Bhawna Tyagi

November 26, 2021

Foundations of Programming: Python

Assignment 06

<Github Link>

How to incorporate functions and classes to create menu in python script?

# Introduction:

This document will walk you through the steps that I have taken to create a menu for the user using functions and classes in the script. The script performs certain operations on the list of dictionary objects. The various functions have been wrapped under two different classes i.e., IO and Processor classes. The IO class contains functions which are going to show the data in the list. However, the processor class contains functions which are responsible for processing and manipulating the dictionary objects (i.e., addition of the dictionary objects in the list, removal of the dictionary objects from the list and saving data to the file along with reuploading data from the file without saving the data in the file).

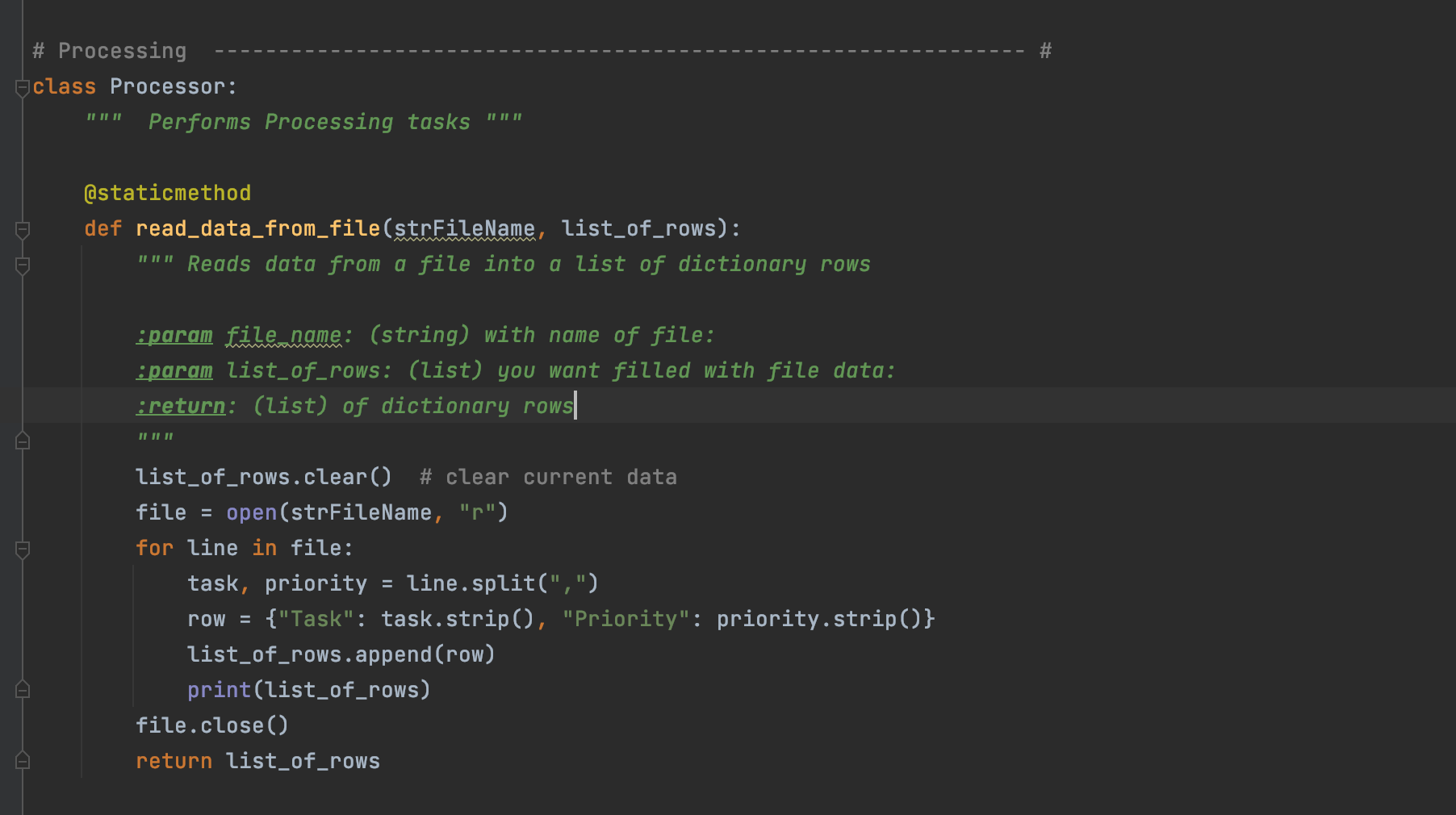
# Break down of the script:

As mentioned above in the introduction section, there are two types of classes in the script. In this section, I am going to explain what each and every function definition is along with the task that has been performed by each one of them in their respective class.

# Processor class:

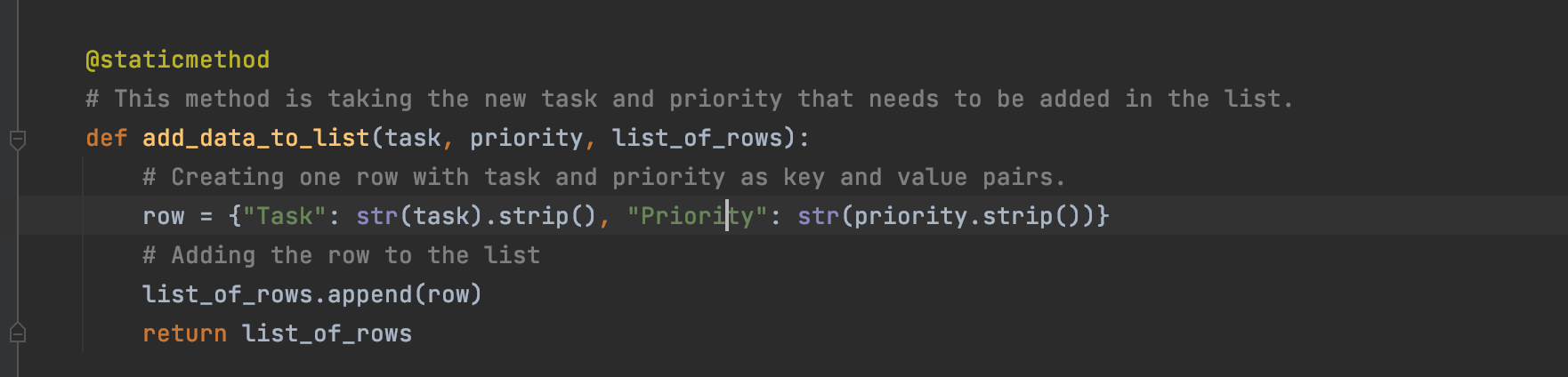
This class processes tasks and contains following functions:

**Read\_data\_from\_file (strFileName, list\_of\_rows):** This function is responsible for reading rows from the existing file i.e., “ToDoList.txt” text file and displays the list back to the user as a result. The function starts with opening the file in the read mode using file variable, iterates through the file line by line using line variable. The line variable grabs the first line in the file, splits the line into task and priority on ‘,’ separator using split () and create row of dictionary. Finally, append the row to the list i.e., list\_of\_rows. This process repeats until the end of the file. The file gets closed once we read each row from the file. At the end, I returned the list of rows to the user. The complete function is shown in Figure 1.



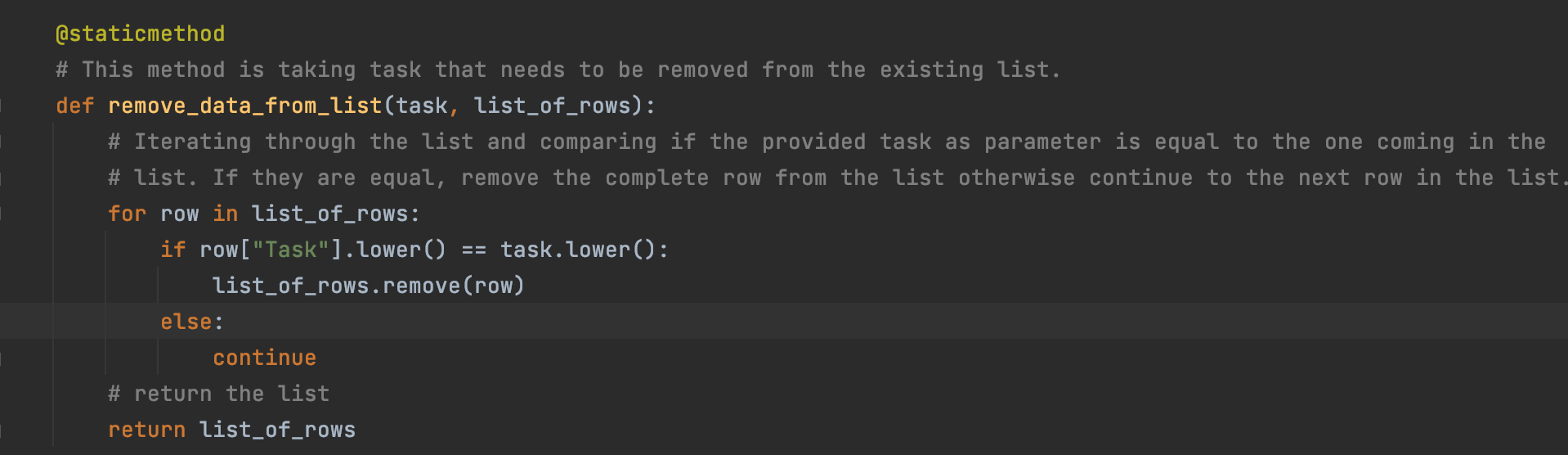
***Figure 1. read\_data\_from\_file () code***

**Add\_data\_to\_list (task, priority, list\_of\_rows):** Thisfunction is responsible for adding more new rows/data to the existing list. It is taking new task, new priority and the existing list of dictionary objects as input parameters. Creating one new row as dictionary object and adding that new row to the list using append (). Finally, returning the list to the user. The complete function code is shown in Figure 2.

****

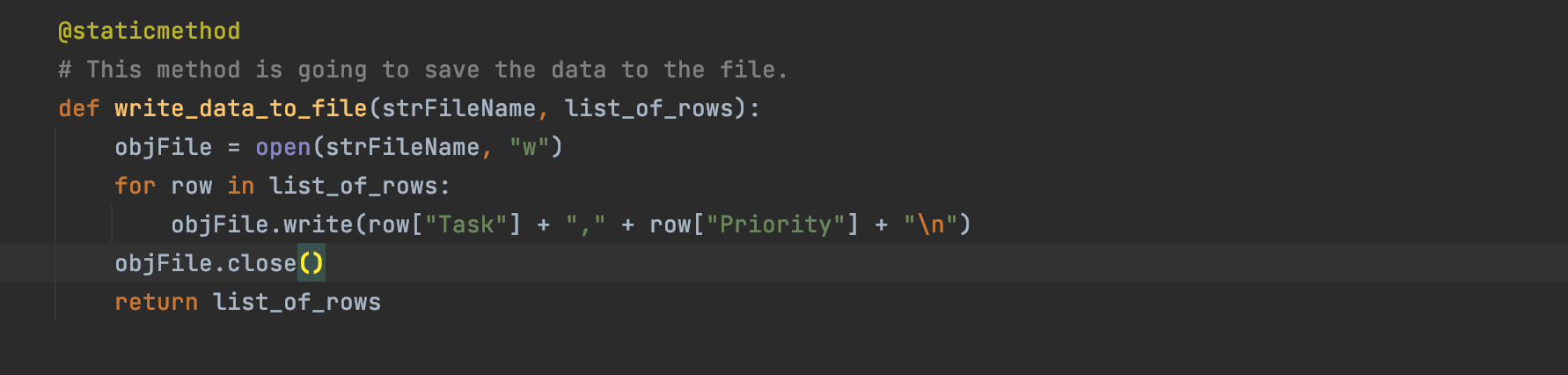
***Figure 2. add\_data\_to\_list () code***

**Remove\_data\_from\_list (task, list\_of\_rows):** This function is responsible for removing the rows from the existing list. It begins by iterating through the list via row variable and comparing the item coming from the list with the parameter passed into the function. If they match then we are entire row from the list. If not then continue to the next task using continue keyword. This process continues until we find the required task that we want to remove the list. Finally, return the list as result to the user. The function code is shown in Figure 3.

**

*Figure 3. remove\_data\_from\_list () code*

Write\_data\_to\_file (strFileName, list\_of\_rows): This function is responsible for saving data to the file. Initially, the file gets opened in the write mode using objFile variable. Then, iterate through the list via row variable and write the new row to the existing list. Once the new data gets written to the list, the file gets closed using close (). Finally, returning the list to the user as a result. The complete function code is shown in Figure 4.

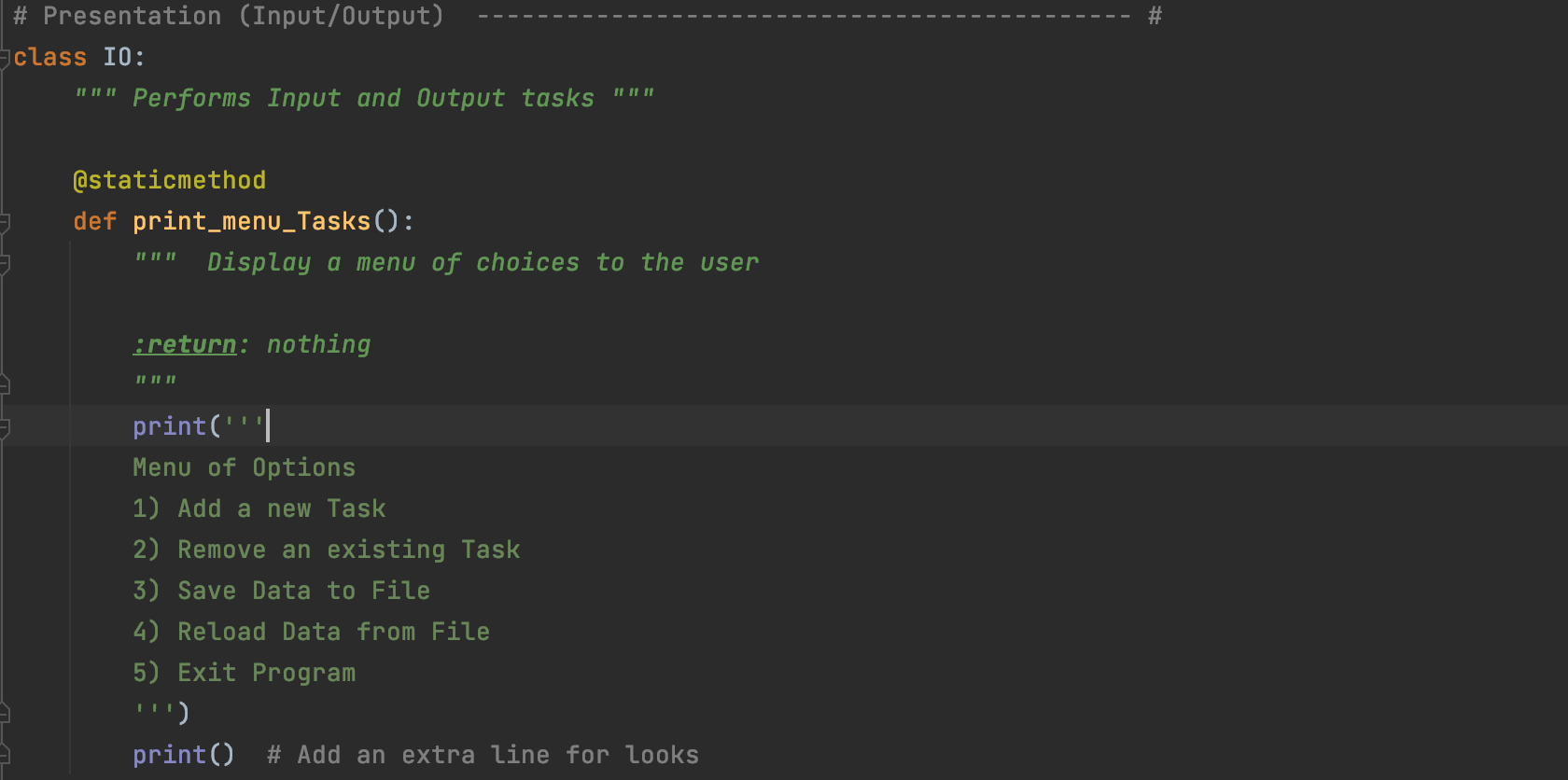


***Figure 4. write\_data\_to\_file () code***

# IO class:

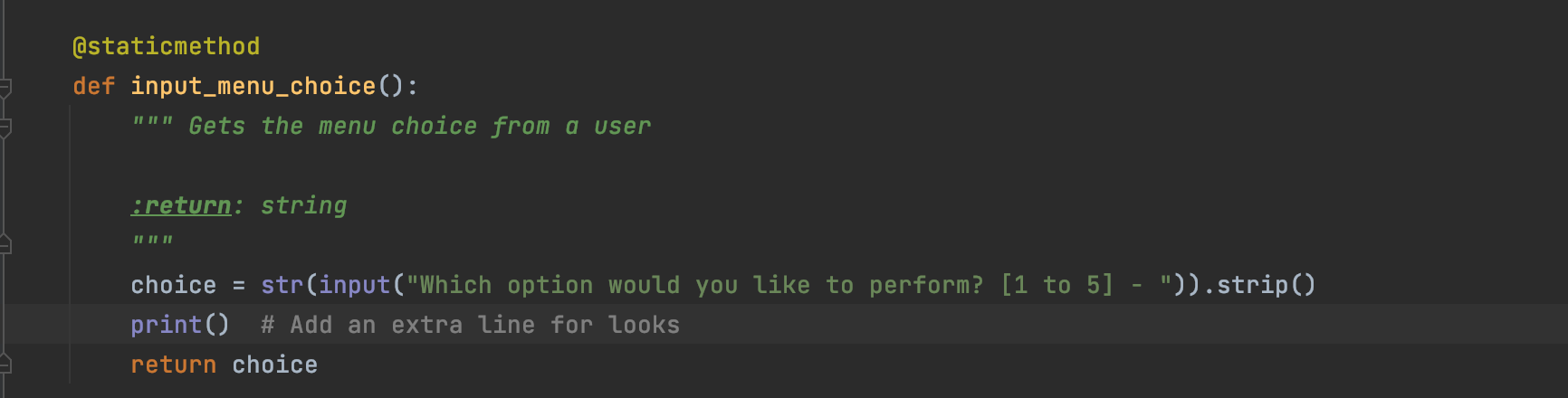
This class performs input and output tasks. It contains following functions:

Print\_menu\_Tasks () : This function is responsible for showing the menu of options to the user as shown in Figure 5.



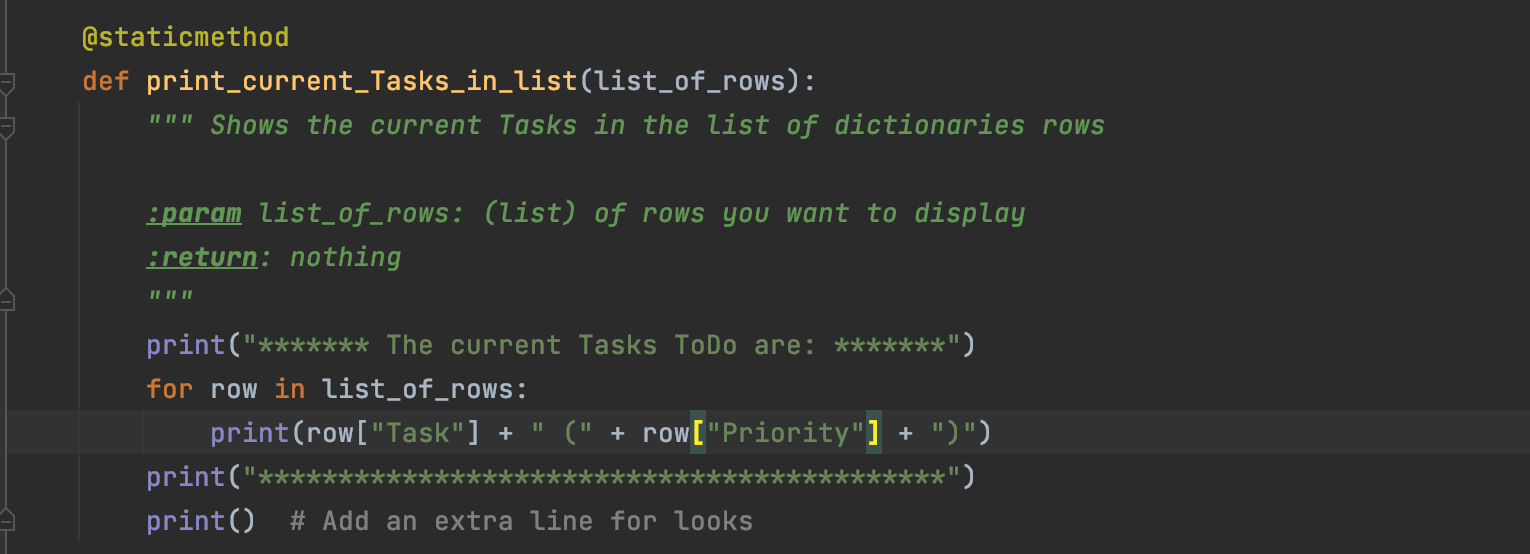
***Figure 5. print\_menu\_Tasks () code***

**Input\_menu\_choice ():** This function is responsible for getting the menu choice from a user using input () and returned that choice from the function. This complete function is shown in Figure 6.

**

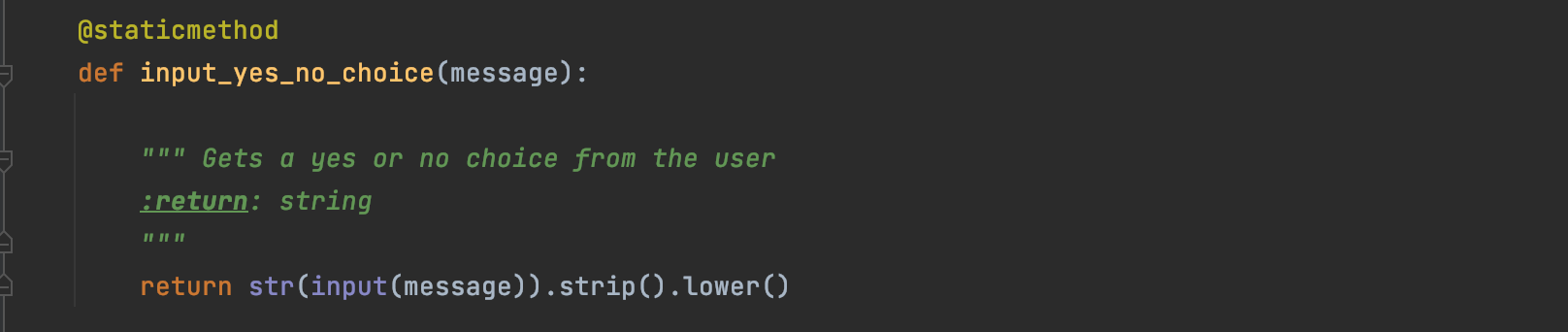
***Figure 6. input\_menu\_choice () code***

**Print\_current\_tasks\_in\_list (list\_of\_rows):** This function is responsible for printing out the current tasks in the list by iterating through every task in the list.The complete function code is shown in Figure 7.

**

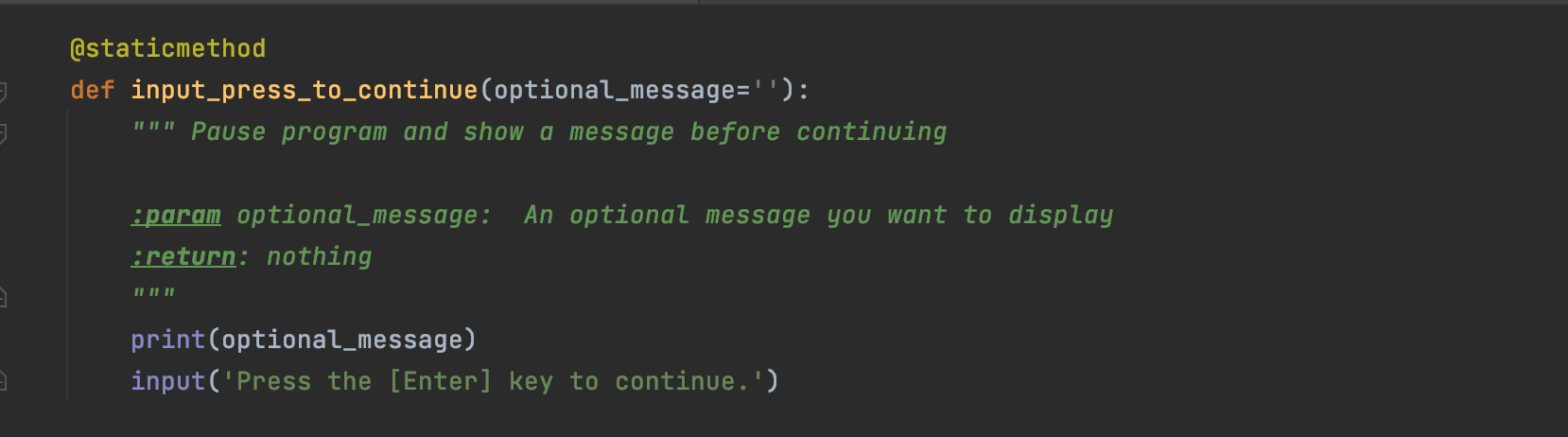
*Figure 7. print\_current\_Tasks\_in\_list() code*

**Input\_yes\_or\_no\_choice (message)**: This function is responsible for taking the user input as “y” or “n” as input user choice and return the choice from the function. The function takes a message as an input parameter. The message can be “Do you want to save the data to the file?”. This function code is shown in Figure 8.

**

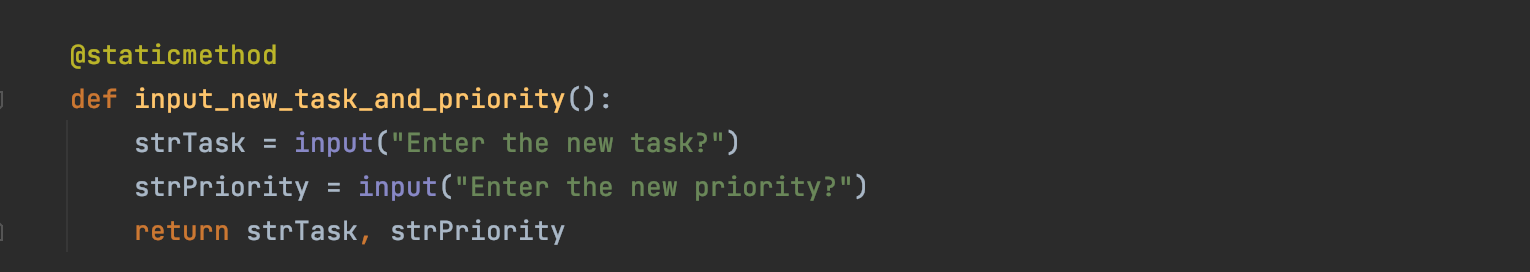
***Figure 8. input\_yes\_no\_choice() code***

**Input\_press\_to\_continue (optional\_message)**: This function is going to take “y/n” as user input if the user wants to continue with the process. The intention is to show the message before continuing with the process. This is shown in Figure 9.



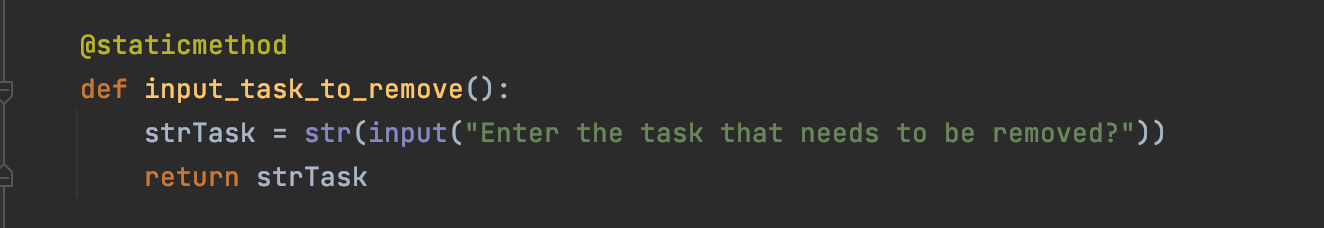
***Figure 9. input\_press\_to\_continue () code***

**Input\_new\_task\_priority (): This function is responsible for taking new task and priority as user input using input (). Finally returning new task and priority as return values from the function. The complete function code is shown in Figure 10.**

****

***Figure 10. input\_new\_task\_and\_priority () code***

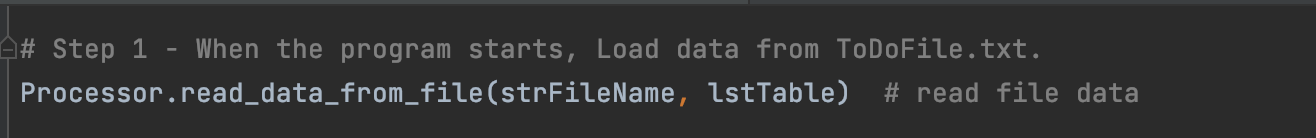
**Input\_task\_to\_remove ():** This function is responsible for taking task that user wants to remove from the list. The complete function is shown in Figure 11.



***Figure 11. input\_task\_to\_remove () code***

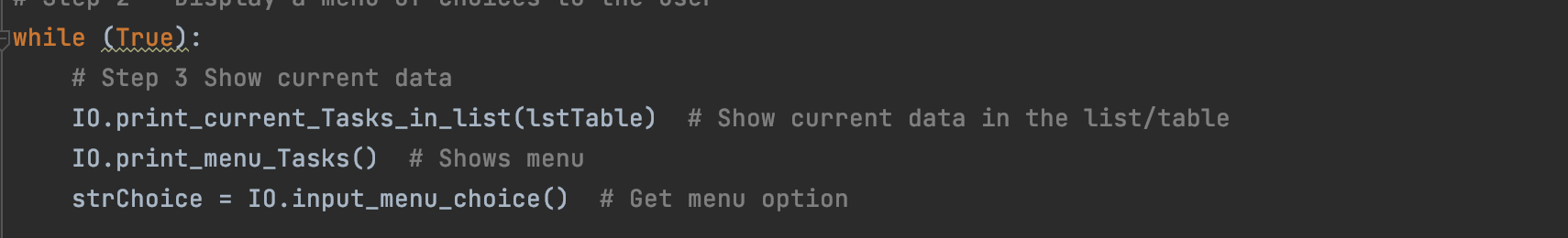
# Description of the Main body:

The first step is to load the data from the existing list. In order to do this, the function read\_data\_from\_file () is called from Processor class. As soon as this function will be called, the logic within the function gets executed. The function invocation is shown in below Figure 12.



***Figure 12. read\_data\_from\_file() invocation***

As a next step, we are showing current tasks in the list and also asking the menu options to be entered from the user by calling current\_Tasks\_in\_list () and print\_menu\_Tasks (). In order to ask user choice from provided menu options, input\_menu\_choice () is called. This is shown in Figure 13.



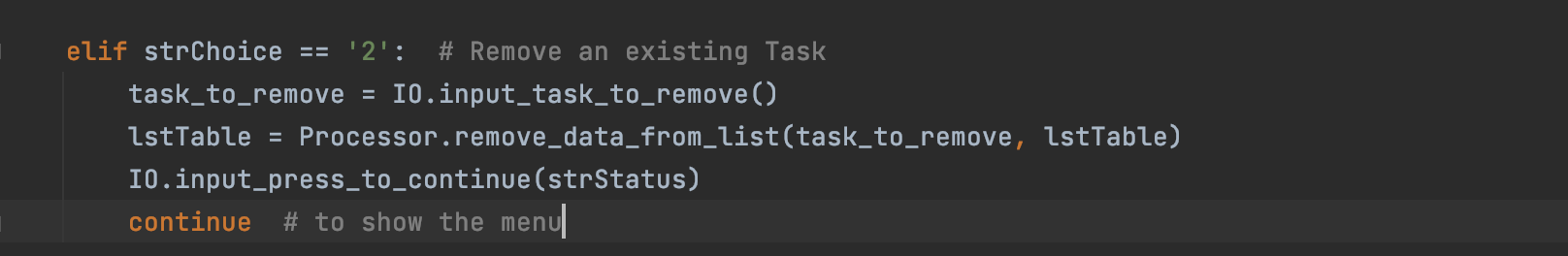
***Figure 13. Various functions invocations***

After showing current tasks in the list, the next step is to call required functions based on the user choice. If user chooses option 1, then first of all, input\_new\_task\_and\_priorty () is called and finally add\_data\_to\_list () is called as shown in Figure 14.



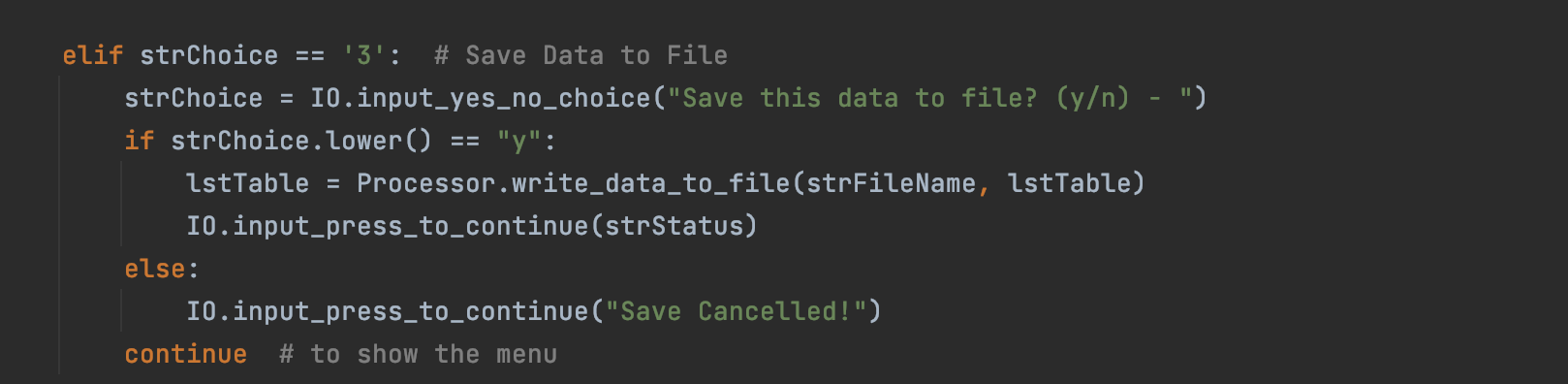
***Figure 14. If user chooses option 1***

If user chooses option 2, then first we will ask user to provide which task needs to be removed from the list and then remove\_data\_from\_list () is called in order to remove the row from the list. This is shown in Figure 15.



***Figure 15. If user chooses option 2***

If user chooses option 3, if the user chooses to save the data to the file, write\_data\_to\_file () is called to write and save the data into the file. If not, then simply display the message “Save cancelled” to the user. This is shown in Figure 16.



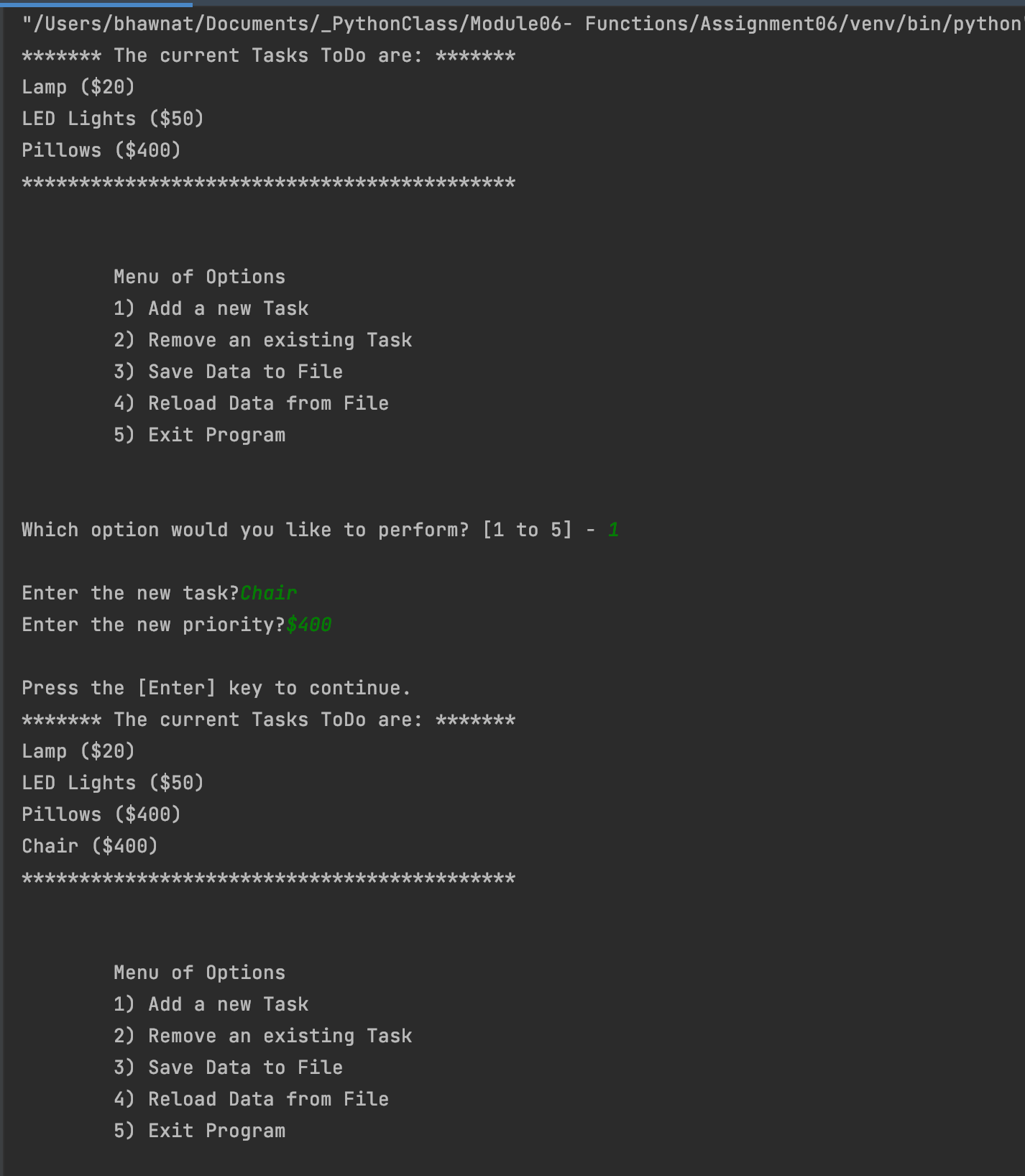
***Figure 16. If user chooses option 3***

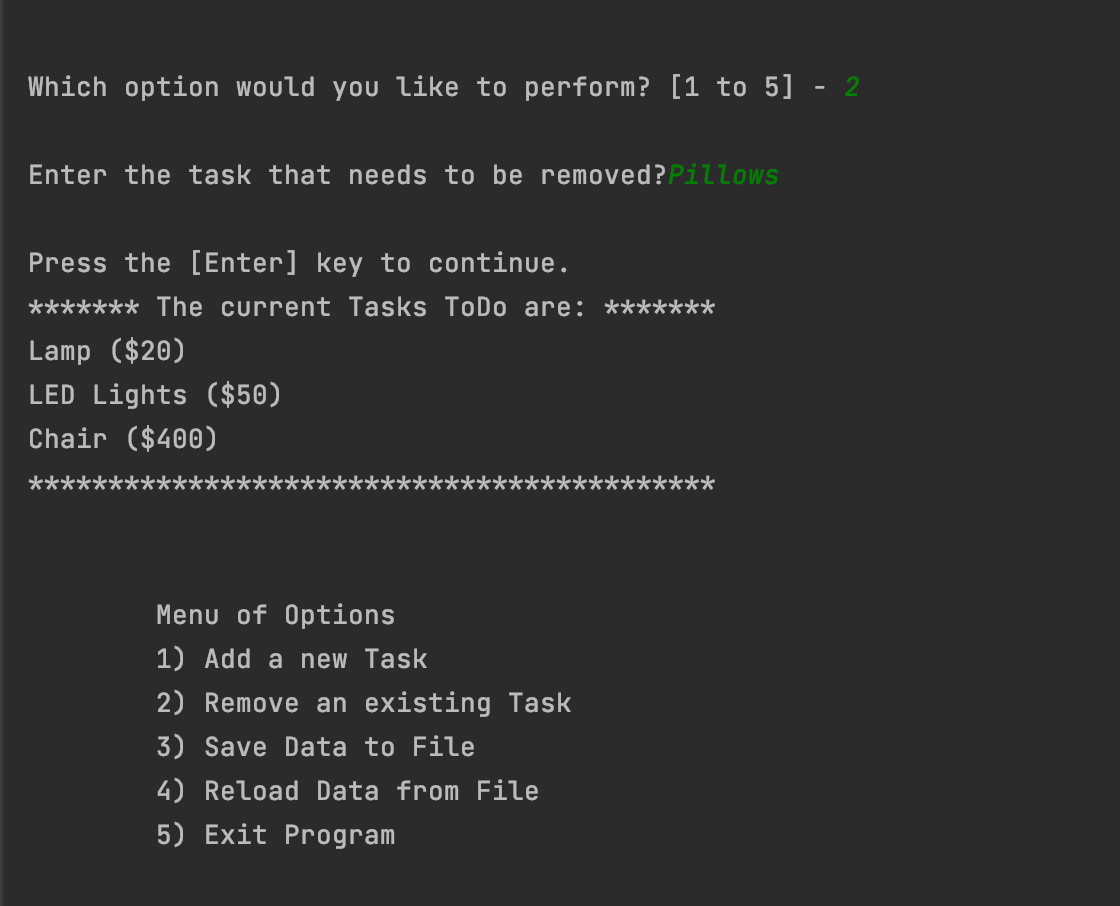
Finally, if user chooses option 4, we simply ask user if they want to reload the data from the file without saving the unsaved data to the file. If the user chooses ‘y’ then we call read\_data\_from\_file () and it will print the data from the existing list without saving new data/tasks in the list. However, if the user chooses another option 5, then this means user wants to exit from the program. Simply display message “Good bye” to the user. This is shown in Figure 17.

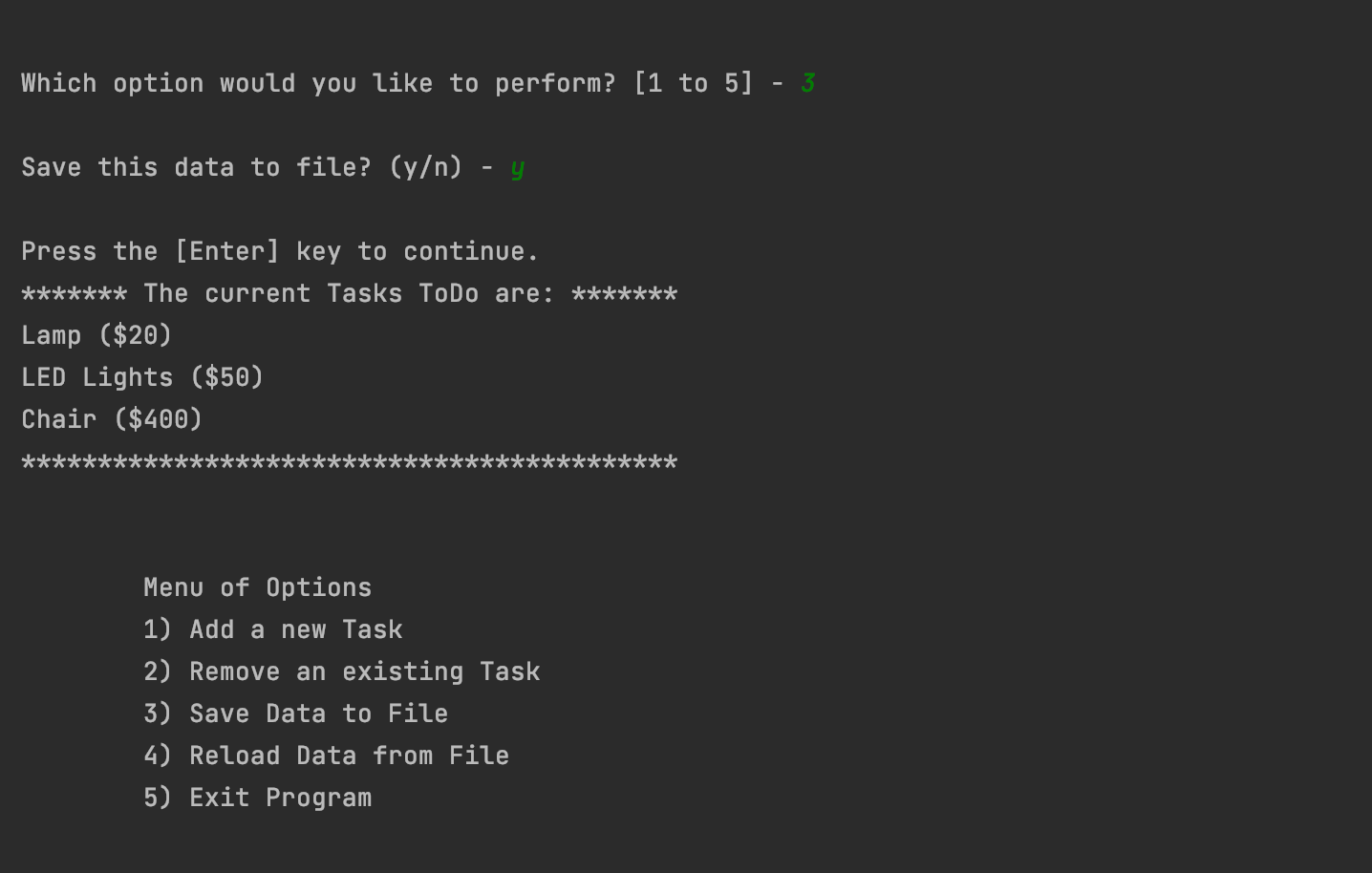


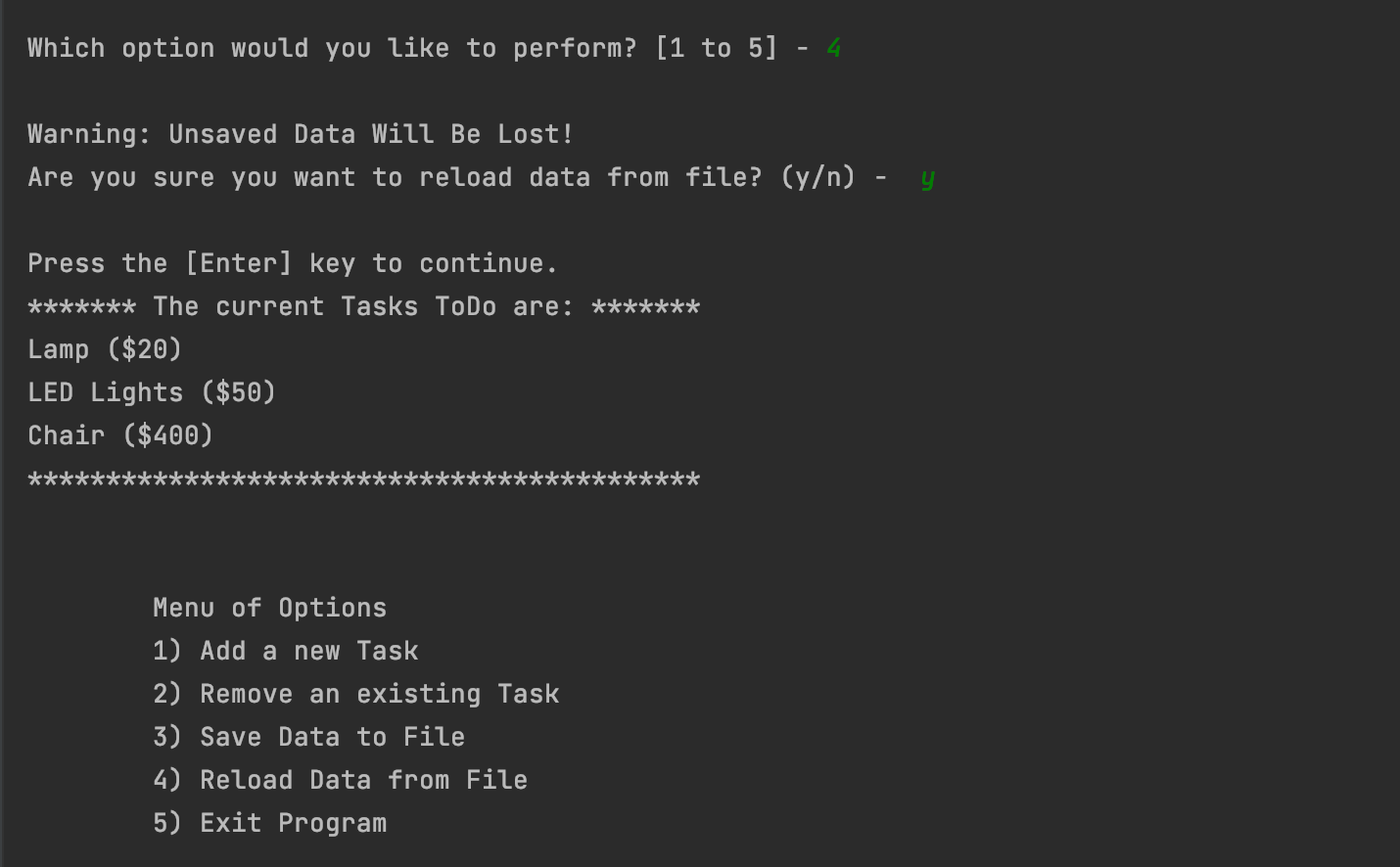
***Figure 17. If user chooses option 4 or 5***

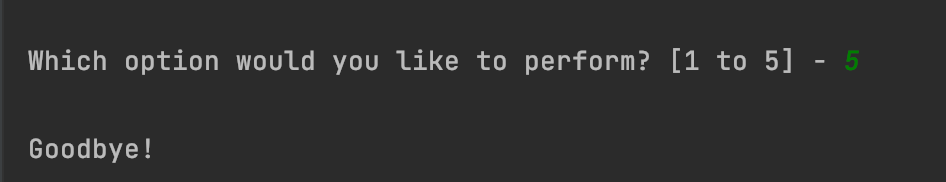
# Execution results in PyCharm/MacOS:

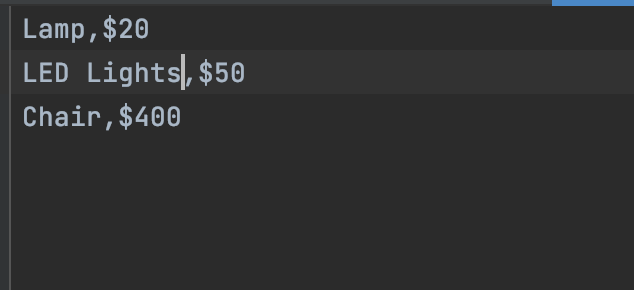




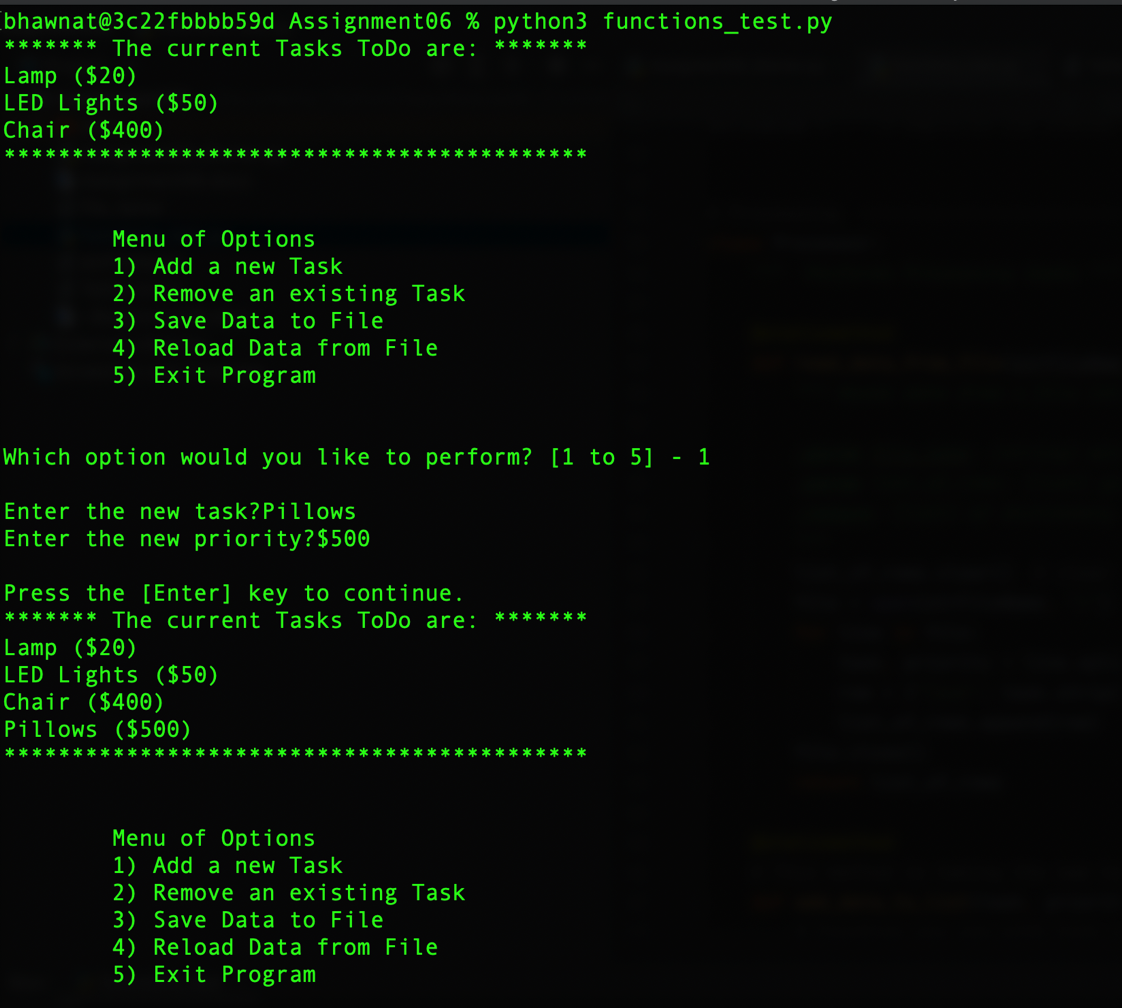




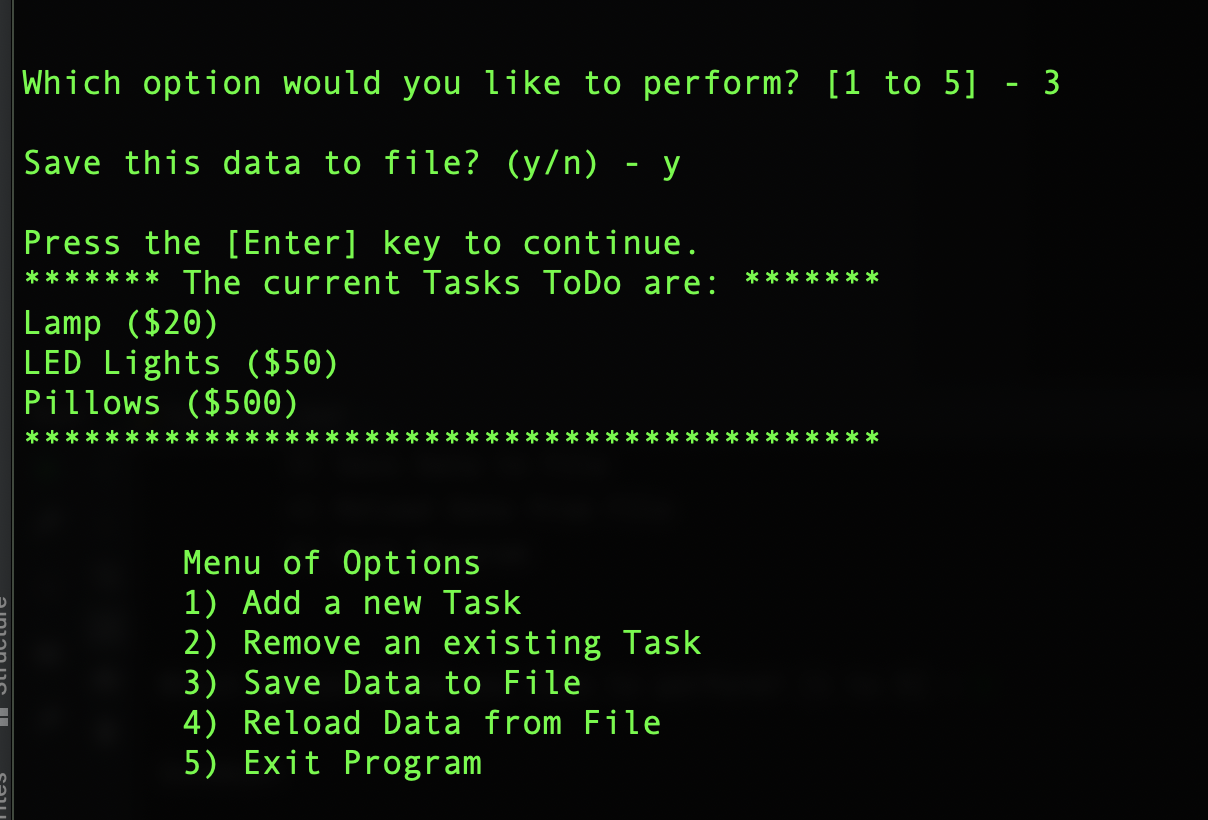


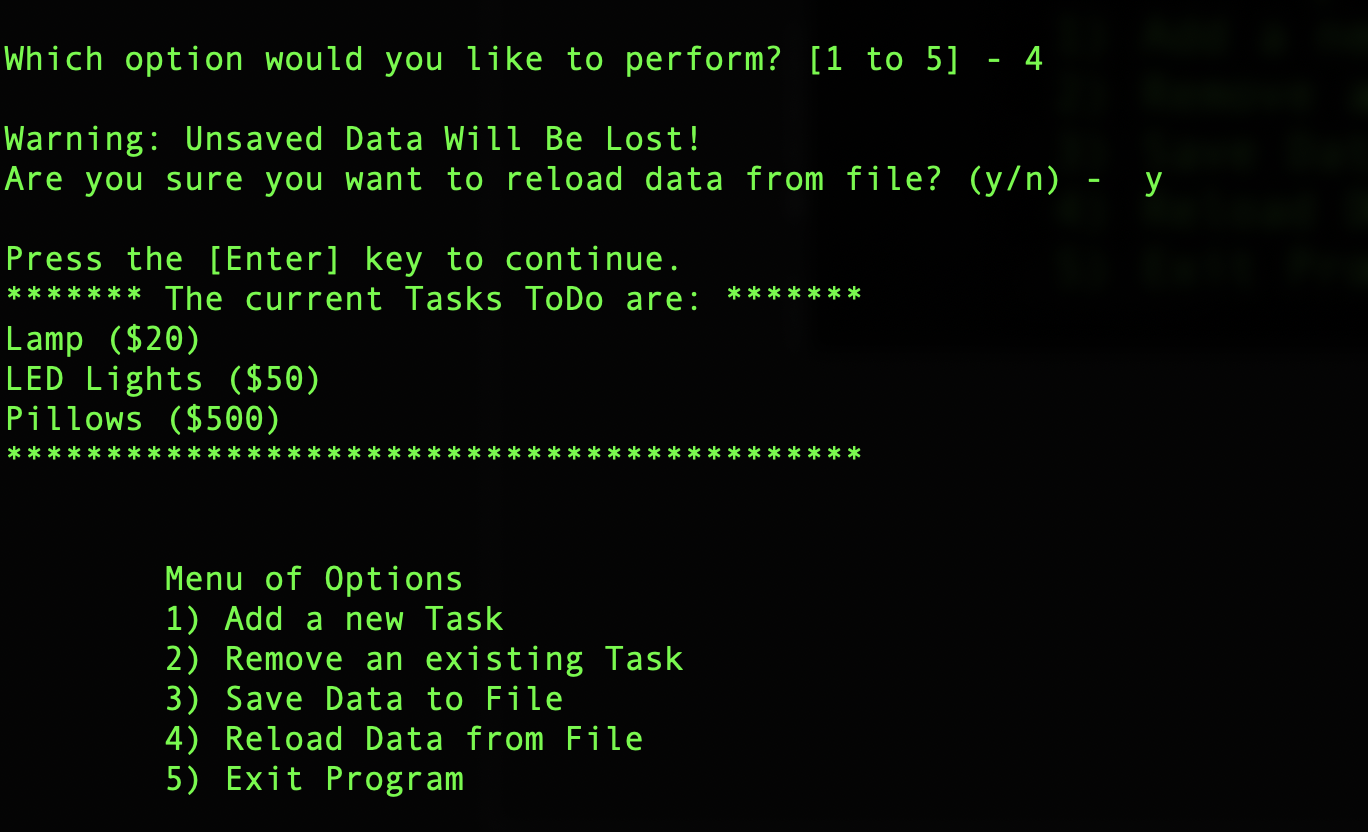


# MacOS:











# Summary:

In this assignment, I had learned about how to use functions and classes in the python script. In addition to this, I learned how to make function invocations from within function. This helps in understanding the separation of concerns within the program.