**Course:** Foundations of Programming, Python

**Assignment:** 06

**Name:**  ToDoFile

**Description:** CreatingPython script that reads data from file and represents each row of data as a dictionary and adds rows to a list object to create a table of data. Script works with user input and existing file.

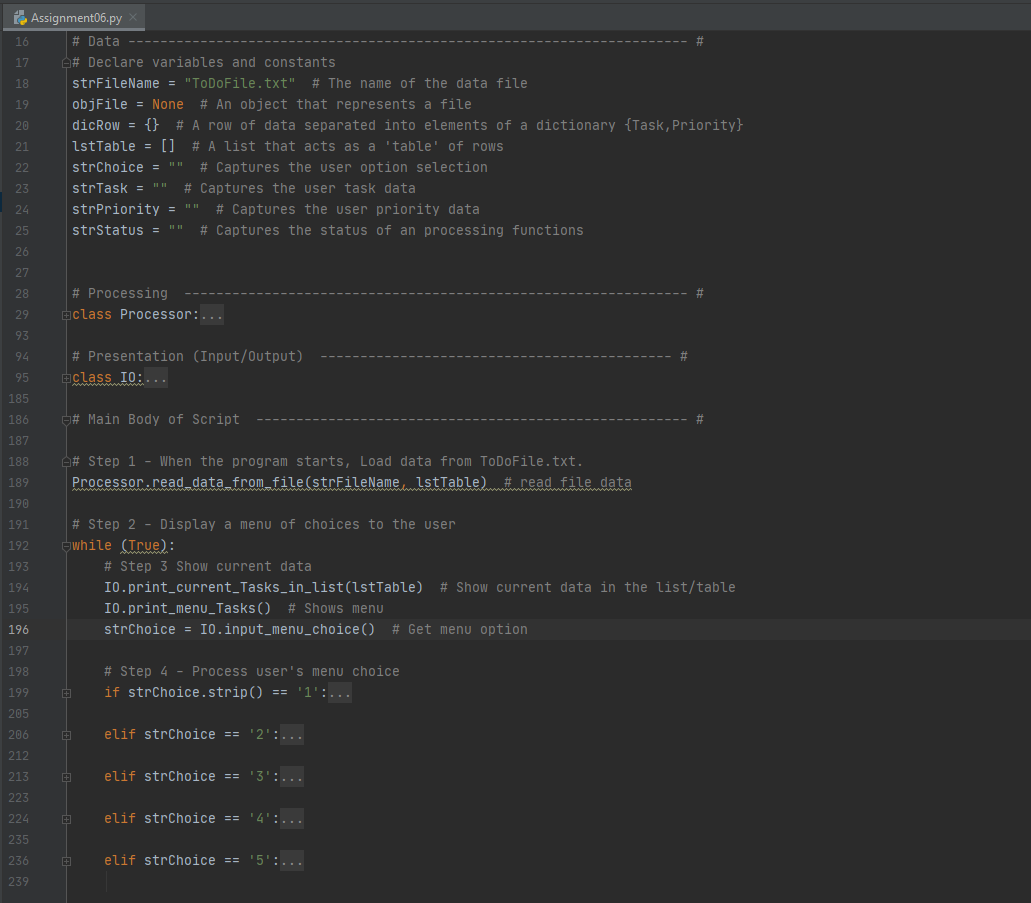
**Created:** 05/25/2020 by Kate Golenkova

**Introduction**

In this assignment I will show how I created a new python script from a template. This script loads each row of data from “**ToDoFile.txt**” file into a Python dictionary and then adds each dictionary row to a Python list (Table). To do so program reads file and prints it out asking user to choose the option from the **Menu** what he wants to do with data – add more data, store it in text file, print it out on the screen, or just exit the program. This assignment is very similar to Assignment05, but now we will use functions to get script works. I will use screenshots to illustrate how the script has been created and tested.

**Pseudo-Code of Script**

In this assignment again I used template with good To Do list, showed in **Figure 1.** below:



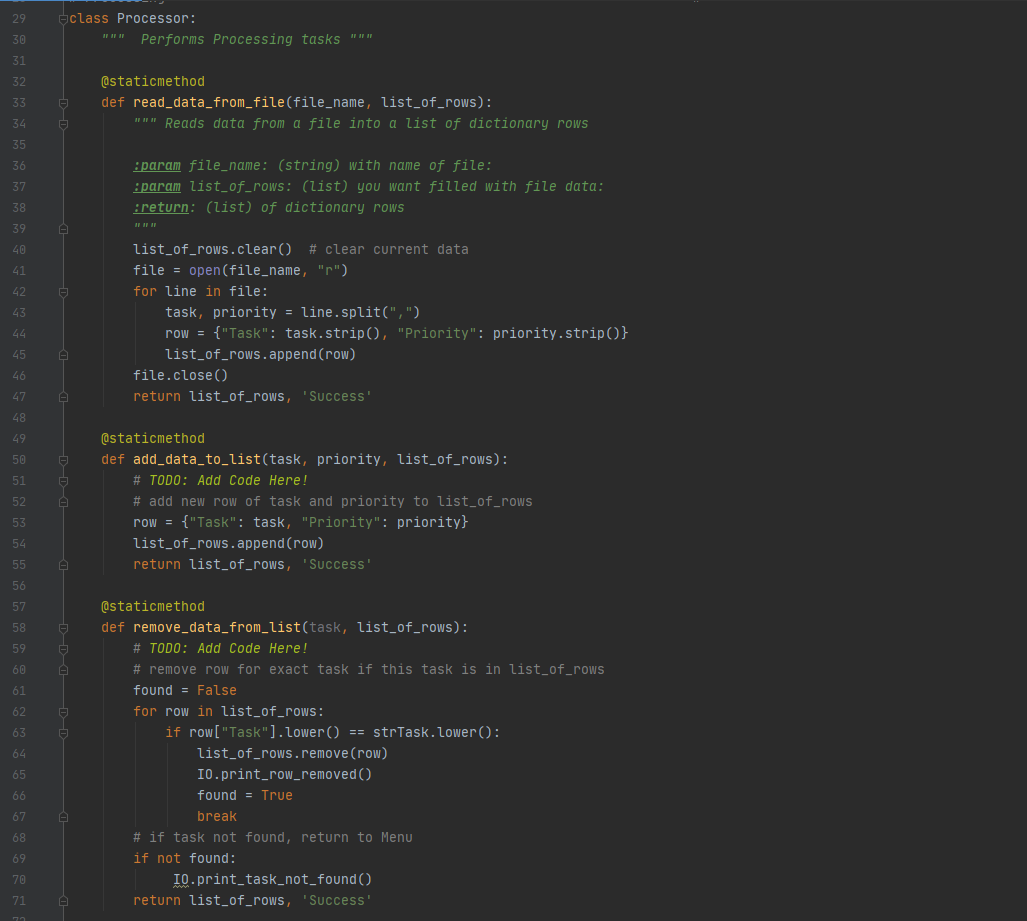
**Figure 1.** The template for new script

The task for this assignment was to create new python script using template and adding more functions. As you can see in **Figure 1**. there are two classes of functions **Processor** and **IO** and main body with while loop. The **Processor** class contains functions that perform processing tasks, IO class contains functions that perform Input/Output tasks.

Also, all the variables were declared already. So, I just created text file ToDoFile.txt and saved the template in my working folder Assignment06 and started to work on my script by adding functions and testing each step.

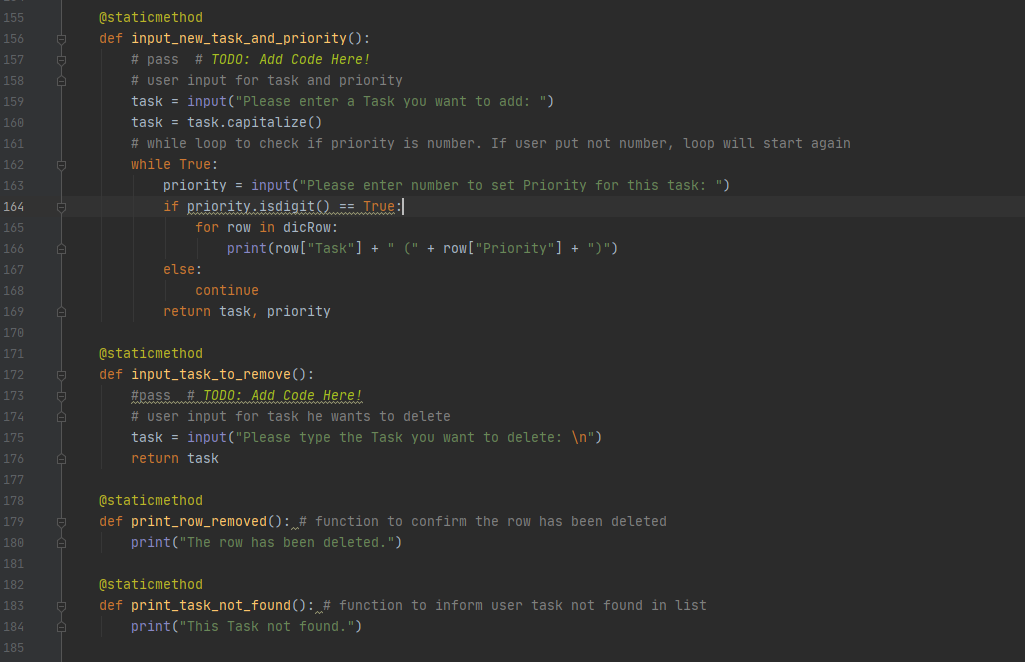
**Python Script that Manages File ToDoFile.txt**

I started to work on my script with **Processor** class, functions for adding, removing and writing data, as we did almost the same in Assignment05. You can see the code with these functions in **Figure 2.** below:



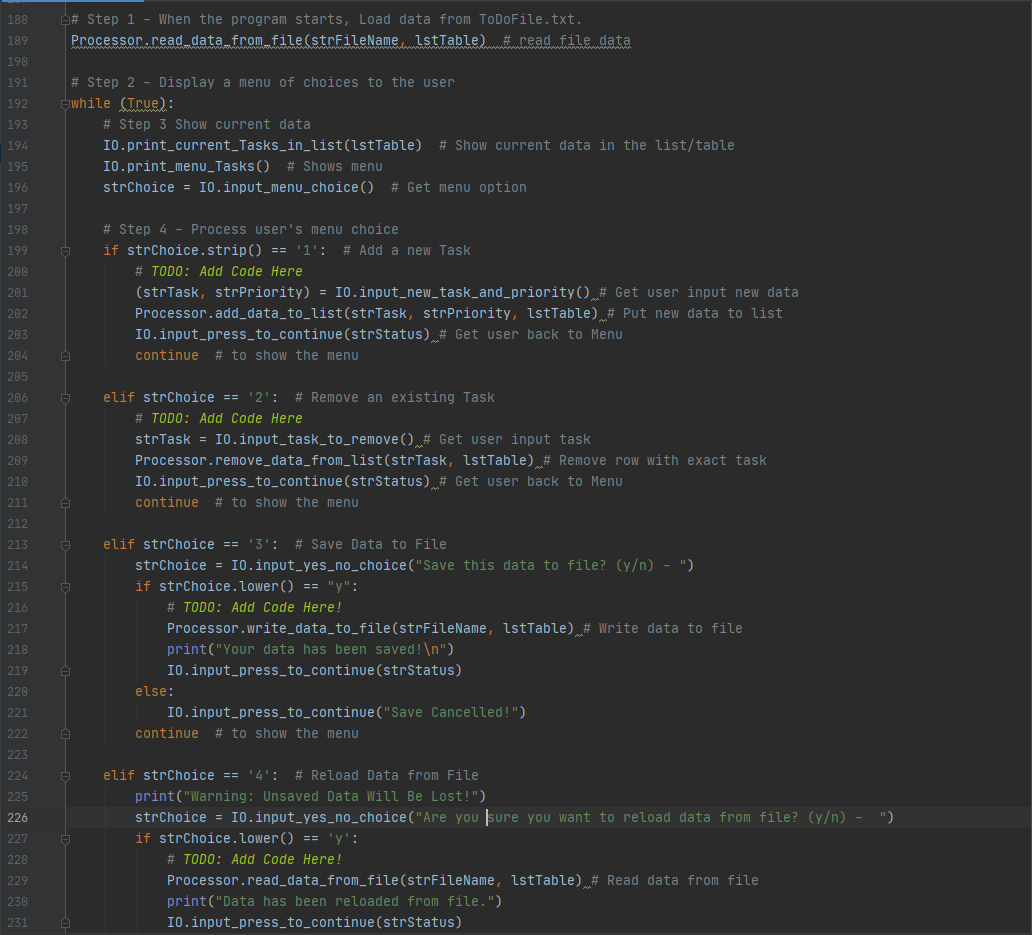
**Figure 2.** The Processor class with new functions.

Then I created functions for **IO** class to get user input data for adding and removing data. Also, I added two very simple functions for printing messages for user. I used them in functions that I created for Processor class. Please see **Figure 3.** on next page to review these functions:



**Figure 3.** IO class new functions.

After all my functions were created, I just added them to the main body of the script with proper arguments. Please see Figure 4. below:

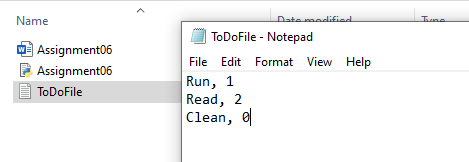


**Figure 4.** Main body of the script.

**Tests of Python Script in PyCharm and CMD**

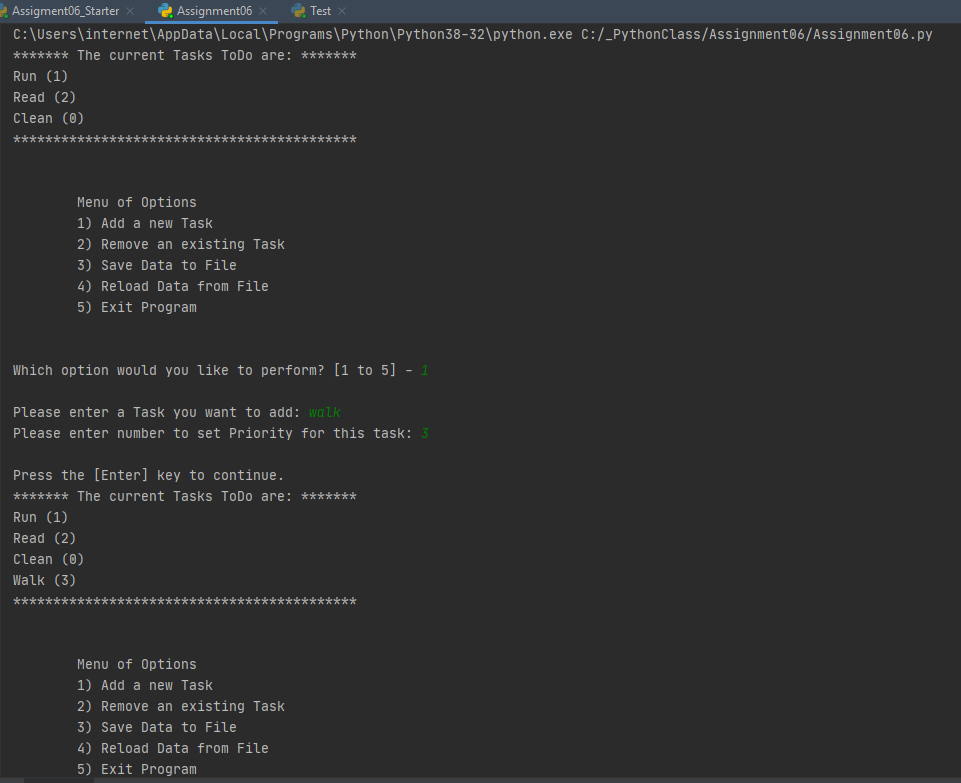
I tested my code almost hundred times during its creation and I just cannot show all the tests in this document. So, I will test it again and show output from PyCharm and Command Line, and results from text file.

First, I put some tasks and priorities in text file. In **Figure 5.** you can find the file ToDoFile.txt with them:



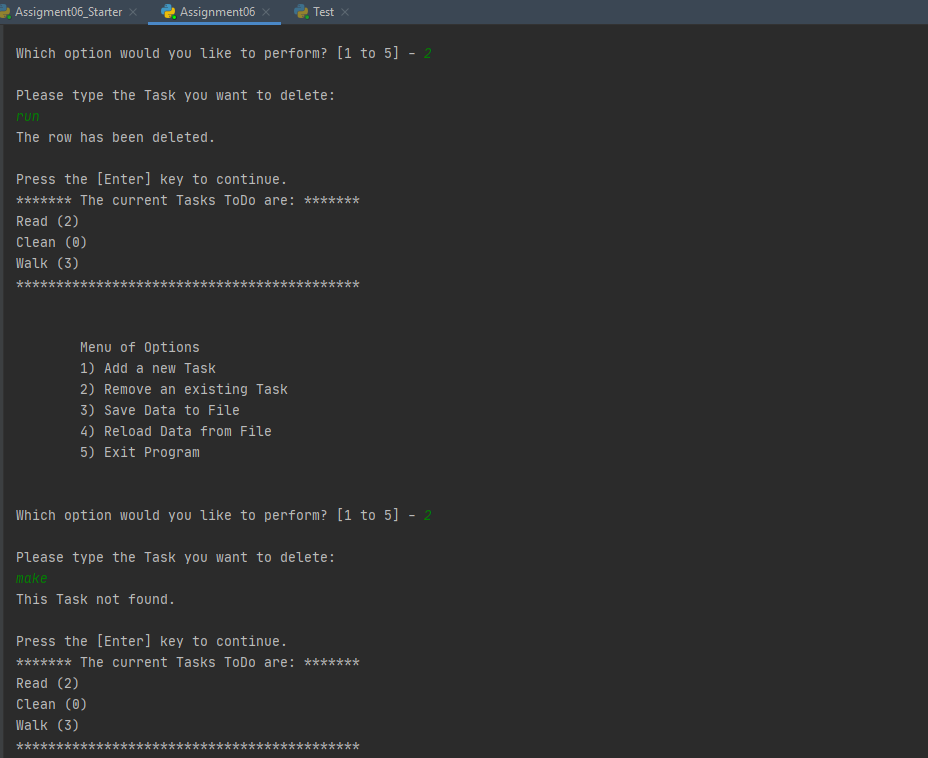
**Figure 5.** The file ToDoFile.txt contains tasks and priorities.

The file has been saved and I run my script in PyCharm. As you can see in **Figure 5.** below the script shows current data from file. Also, you can see that I used option 1 from the Menu to add new task. New task **WALK** has been added to current data.



**Figure 5.** Testing the first option of Menu (PyCharm).

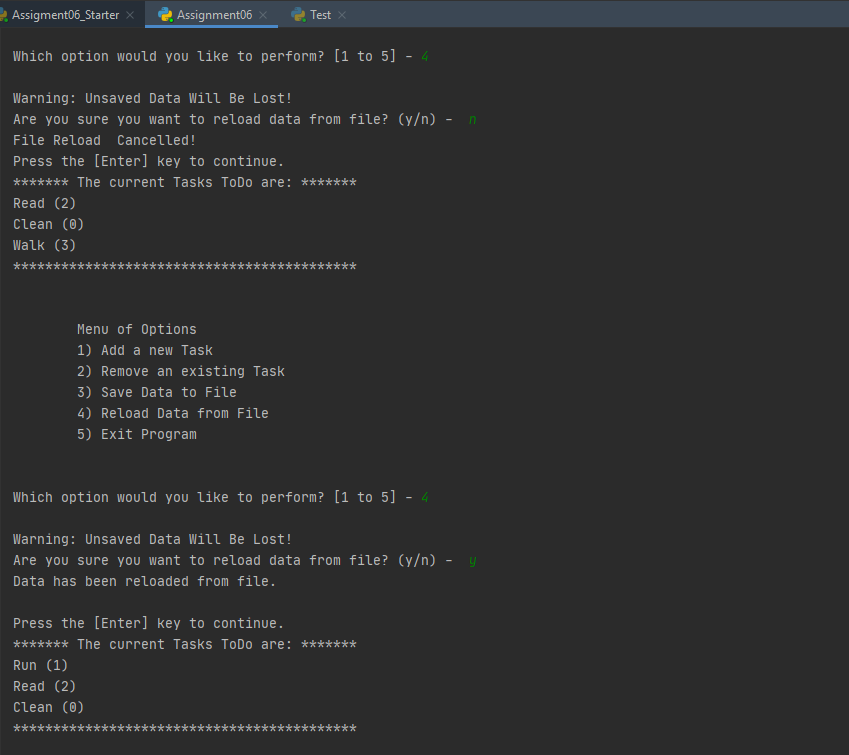
Then I tested option 2 and deleted task **RUN** from the list. To show that my script can check if the task is in the list, I tried to delete task **MAKE** that is not my list. Please see the result in **Figure 6.** below:



**Figure 6.** Testing how script removes data (PyCharm).

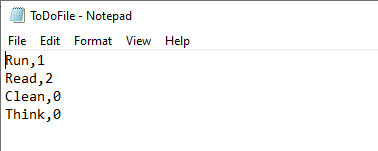
In the Menu there is option 3 that goes after option 2, but I thought it would be better to test reloading data before saving. So, I choose option 4 and the result of this test you can see in **Figure 7.** on the page 6.

Before the data has been reloaded from the file, I successfully tested if I can cancel reloading data in case, I change my mind.



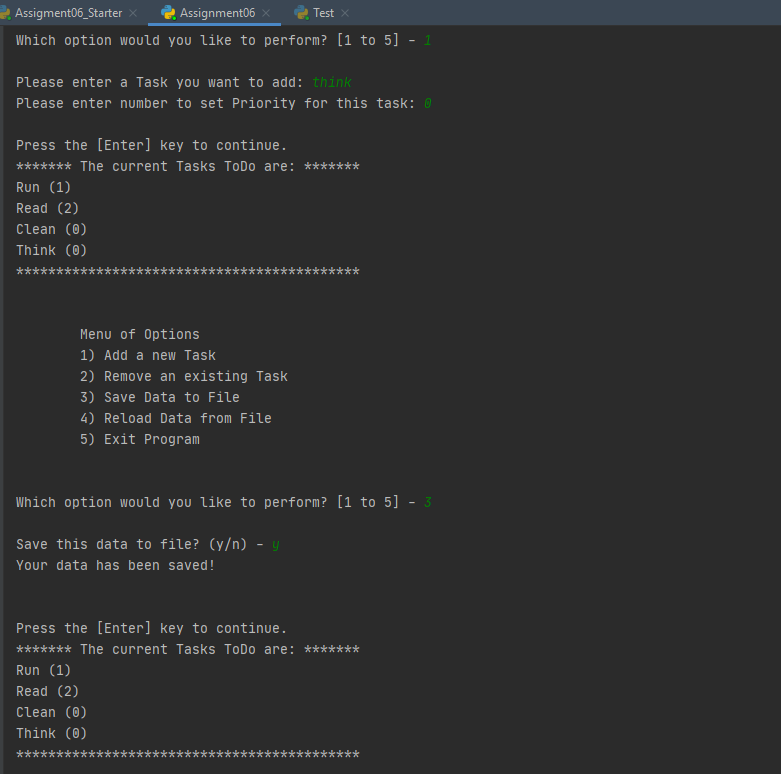
**Figure 7.** Testing the reloading data from file (PyCharm).

To check if I can successfully save data in my file ToDoFile.txt, I added new task **THINK** and saved data using option 3. Please check the text file in **Figure 8.** that contains now four rows of tasks and priorities:



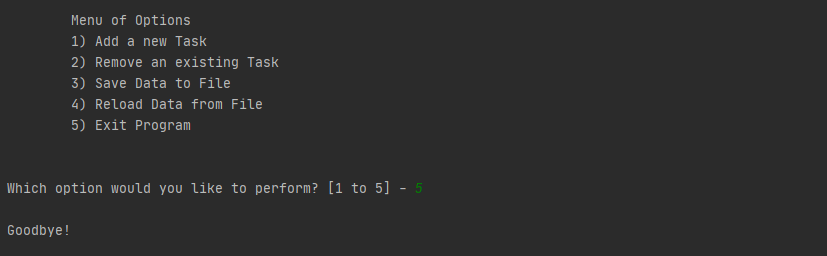
**Figure 8.** New data has been saved in ToDoFile.txt

In **Figure 9.** You can see how new data has been added to a list and how option 3 from the Menu worked:



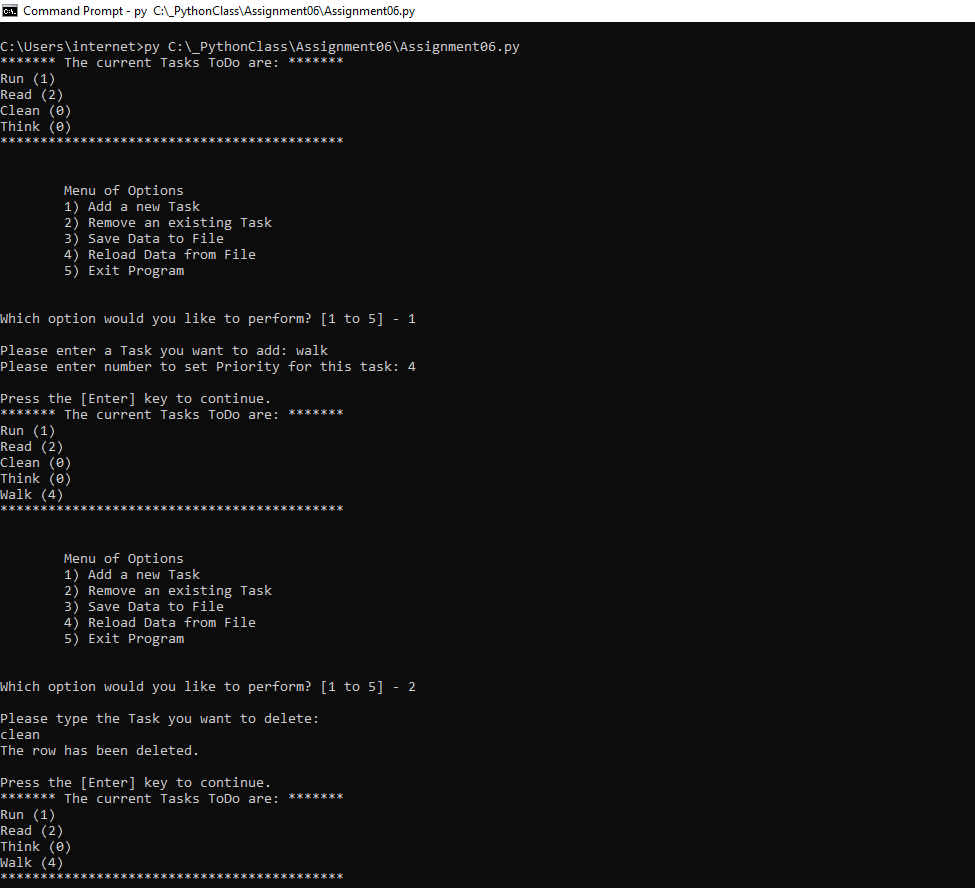
**Figure 9.** Adding new data and saving it to a file (PyCharm).

After all tests of my script in PyCharm have been provided, I choose option 5 to exit the program:



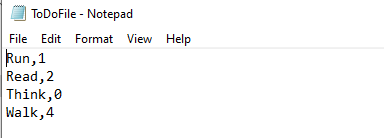
**Figure 10.** Exit the program (PyCharm).

To test my script in Command Line I started with adding and removing data. In **Figure 11.** You can see that I added new task **WALK** and removed task **CLEAN**.



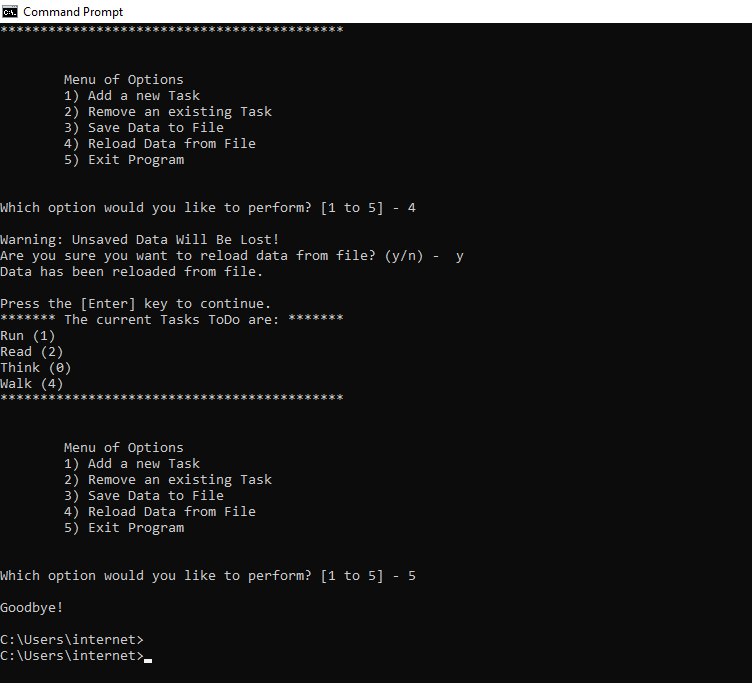
**Figure 11.** Adding and removing data (Command Line)

Then I successfully tested options for saving and reloading data and finally got the next data in my text file that you can see in **Figure 12.** below:



**Figure 12.** new data has been saved to ToDoFile.txt (Command Line)

After first four options have been tested successfully in Command Line and I have got new data in my file ToDoFile.txt, I choose option 5 to exit the program, as you can see in **Figure 13.**:



**Figure 13.** Exiting the program (Command Line)

**Summary**

I read the next chapter in the book and Programing Notes for Module06, I also watched all the videos and used external resources to understand how to operate with functions, classes, local and global variables. As a result, I was able to add my code to new template with functions, classes and loops to get user choose the option and add/delete/save data. Also, I tried to keep it simple as usual.