and edits a user's input prioritized "to do" tasks. The users' input is saved and printed as a formatted list in a text file, "ToDoList.txt:"



```
/Users/janellebernales/Documents/_PythonClass/Assignment05/venv/bin/python /Users/janellebernales/Documents/_PythonClass/Assignment05/Assignment05_Starter.py

    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
```

*Fig. 1: The user menu displayed in "Assignment05_Starter.py"*

Like the latest version of the "HomeInventory.py" script, "Assignment05_Starter.py" can store multiple lines of user input, display the user's input, and write this data to a file in a table format. However, "Assignment05_Starter.py" allows users to delete previous input, too.

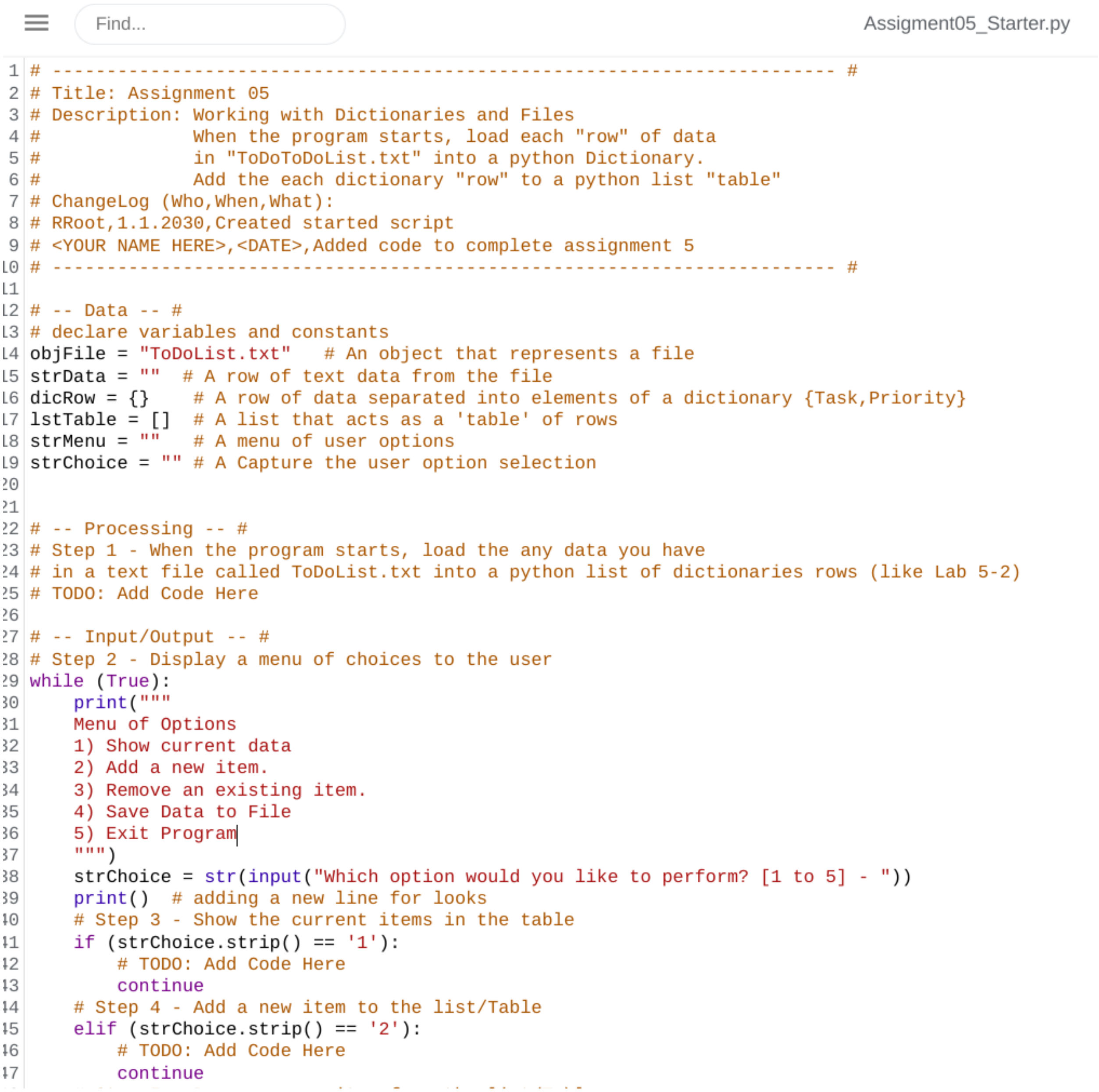
In this module I learned how to use the "append" and "remove" functions to edit a list of user input. I also learned how to upload code to Github, a popular version control system website for sharing code online.[1]

[1] What is GitHub, and What Is It Used For? Brown, Korbin, 2019. Accessed May 17, 2023. http://www.howtogeek.com/180167/htg-explains-what-is-github-and-what-do-geeks-use-it-for/

## Creating the Program

To make this program I uploaded the provided script for Module 05 titled “Assignment05_Starter.py” within the “Assignment05” subfolder in my “_PythonClass” folder. I also created a new project in PyCharm titled “Assignment05.”

Once I added the script to my PyCharm project, I began editing the code included in the project:



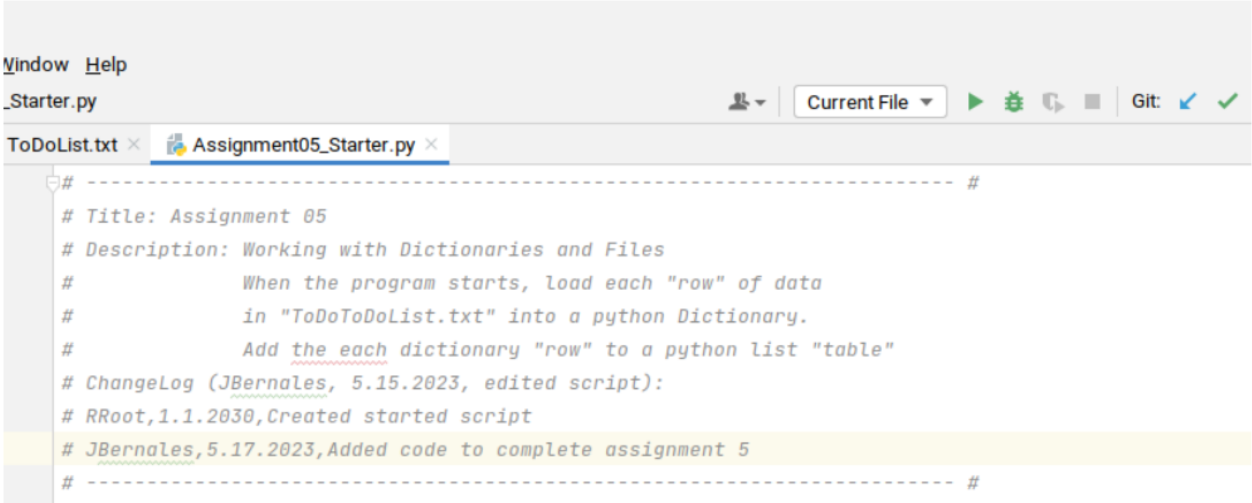
```
# ------------------------------------------------------------------------ #
# Title: Assignment 05
# Description: Working with Dictionaries and Files
#              When the program starts, load each "row" of data
#              in "ToDoToDoList.txt" into a python Dictionary.
#              Add the each dictionary "row" to a python list "table"
# ChangeLog (Who,When,What):
# RRoot,1.1.2030,Created started script
# <YOUR NAME HERE>,<DATE>,Added code to complete assignment 5
# ------------------------------------------------------------------------ #

# -- Data -- #
# declare variables and constants
objFile = "ToDoList.txt"   # An object that represents a file
strData = ""  # A row of text data from the file
dicRow = {}    # A row of data separated into elements of a dictionary {Task,Priority}
lstTable = []  # A list that acts as a 'table' of rows
strMenu = ""   # A menu of user options
strChoice = "" # A Capture the user option selection


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

# -- Input/Output -- #
# Step 2 - Display a menu of choices to the user
while (True):
    print("""
    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
    """)
    strChoice = str(input("Which option would you like to perform? [1 to 5] - "))
    print()  # adding a new line for looks
    # Step 3 - Show the current items in the table
    if (strChoice.strip() == '1'):
        # TODO: Add Code Here
        continue
    # Step 4 - Add a new item to the list/Table
    elif (strChoice.strip() == '2'):
        # TODO: Add Code Here
        continue
```

*Fig. 2: The original provided version of the “Assignment05_Starter.py” script*

Editing someone else’s script was much easier than creating brand new code in some ways and a bit more challenging in others. By having a pre-written script I already had a roadmap to follow to accomplish the program’s goals which made getting started a lot easier.

However, it also challenged me to think through each step of the coding process more closely and adapt my usual manner of writing. Some of the variable names seemed confusing to me (like “lstTable” which I kept confusing as “1stTable,” causing errors several times. However, I kept these variable names to remain consistent with the original version of the code.

I updated the comments section in the header accordingly:

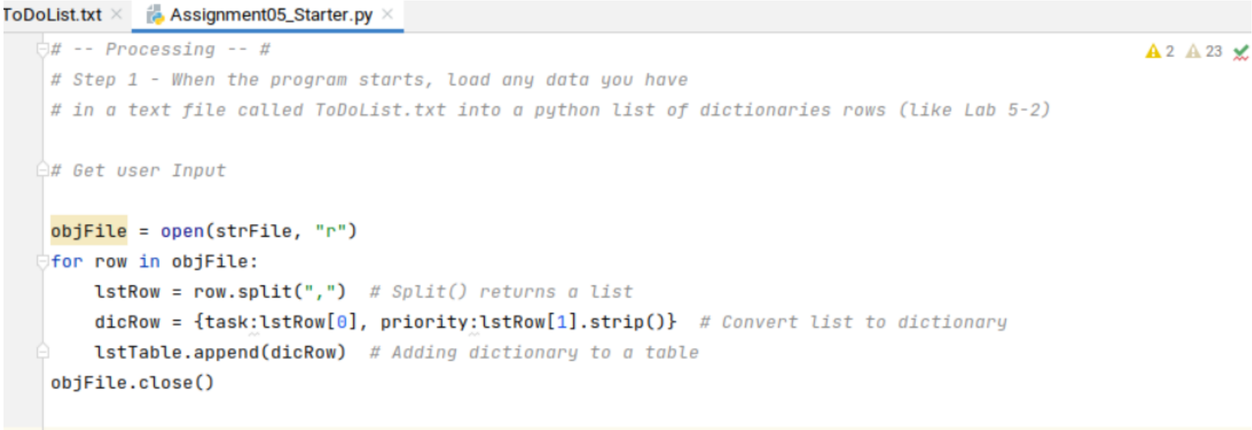


```
# ---------------------------------------------------------------------- #
# Title: Assignment 05
# Description: Working with Dictionaries and Files
#              When the program starts, load each "row" of data
#              in "ToDoToDoList.txt" into a python Dictionary.
#              Add the each dictionary "row" to a python list "table"
# ChangeLog (JBernales, 5.15.2023, edited script):
# RRoot,1.1.2030,Created started script
# JBernales,5.17.2023,Added code to complete assignment 5
# ---------------------------------------------------------------------- #
```

*Fig. 3: Updated script header for “Assignment05_Starter.py”*

## Create a Text File and a List from User Input

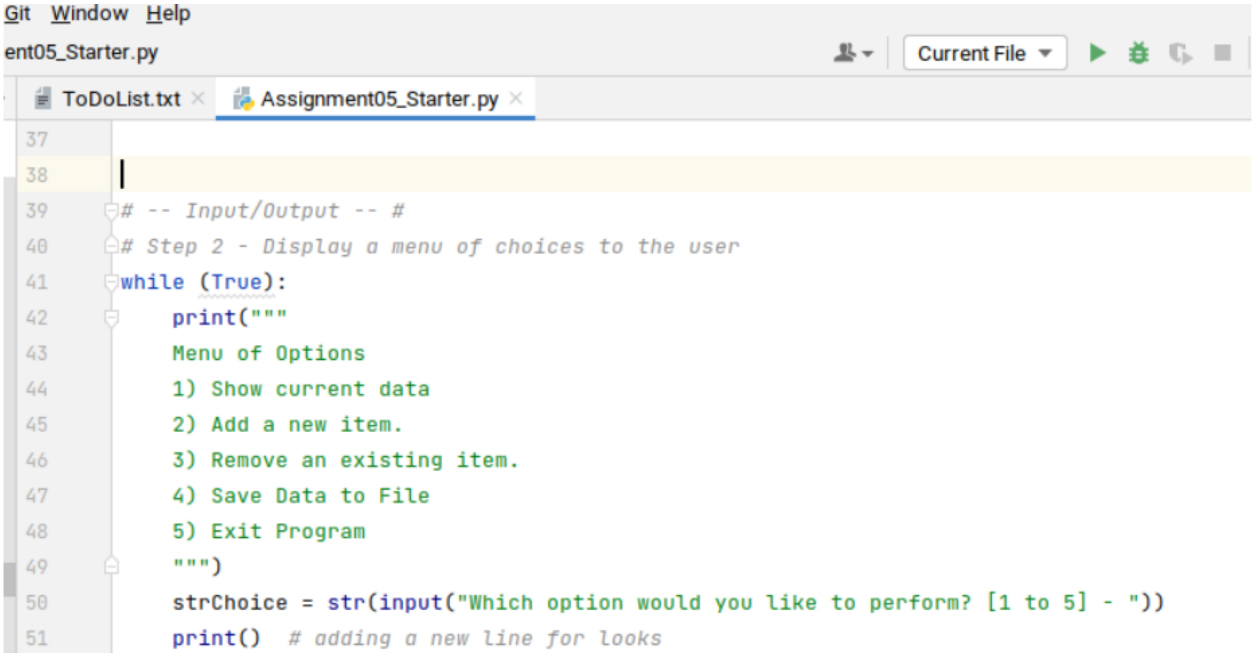
Next, I created a text file titled “ToDoList” in the same folder as Assignment05 as instructed. I used a similar code format as in Lab 5-2 to create initial commands to open and write to this file. The “lstRow” command returns the user’s input tasks and priorities as a list divided into rows (“dicRow”) that is converted to a dictionary before being added to the table in the text file:



```
# -- Processing -- #
# 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

objFile = open(strFile, "r")
for row in objFile:
    lstRow = row.split(",")  # Split() returns a list
    dicRow = {task:lstRow[0], priority:lstRow[1].strip()}  # Convert list to dictionary
    lstTable.append(dicRow)  # Adding dictionary to a table
objFile.close()
```

*Fig. 4: Initial script adapted from Lab 5-2 to open, write, and store user input in “ToDoList.txt” file*

Next, I filled out the unwritten sections within the user menu code contained within “Step 2.” Having an “if / elif” loop already written made creating this program much easier than previous scripts I’ve written from scratch. Because the function of this loop is very similar to the one we used for the user menu in “HomeInventory.py,” I was able to recycle and adapt several lines of code from that script to build out functions in this script’s user menu:

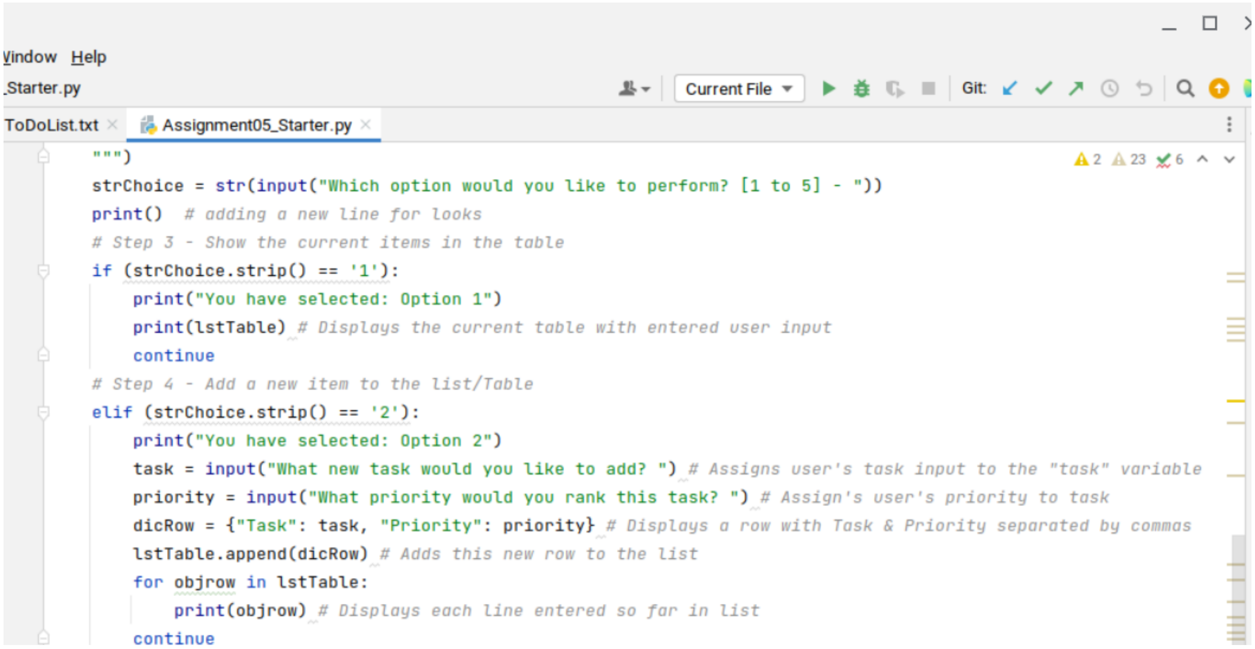


```


# -- Input/Output -- #
# Step 2 - Display a menu of choices to the user
while (True):
    print("""
    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
    """)
    strChoice = str(input("Which option would you like to perform? [1 to 5] - "))
    print()  # adding a new line for looks
```

*Fig. 5: The provided user menu in the “Assignment05_Starter.py” script which is continuously displayed with a “while(True)” condition*

I chose to keep several of the same user interaction features from “HomeInventory.py,” including printing “You have selected: Option #X” confirming the user’s menu selection, using the input command to record the user’s task and assigned priority number to the list to be included in the “ToDoList.txt” file, and “Press enter to return to the main menu” input command after completing each option.

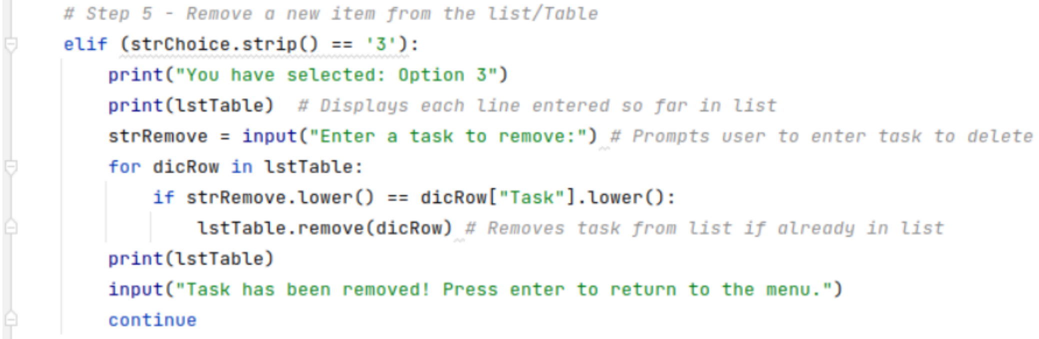
Although these steps are not necessary, I chose to include these prompts because I like to see confirmation of my selected choice, its outcome, and redirection back to the main menu for clarification:



```
"""
strChoice = str(input("Which option would you like to perform? [1 to 5] - "))
print()  # adding a new line for looks
# Step 3 - Show the current items in the table
if (strChoice.strip() == '1'):
    print("You have selected: Option 1")
    print(lstTable) # Displays the current table with entered user input
    continue
# Step 4 - Add a new item to the list/Table
elif (strChoice.strip() == '2'):
    print("You have selected: Option 2")
    task = input("What new task would you like to add? ") # Assigns user's task input to the "task" variable
    priority = input("What priority would you rank this task? ") # Assign's user's priority to task
    dicRow = {"Task": task, "Priority": priority} # Displays a row with Task & Priority separated by commas
    lstTable.append(dicRow) # Adds this new row to the list
    for objrow in lstTable:
        print(objrow) # Displays each line entered so far in list
    continue
```

*Fig. 6: Steps 3 and 4 in “Assignment05_Starter.py” display all user input currently entered in the program and record user input for a task and its assigned priority, respectively.*

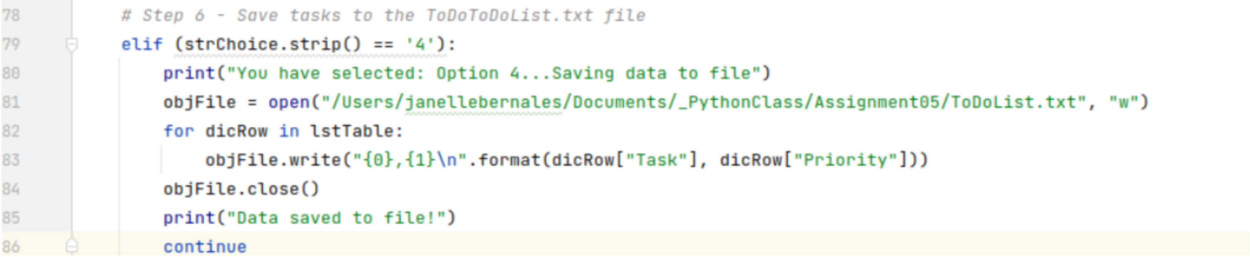
Unlike in “HomeAssignment.py,” “Assignment05_Starter.py” enables a user to delete a previously input task and priority. This is done by using the “remove” command in Step 5 of the code. Using “strRemove” with “dicRow” and “lstTable” will delete any row found in the table that matches the user’s input in this section:



```
# Step 5 - Remove a new item from the list/Table
elif (strChoice.strip() == '3'):
    print("You have selected: Option 3")
    print(lstTable)  # Displays each line entered so far in list
    strRemove = input("Enter a task to remove:") # Prompts user to enter task to delete
    for dicRow in lstTable:
        if strRemove.lower() == dicRow["Task"].lower():
            lstTable.remove(dicRow) # Removes task from list if already in list
    print(lstTable)
    input("Task has been removed! Press enter to return to the menu.")
    continue
```

*Fig. 7: This section of the code enables a user to remove a previously entered prioritized task in the table.*

In Step 6, I included the same “objFile = open()”, “objFile.write…”, and “objFile.close() commands from my “HomeInventory.py” script to save the user’s previously input tasks and priorities to the “ToDoList” text file.



```
    # Step 6 - Save tasks to the ToDoToDoList.txt file
    elif (strChoice.strip() == '4'):
        print("You have selected: Option 4...Saving data to file")
        objFile = open("/Users/janellebernales/Documents/_PythonClass/Assignment05/ToDoList.txt", "w")
        for dicRow in lstTable:
            objFile.write("{0},{1}\n".format(dicRow["Task"], dicRow["Priority"]))
        objFile.close()
        print("Data saved to file!")
        continue
```

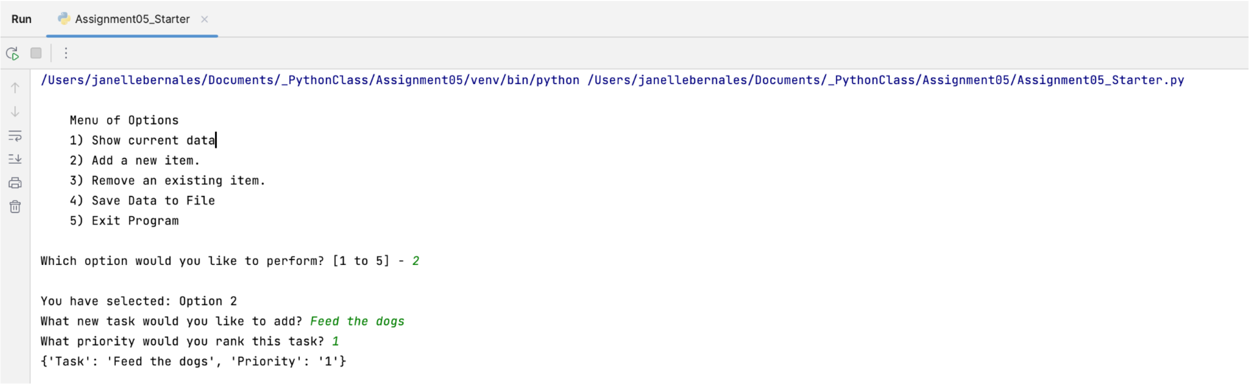
*Fig. 8: Using "objFile" along with the open command, the file path name, and "w" allows the program to write the user's previously entered task and priority inputs into the "ToDoList" text file.*

Note, PyCharm provided the curly-brackets formatting used within "objFile.write()" after I ran into a type error when I previously excluded them in the same line. If I had written this script within a more bare bones script editor like IDLE, I likely would have been stumped here because it wouldn't have flagged the error and provided a potential fix.

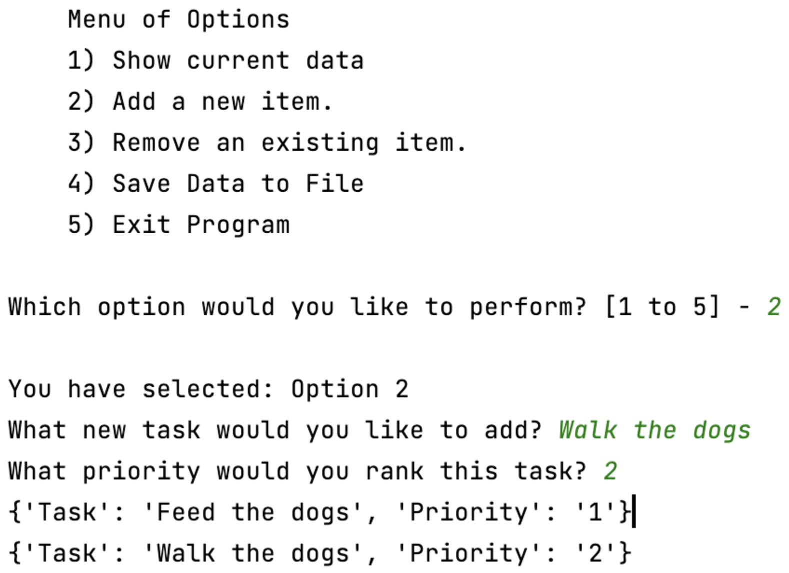
With the loop code completed, it's time to run the program.

## Run the Program

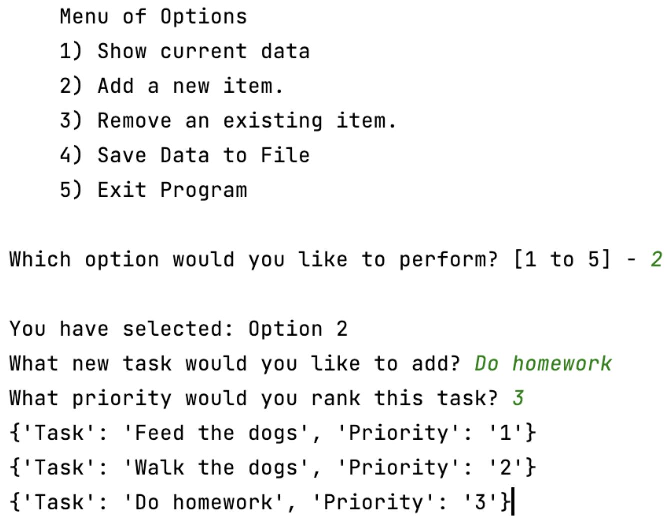
After testing the code several times in PyCharm, consulting the Canvas course message board, and reviewing classmates' code shared via GitHub, I was able to adjust my code and successfully run it in PyCharm:



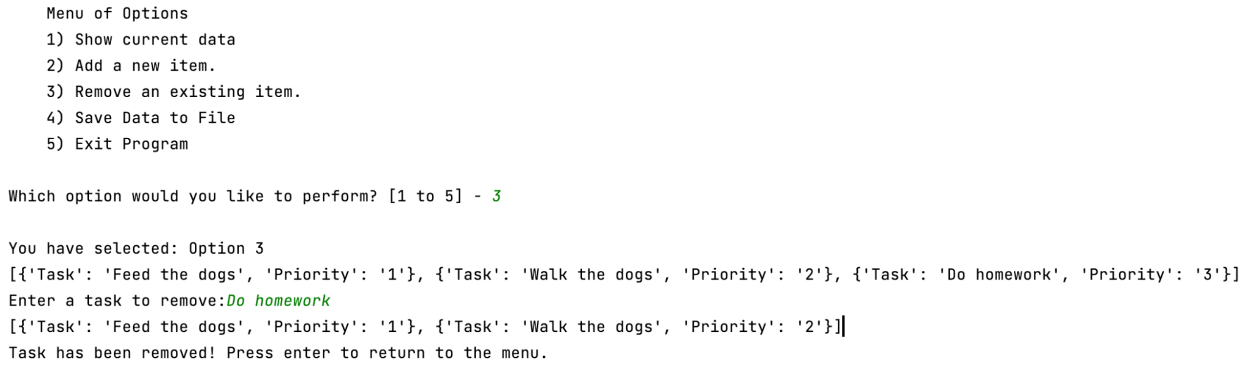
```
/Users/janellebernales/Documents/_PythonClass/Assignment05/venv/bin/python /Users/janellebernales/Documents/_PythonClass/Assignment05/Assignment05_Starter.py

    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

You have selected: Option 2
What new task would you like to add? Feed the dogs
What priority would you rank this task? 1
{'Task': 'Feed the dogs', 'Priority': '1'}
```



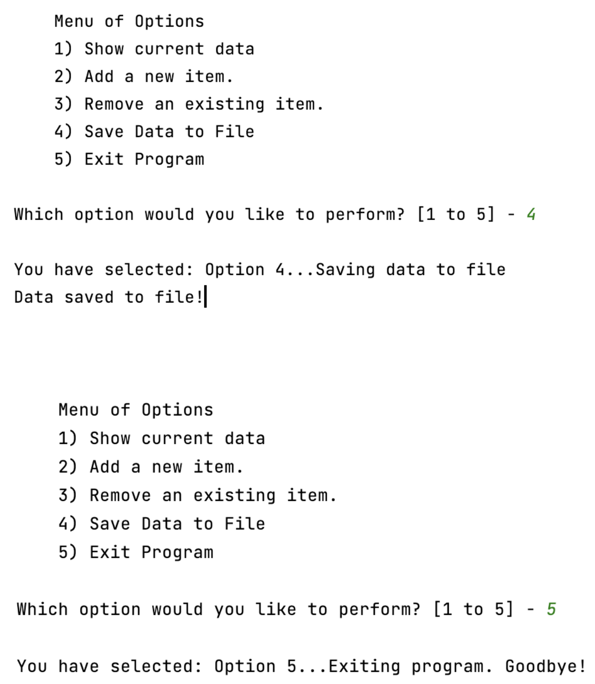
```
    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

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

You have selected: Option 2
What new task would you like to add? Walk the dogs
What priority would you rank this task? 2
{'Task': 'Feed the dogs', 'Priority': '1'}
{'Task': 'Walk the dogs', 'Priority': '2'}
```



```
    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

You have selected: Option 2
What new task would you like to add? Do homework
What priority would you rank this task? 3
{'Task': 'Feed the dogs', 'Priority': '1'}
{'Task': 'Walk the dogs', 'Priority': '2'}
{'Task': 'Do homework', 'Priority': '3'}
```



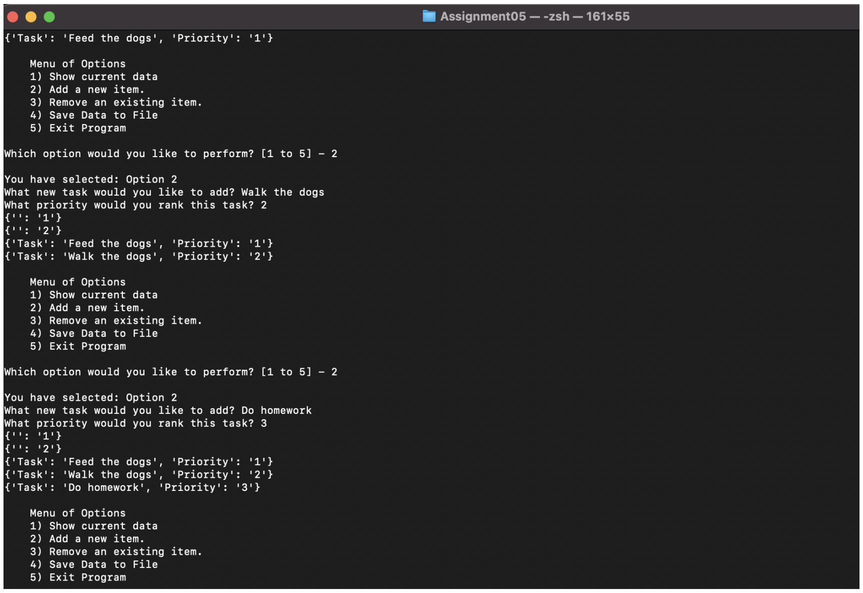
```
    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

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

You have selected: Option 3
[{'Task': 'Feed the dogs', 'Priority': '1'}, {'Task': 'Walk the dogs', 'Priority': '2'}, {'Task': 'Do homework', 'Priority': '3'}]
Enter a task to remove:Do homework
[{'Task': 'Feed the dogs', 'Priority': '1'}, {'Task': 'Walk the dogs', 'Priority': '2'}]
Task has been removed! Press enter to return to the menu.
```



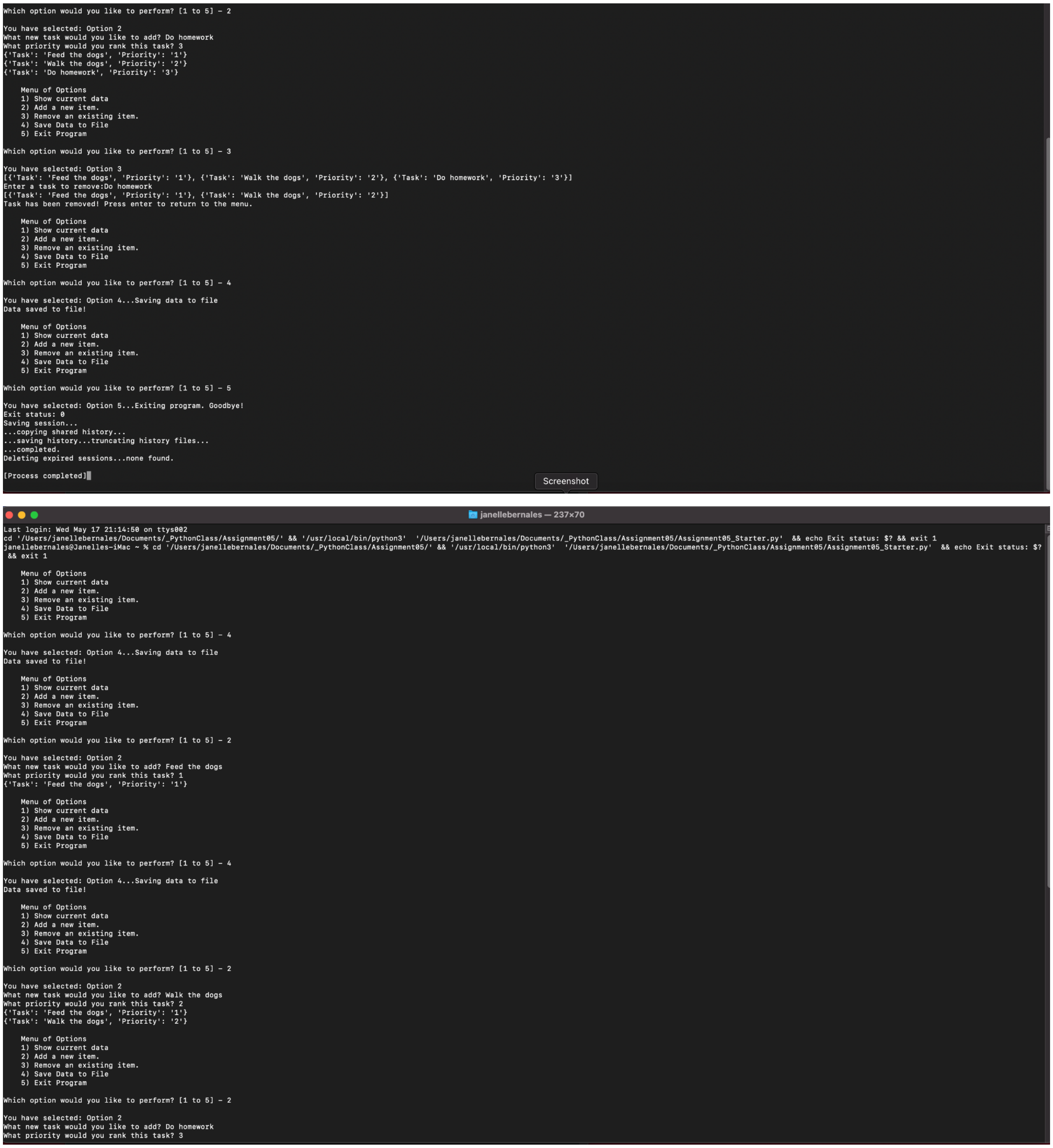
```
    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 4

You have selected: Option 4...Saving data to file
Data saved to file!



    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] - 5

You have selected: Option 5...Exiting program. Goodbye!
```

*Fig. 9: “Assignment05_Starter.py” has successfully run in PyCharm.*

This is what the “Assignment05_Starter.py” script looks like after running successfully in Terminal:



```
{'Task': 'Feed the dogs', 'Priority': '1'}

    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

You have selected: Option 2
What new task would you like to add? Walk the dogs
What priority would you rank this task? 2
{'': '1'}
{'': '2'}
{'Task': 'Feed the dogs', 'Priority': '1'}
{'Task': 'Walk the dogs', 'Priority': '2'}

    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

You have selected: Option 2
What new task would you like to add? Do homework
What priority would you rank this task? 3
{'': '1'}
{'': '2'}
{'Task': 'Feed the dogs', 'Priority': '1'}
{'Task': 'Walk the dogs', 'Priority': '2'}
{'Task': 'Do homework', 'Priority': '3'}

    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

You have selected: Option 2
What new task would you like to add? Do homework
What priority would you rank this task? 3
{'Task': 'Feed the dogs', 'Priority': '1'}
{'Task': 'Walk the dogs', 'Priority': '2'}
{'Task': 'Do homework', 'Priority': '3'}

    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

You have selected: Option 3
[{'Task': 'Feed the dogs', 'Priority': '1'}, {'Task': 'Walk the dogs', 'Priority': '2'}, {'Task': 'Do homework', 'Priority': '3'}]
Enter a task to remove:Do homework
[{'Task': 'Feed the dogs', 'Priority': '1'}, {'Task': 'Walk the dogs', 'Priority': '2'}]
Task has been removed! Press enter to return to the menu.

    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

You have selected: Option 4...Saving data to file
Data saved to file!

    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] - 5

You have selected: Option 5...Exiting program. Goodbye!
Exit status: 0
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.
Deleting expired sessions...none found.

[Process completed]
```

```
Last login: Wed May 17 21:14:50 on ttys002
cd '/Users/janellebernales/Documents/_PythonClass/Assignment05/' && '/usr/local/bin/python3'  '/Users/janellebernales/Documents/_PythonClass/Assignment05/Assignment05_Starter.py'  && echo Exit status: $? && exit 1
janellebernales@Janelles-iMac ~ % cd '/Users/janellebernales/Documents/_PythonClass/Assignment05/' && '/usr/local/bin/python3'  '/Users/janellebernales/Documents/_PythonClass/Assignment05/Assignment05_Starter.py'  && echo Exit status: $?
 && exit 1

    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

You have selected: Option 4...Saving data to file
Data saved to file!

    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

You have selected: Option 2
What new task would you like to add? Feed the dogs
What priority would you rank this task? 1
{'Task': 'Feed the dogs', 'Priority': '1'}

    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

You have selected: Option 4...Saving data to file
Data saved to file!

    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

You have selected: Option 2
What new task would you like to add? Walk the dogs
What priority would you rank this task? 2
{'Task': 'Feed the dogs', 'Priority': '1'}
{'Task': 'Walk the dogs', 'Priority': '2'}

    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

You have selected: Option 2
What new task would you like to add? Do homework
What priority would you rank this task? 3
```

*Fig. 10: “Assignment05_Starter.py” has successfully run in Terminal*

## Summary

In Module 05 I learned how editing another programmer’s code presents both advantages and some challenges when creating a program. In addition, I learned:

- **Make sure you're referencing the right variables.** Most of the errors I ran into when testing my code came from me referencing the wrong variable on accident (i.e., referring to "objFile" instead of "dicRow," or inaccurately referring to variables, like typing "1stTable" instead of "lstTable.") It seems like such a small error but it can completely derail your program.
- **GitHub is an easy-to-use tool to share, compare, and update your code with other programmers.** Being able to easily access other programmers' code files and compare them to mine was very helpful in this module. The ability to review others' code allowed me to spot my own mistakes more easily while also seeing examples of how a particular code block can be written in a more succinct, readable way.