Sherin Soman

August 16, 2021

Assignment 06

<https://github.com/SherinJoel/IntroToProg-Python-Mod06>

**Create a script using custom functions**

**Introduction**

In this Assignment, I explained the steps I have done to create a script using custom functions and text files. I start my program by loading the data in a text file called ToDoFile.txt into a python list of dictionaries rows. The script continues to display a menu of choices to the user until the user ask to exit the program. I used a printed "menu" to guide the user through this process.

**Python Functions**

A function is a block of code which only runs when it is called. We can pass data, known as parameters, into a function. A function can return data as a result.

In Python a function is defined using the defkeyword.

To call a function, use the function name followed by parenthesis.

*Benefits of Functions:*

* Functions allow the same piece of code to run multiple times
* Functions break long programs up into smaller components
* Functions can be shared and used by other programmers

**Python Class**

A class is a user-defined blueprint or prototype from which objects are created. Creating a new class creates a new type of object, allowing new instances of that type to be made. Each class instance can have attributes attached to it for maintaining its state. Class instances can also have methods (defined by their class) for modifying their state.

***Python Static method:***

Any method we create in a class will automatically be created as an instance method. To make a method a static method, add @staticmethod decorator before the method definition.

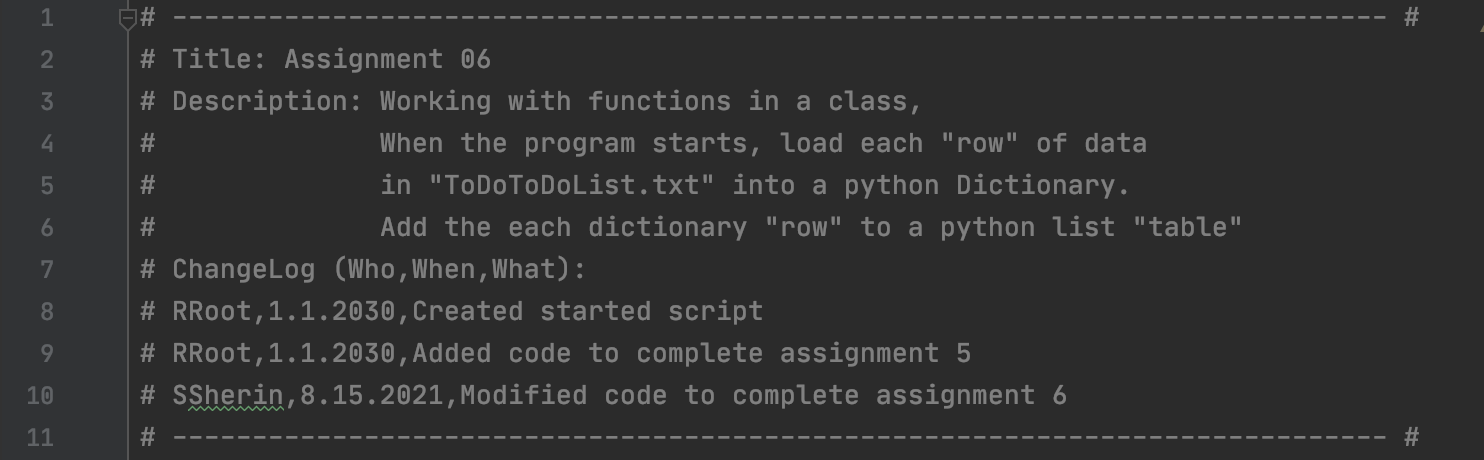
A static method is bound to the class and not the object of the class. Therefore, we can call it using the class name.

**Create a new Project in PyCharm**

To create a new Project in PyCharm, I created a sub-folder called Assignment 06 inside of the \_PythonClass folder and used \_PythonClass\Assignment06 as its location to create the new project. Within the project, I added the starter file, "Assignment06\_Starter.py.

**Adding code to the Python Script**

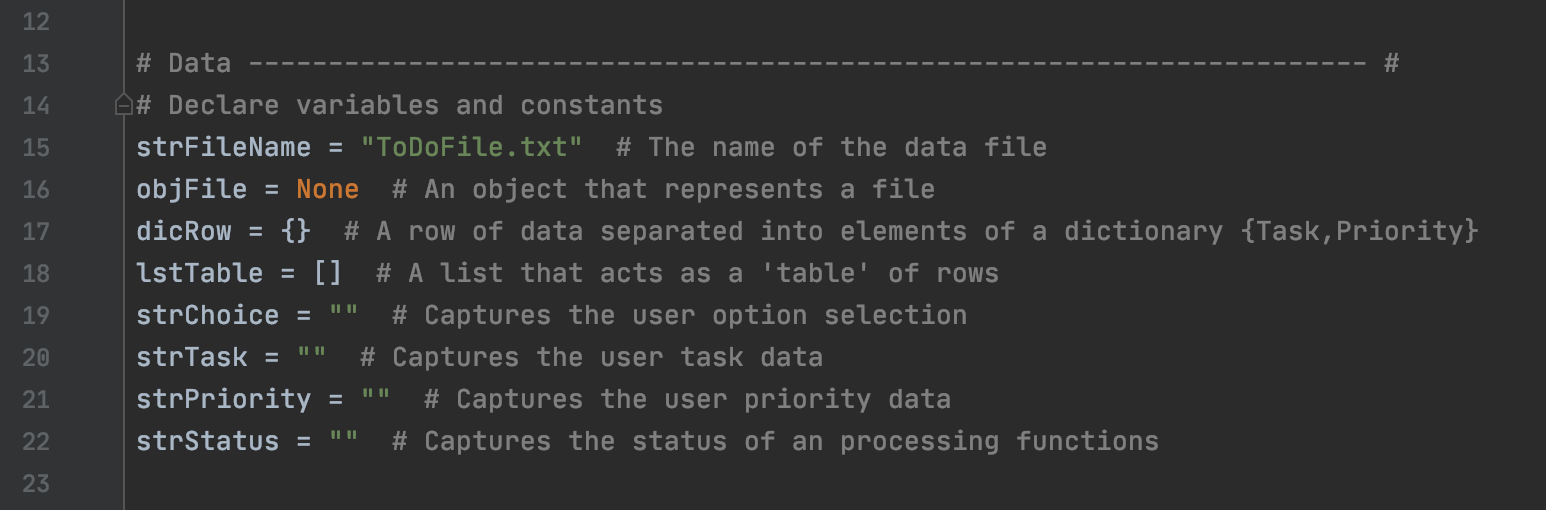
I started my script by updating the change log in the script's header.

*Figure 1: Script Header*

**Declaring variables and constants**

Variables are containers to store values. Variable declared outside of the function or in global scope is known as a global variable. This means that a global variable can be accessed inside or outside of the function.

A variable declared inside the function's body or in the local scope is known as a local variable.

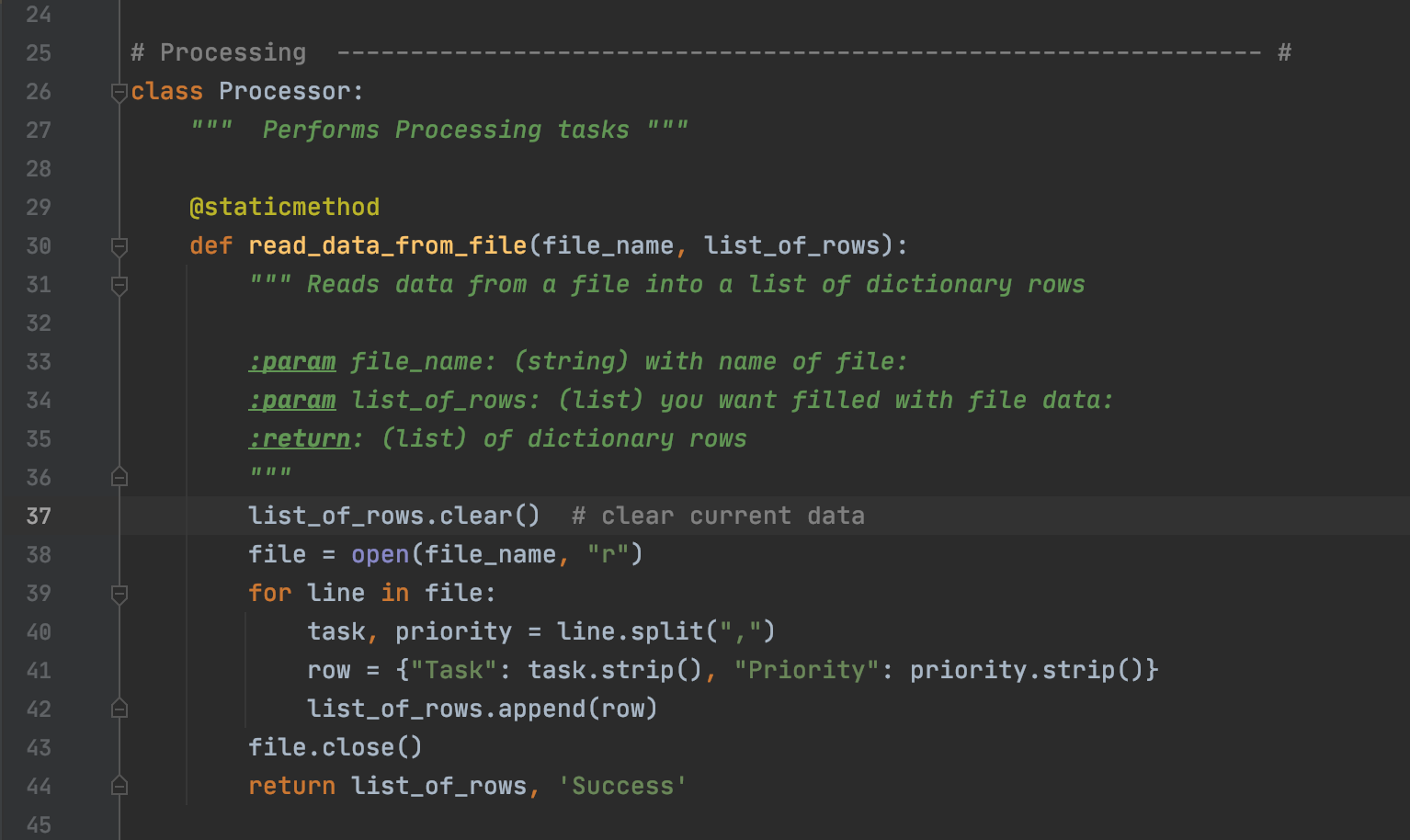
*Figure 2: Declaring Variables and Constants*

**Performing Processing Tasks – Step1**

In this script, Tasks to be performed are defined in a Class Processor.

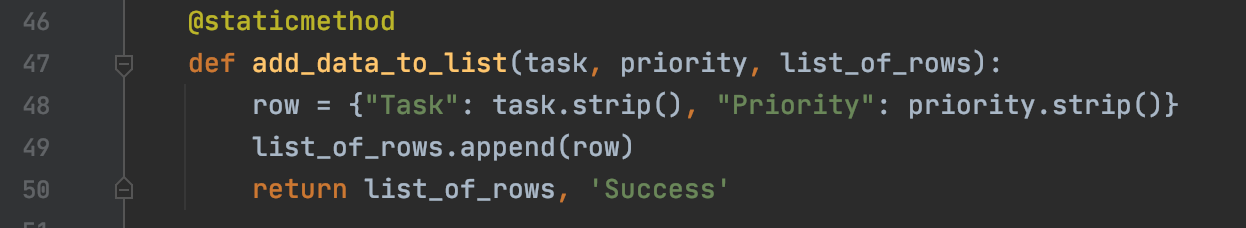
***Reading data from a file into a list of dictionary rows***

This task is defined by a function which returns the list of dictionary rows. To read data from a file, the file is opened in “r” mode. Using a for loop, each row in the text file is read and splits at the specified separator (comma) using the split() method and is stored in the variables task and priority. These values are then appended to the list as dictionary rows using the append(). The file is closed using the close().

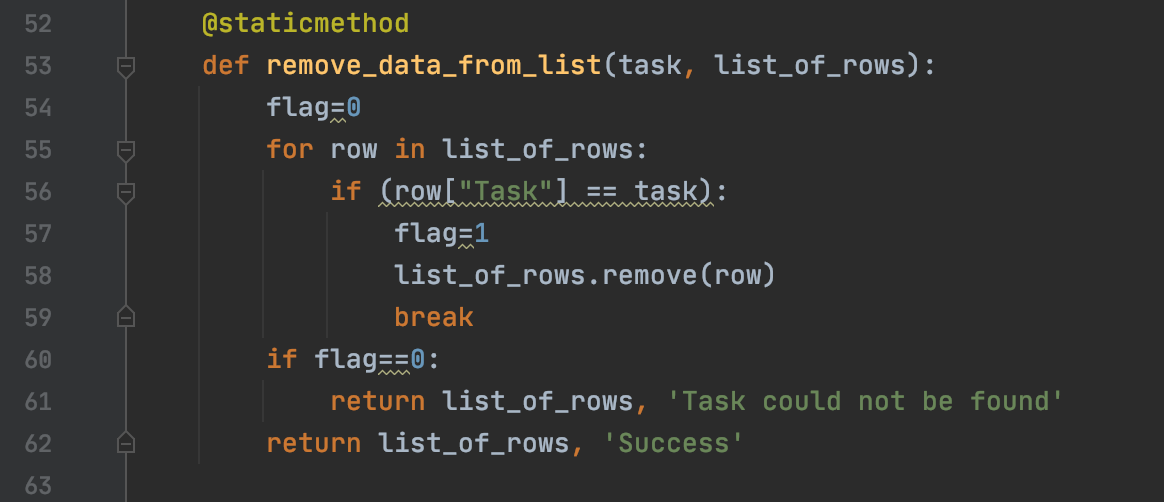
*Figure3: Function reading data from file and returning the list of dictionary rows*

***Adding data to List***

This function gets the new task, new priority and the current list of rows as its parameters and returns the list after appending the data and the string 'Success' to indicate the status of the operation.

*******Figure 4: Function adding new task and priority to the list and returning the list of dictionary rows*

***Remove data from List***

This function accepts the task to be removed and the current list as its parameters. Using a for loop, each dictionary row task is checked with the task to be removed and if a match is found, the task and priority of the corresponding row is removed. This function returns the list of rows (after removing the task) and the string 'Success' if a match is found and if not found, it returns the list of rows (without changing) and the string 'Task could not be found'.

*Figure 5: Function removing the task in the list and returning the new list*

***Write data to file***

The file is opened in write mode and the elements from the list are written to the file.

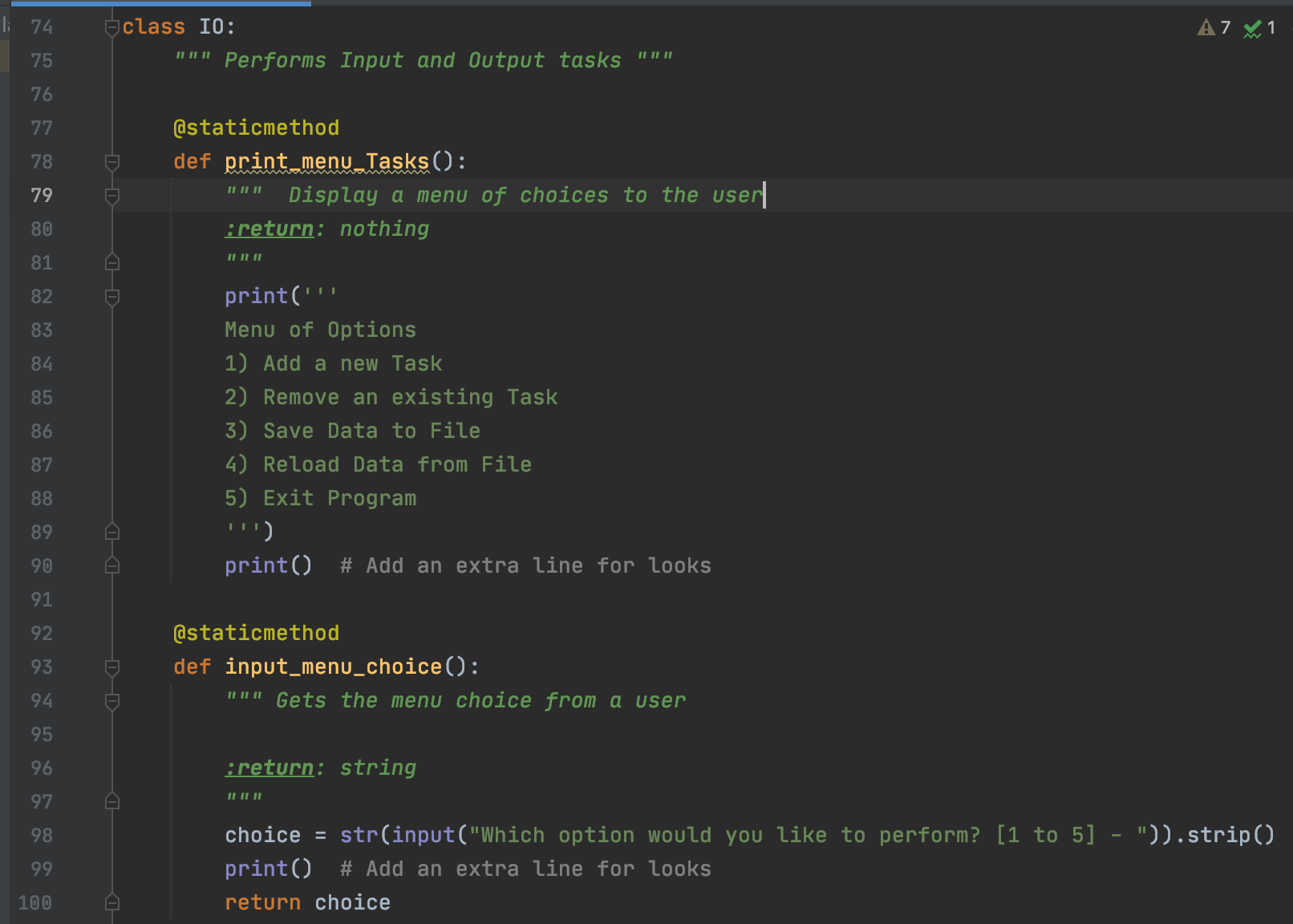
*Figure 6: Function writing data to file*

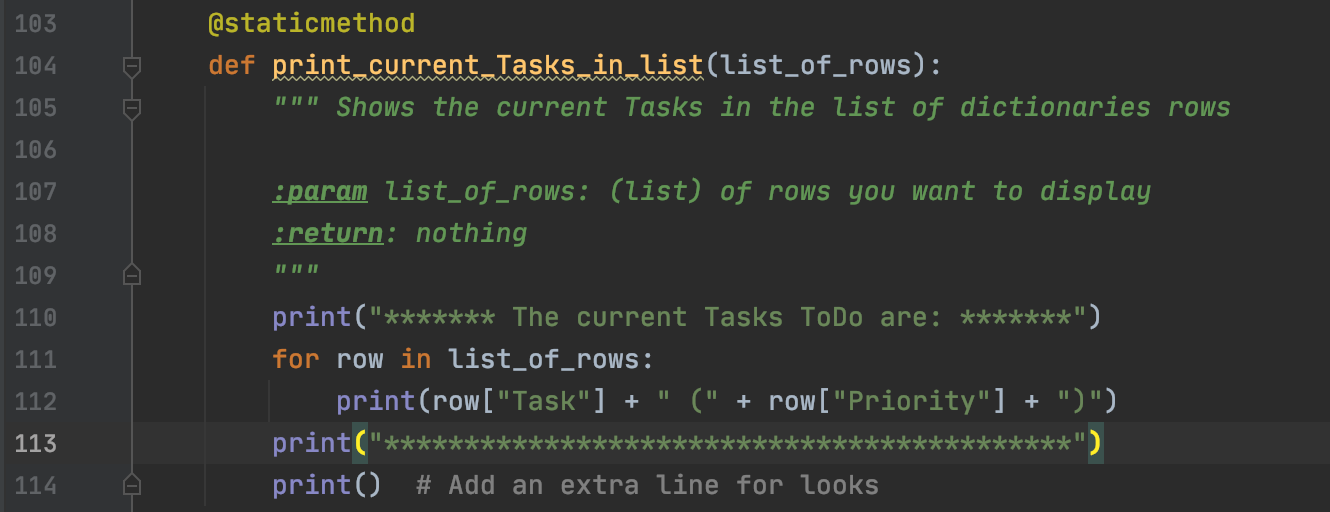
**Performing Input and Output Tasks – Step2**

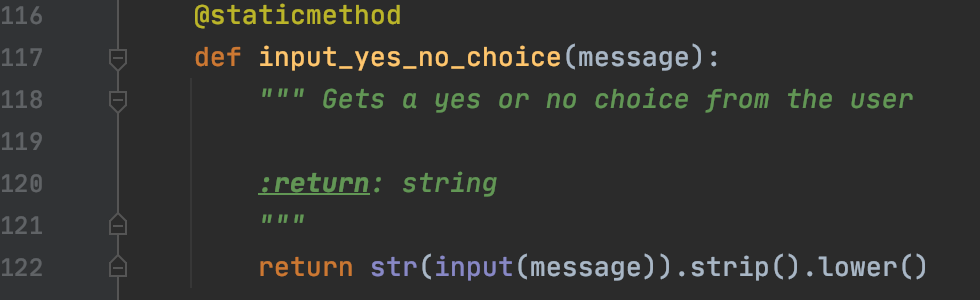
The input received from the user and the output displayed are defined in a class IO.

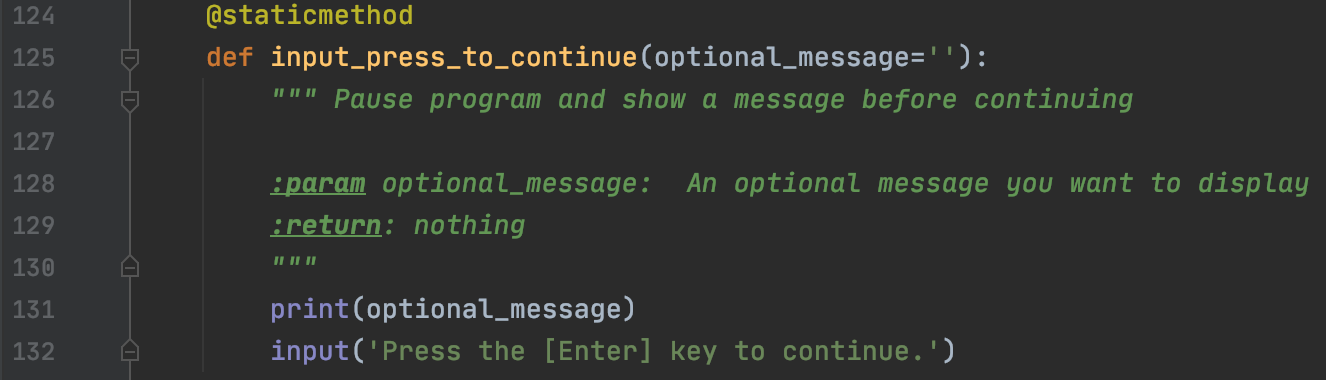
***Displaying a menu of choices to the user and getting the user input choice***

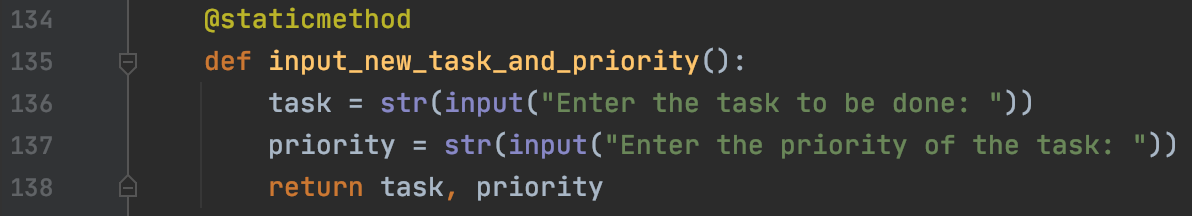
Menu choices are displayed to the user using the function def print\_menu\_Tasks. The function input\_menu\_choice gets the user input choice of which task to be performed and returns the choice.

*Figure 7: Displaying menu choices and getting user input choice*

*Figure 8: Displaying the current Tasks in the list of dictionaries rows*

*Figure 9: Getting a yes or no choice from the user*

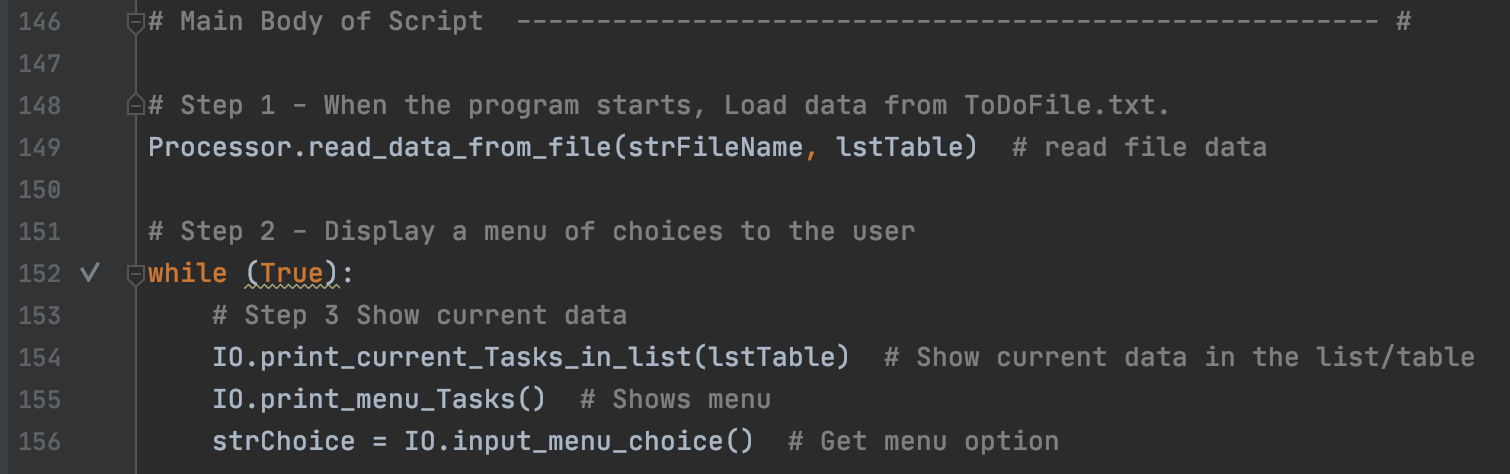
*Figure 10: Displaying a message before continuing*

*Figure 11: Getting the new Task and Priority to be added to List*

*Figure 12: Getting the name of Task to be Removed from List*

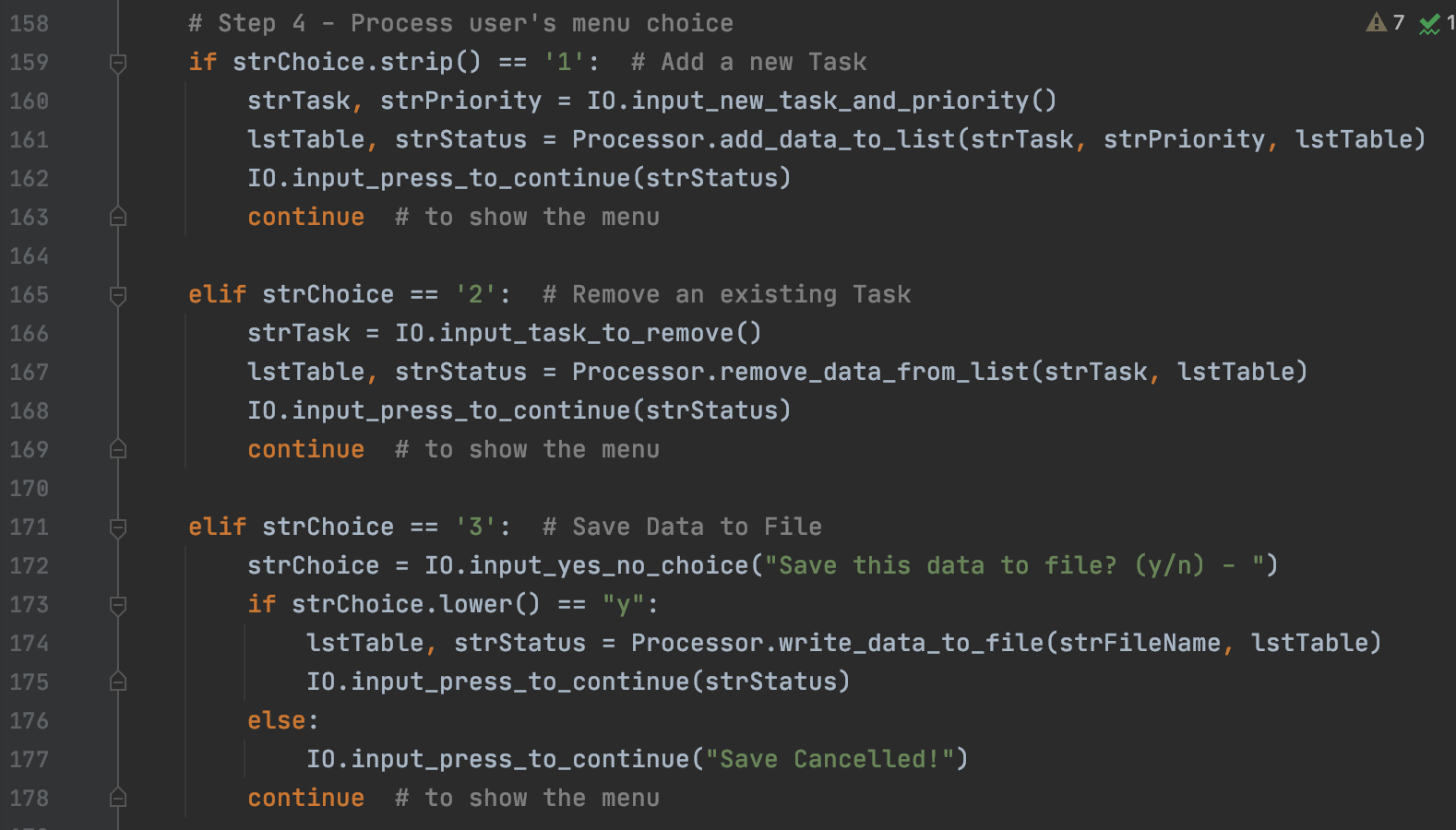
**Calling functions in the main body of the script – Step 3**

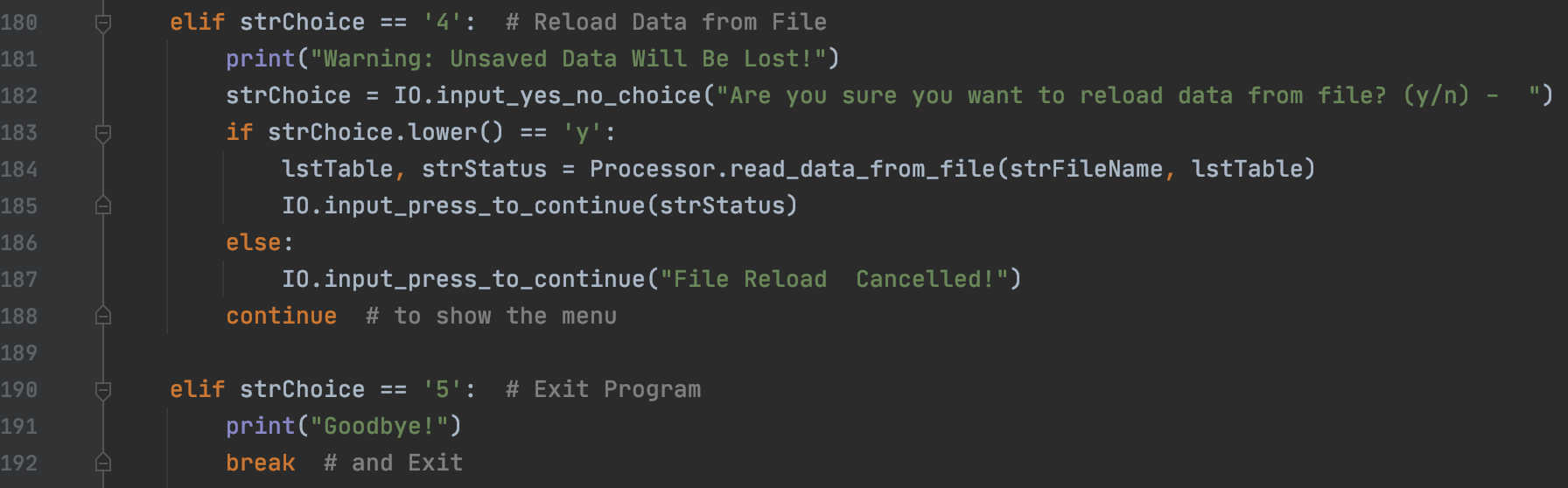
As the program starts, the data from the file is loaded into the list by calling the function read\_data\_from\_file in the Class Processor. After loading the file data into list, the current data in the list is displayed by the calling the function print\_current\_Tasks\_in\_list in the Class IO.

Then the menu choices are displayed, and the user input choice returned from the function is captured in the variable strChoice.

*Figure 13: Calling function to display the current list and getting the user input choice*

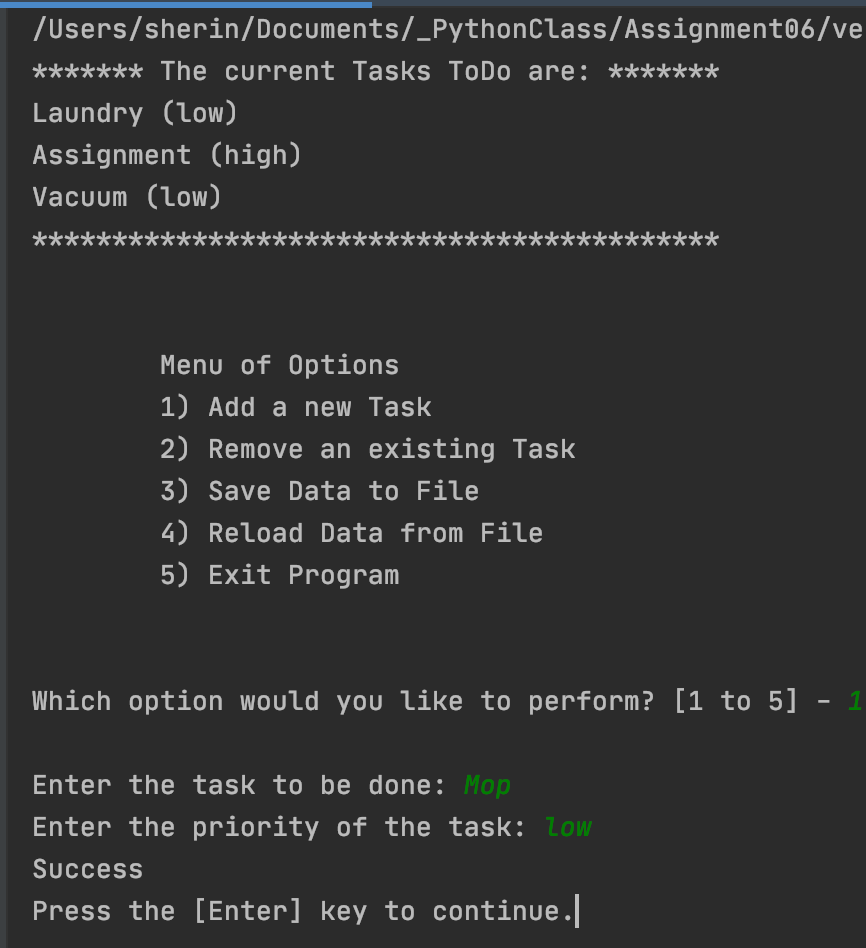
If the user choice is 1, then the function to get the new task and priority from the user is called. The task and priority returned from that function is passed as arguments to the function add data to list when it is called. The list of rows and the string 'Success' returned from the function add\_data\_to\_list is captured in the variables lstTable and strStatus. After each step, a message is displayed before continuing by calling the IO.input\_press\_to\_continue(strStatus) function where the messages 'success' and 'press Enter to continue' are displayed.

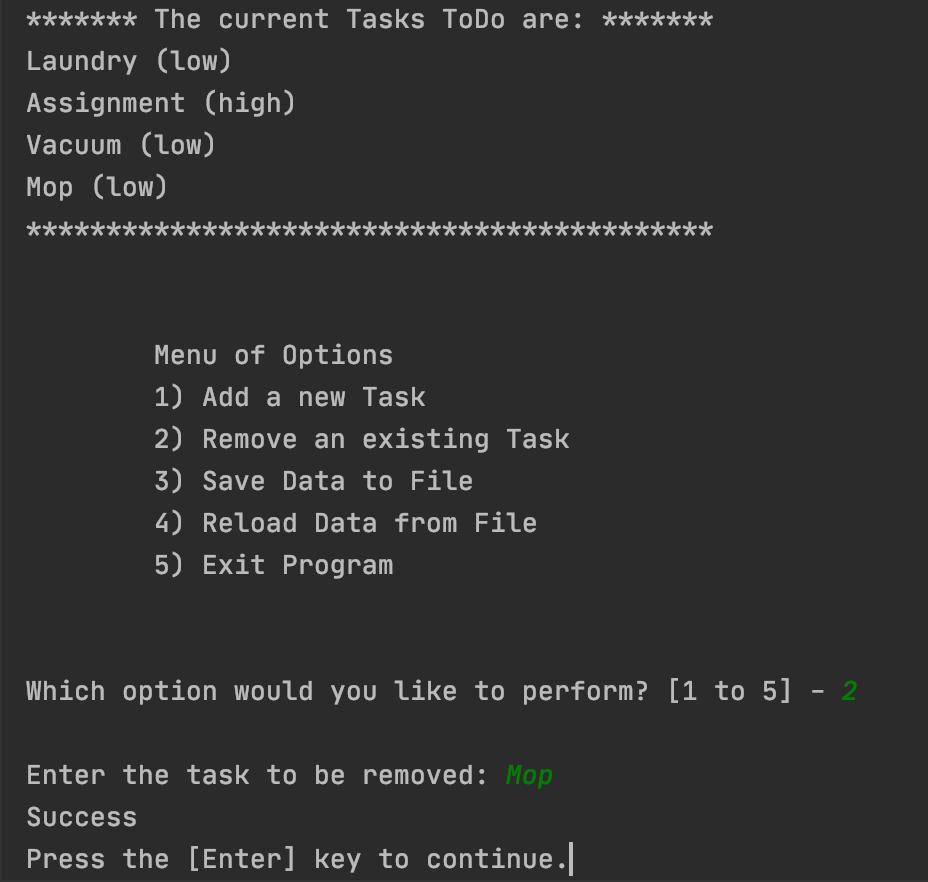
*Figure 14: Calling function to add, remove and save data to file*

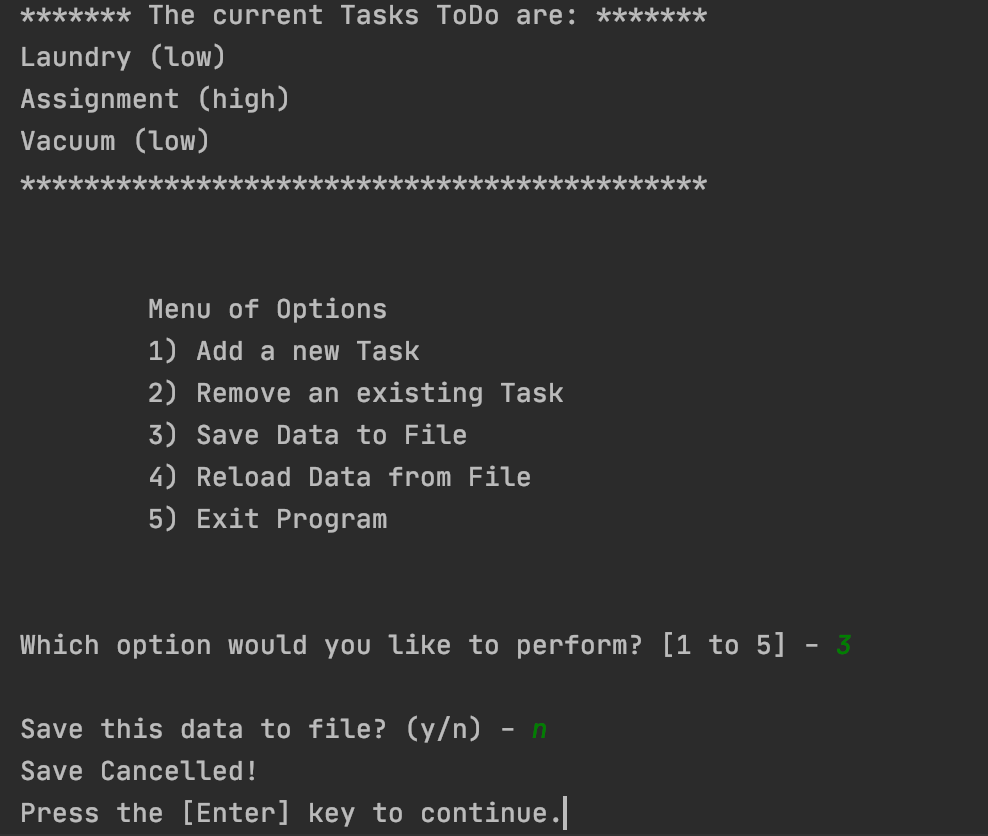
*Figure 15: Calling function to reload data from file*

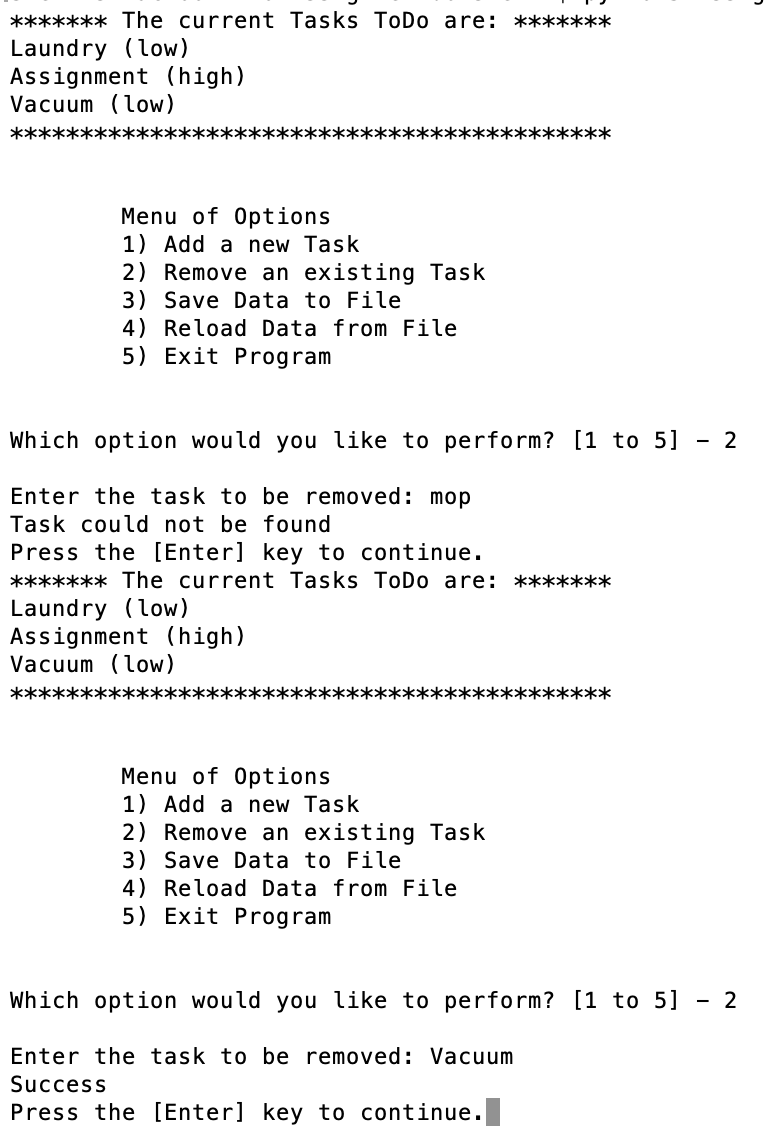
***Running the Script***

With the script created in its proper location, I run the script in both PyCharm (Figure 16) and an OS command/shell window (Figure 17).

*Figure 16.1: Adding new task and priority*

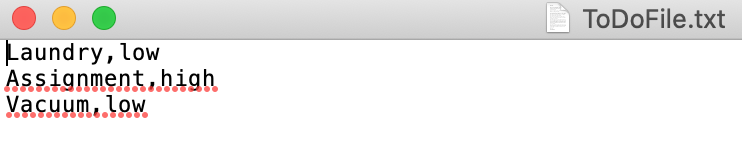
*Figure 16.2: Removing task*

*Figure 16.3: Saving data to file*

*Figure 17: A screenshot of the script running in Command Window.*

***Verifying the result***

Locate the text file and open it in a text editor

*Figure 18: Verifying that the file contains data*

***Summary***

In this assignment, I was able to write a python script using functions, lists, dictionaries, and text files. The script is executed both in PyCharm and in command window to verify the results and the script ran as expected. I also post the files on a public GitHub repository to review.