LOUIS YENCKEN

08/12/2020

IT FDN 110 A

Assignment 06

GitHub Link:

Assignment056\_Starter.py using Classes and Functions

**Objective:**

The program has been created to update the status of a users task and the corresponding priority. We do this through the use of input and the function executing the input.

**Introduction:**

To do this the program as seen in Fig 1 Assignment06 script from PyCharm the script uses a list of menus that are very similar to that of Assignment 05. The difference is now that we are using functions to execute the block of code. It does this by using a while loop which will loop through the menu list until is finds its True value which will then execute the correspodning block of code. After the function has been completed it will return the user back to the menu. This can continue until the user decides to exit the program by selecting the corresponding menu choice which as seen in Fig 1 is selection ‘5’. This will then give the user a message and exit the script. There are five choices that the user can select. These will be outlined in further sections of this assignment.

# ---------------------------------------------------------------------------- #  
# Title: Assignment 06  
# Description: Working with functions in a class,  
# 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  
# RRoot,1.1.2030,Added code to complete assignment 5  
# <LYencken>,<08/11/2020>,Modified code to complete assignment 6  
# ---------------------------------------------------------------------------- #  
  
# Data ---------------------------------------------------------------------- #  
# Declare variables and constants  
strFileName = "ToDoFile.txt" # The name of the data file  
objFile = None # An object that represents a file  
dicRow = {} # A row of data separated into elements of a dictionary {Task,Priority}  
lstTable = [] # A list that acts as a 'table' of rows  
strChoice = "" # Captures the user option selection  
strTask = "" # Captures the user task data  
strPriority = "" # Captures the user priority data  
strStatus = "" # Captures the status of an processing functions  
  
  
# Processing --------------------------------------------------------------- #  
class Processor:  
 *""" Performs Processing tasks """* @staticmethod  
 def read\_data\_from\_file(file\_name**,** list\_of\_rows):  
 *""" Reads data from a file into a list of dictionary rows* ***:param*** *file\_name: (string) with name of file:* ***:param*** *list\_of\_rows: (list) you want filled with file data:* ***:return****: (list) of dictionary rows  
 """* list\_of\_rows.clear() #clear current data  
  
  
  
 file = open(file\_name**,** 'r')  
 for line in file:  
 task**,** priority = line.split(",")  
 row = {"Task": task.strip()**,** "Priority": priority.strip()}  
 list\_of\_rows.append(row)  
 file.close()  
 return list\_of\_rows**,** 'Success'  
  
 @staticmethod  
 def add\_data\_to\_list(task**,** priority**,** list\_of\_rows):  
 *""" Adds task from input to the list of dictionary rows* ***:param*** *task: (string) with name of task:* ***:param*** *priority: (string) with priority of task:* ***:param*** *list\_of\_rows: (list) you want filled with additional task:* ***:return****: (list) of dictionary rows  
 """* dicRow = {"Task": task.strip()**,** "Priority": priority.strip()}  
 list\_of\_rows.append(dicRow)  
 return list\_of\_rows**,** 'Success'  
  
 @staticmethod  
 def remove\_data\_from\_list(task**,** list\_of\_rows):  
 *""" Removes task from input from the list of dictionary rows* ***:param*** *task: (string) with name of task:* ***:param*** *list\_of\_rows: (list) you want task removed from:* ***:return****: (list) of dictionary rows  
 """* bool\_list = False  
 for dicRow in list\_of\_rows:  
 if dicRow["Task"] == task:  
 list\_of\_rows.remove(dicRow)  
 bool\_list = True  
 if bool\_list == False:  
 print("cannot find task")  
 return list\_of\_rows**,** 'Success'  
  
 @staticmethod  
 def write\_data\_to\_file(file\_name**,** list\_of\_rows):  
 *""" Writes data from list of dictionary rows to file* ***:param*** *file\_name: (string) with name of file:* ***:param*** *list\_of\_rows: (list) you want added to text file:* ***:return****: (list) of dictionary rows  
 """* objFile = open(file\_name**,** 'w')  
 for line in list\_of\_rows:  
 objFile.write(line["Task"] + ',' + line["Priority"] + '\n')  
 objFile.close()  
 return list\_of\_rows**,** 'Success'  
  
  
# Presentation (Input/Output) -------------------------------------------- #  
class IO:  
 *""" Performs Input and Output tasks """* @staticmethod  
 def print\_menu\_Tasks():  
 *""" Display a menu of choices to the user* ***:return****: nothing  
 """* print('''  
 Menu of Options  
 1) Add a new Task  
 2) Remove an existing Task  
 3) Save Data to File   
 4) Reload Data from File  
 5) Exit Program  
 ''')  
 print() # Add an extra line for looks  
  
 @staticmethod  
 def input\_menu\_choice():  
 *""" Gets the menu choice from a user* ***:return****: string  
 """* choice = str(input("Which option would you like to perform? [1 to 5] - ")).strip()  
 print() # Add an extra line for looks  
 return choice  
  
 @staticmethod  
 def print\_current\_Tasks\_in\_list(list\_of\_rows):  
 *""" Shows the current Tasks in the list of dictionaries rows* ***:param*** *list\_of\_rows: (list) of rows you want to display* ***:return****: nothing  
 """* print("\*\*\*\*\*\*\* The current Tasks ToDo are: \*\*\*\*\*\*\*")  
 for dicRow in list\_of\_rows:  
 print(dicRow["Task"] + " (" + dicRow["Priority"] + ")")  
 print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")  
 print() # Add an extra line for looks  
  
 @staticmethod  
 def input\_yes\_no\_choice(message):  
 *""" Gets a yes or no choice from the user* ***:return****: string  
 """* return str(input(message)).strip().lower()  
  
 @staticmethod  
 def input\_press\_to\_continue(optional\_message=''):  
 *""" Pause program and show a message before continuing* ***:param*** *optional\_message: An optional message you want to display* ***:return****: nothing  
 """* print(optional\_message)  
 input('Press the [Enter] key to continue.')  
  
 @staticmethod  
 def input\_new\_task\_and\_priority():  
  
 task = str(input("What task is to be added: ")).lower().strip()  
 priority = str(input("What is the priority: ")).lower().strip()  
 return task**,** priority  
  
 @staticmethod  
 def input\_task\_to\_remove():  
  
 task = str(input("What task do you want removed: ")).lower().strip()  
 return task  
  
  
# Main Body of Script ------------------------------------------------------ #  
  
# Step 1 - When the program starts, Load data from ToDoFile.txt.  
Processor.read\_data\_from\_file(strFileName**,** lstTable) # read file data  
  
# Step 2 - Display a menu of choices to the user  
while (True):  
 # Step 3 Show current data  
 IO.print\_current\_Tasks\_in\_list(lstTable) # Show current data in the list/table  
 IO.print\_menu\_Tasks() # Shows menu  
 strChoice = IO.input\_menu\_choice() # Get menu option  
  
 # Step 4 - Process user's menu choice  
 if strChoice.strip() == '1': # Add a new Task  
 task**,** priority = IO.input\_new\_task\_and\_priority()  
 lstTable**,** strStatus = Processor.add\_data\_to\_list(task**,** priority**,** lstTable)  
 IO.input\_press\_to\_continue(strStatus)  
 continue # to show the menu  
  
 elif strChoice == '2': # Remove an existing Task  
 task = IO.input\_task\_to\_remove()  
 lstTable**,** strStatus = Processor.remove\_data\_from\_list(task**,** lstTable)  
 IO.input\_press\_to\_continue(strStatus)  
 continue # to show the menu  
  
 elif strChoice == '3': # Save Data to File  
 strChoice = IO.input\_yes\_no\_choice("Save this data to file? (y/n) - ")  
 if strChoice.lower() == "y":  
 lstTable**,** strStatus = Processor.write\_data\_to\_file(strFileName**,** lstTable)  
 IO.input\_press\_to\_continue(strStatus)  
 else:  
 IO.input\_press\_to\_continue("Save Cancelled!")  
 continue # to show the menu  
  
 elif strChoice == '4': # Reload Data from File  
 print("Warning: Unsaved Data Will Be Lost!")  
 strChoice = IO.input\_yes\_no\_choice("Is this really a good idea? (y/n) - ")  
 if strChoice.lower() == 'y':  
 lstTable**,** strStatus = Processor.read\_data\_from\_file(strFileName**,** lstTable)  
 IO.input\_press\_to\_continue(strStatus)  
 else:  
 IO.input\_press\_to\_continue("File Reload Cancelled!")  
 continue # to show the menu  
  
 elif strChoice == '5': # Exit Program  
 print("Adios!")  
 exit() # and Exit

**Fig 1: Script in PyChrm**

**Menu Selection and body of script:**

The first selection of menu choice as seen in Fig 2: under menu options is ‘Add a new Task’. This is done by requesting input from the user. In this case the user must input all task in lower case. This is to ensure that there in consistency for when/if a task needs to be removed. Once data has been input the script will then append the text file which has been opened in read at the beginning of the script under ‘processing’. Note the list will only have the items placed inside the text file once the third option under selection ‘3’ has been selected. This will write user input into the ToDoList.txt file.

**Printing Current task**

As seen in Fig 2 two items have been added. These are ‘eat’ which has a low priority and ‘sleep’ which has a high priority. This is then displayed to the user. This display is done by using a function that reads the current data that has been placed into the text file which will then also use ‘Processor.read\_data\_from\_file(strFileName**,** lstTable’. This as its name implies reads data from a file which is under ‘’strFileName and ‘lstTable’. It will then print the file contents and display it to the user. This ensures that the script is more user friendly for when a task needs to be removed as shown in Fig 2.

C:\\_PythonClass\Assignment06\venv\Scripts\python.exe C:/\_PythonClass/Assignment06/Assignment06\_Starter.py

\*\*\*\*\*\*\* The current Tasks ToDo are: \*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Menu of Options

1) Add a new Task

2) Remove an existing Task

3) Save Data to File

4) Reload Data from File

5) Exit Program

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

What task is to be added: eat

What is the priority: low

Success

Press the [Enter] key to continue.

\*\*\*\*\*\*\* The current Tasks ToDo are: \*\*\*\*\*\*\*

eat (low)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Menu of Options

1) Add a new Task

2) Remove an existing Task

3) Save Data to File

4) Reload Data from File

5) Exit Program

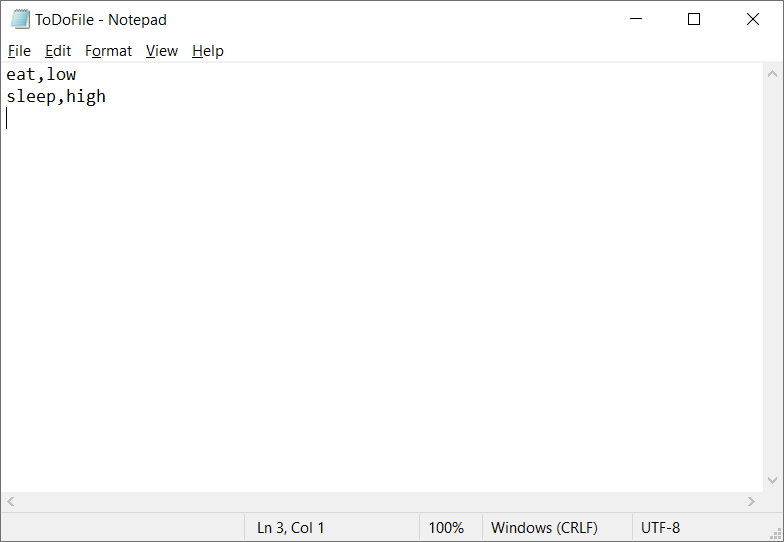
Which option would you like to perform? [1 to 5] - 1

What task is to be added: sleep

What is the priority: high

Success

Press the [Enter] key to continue.

\*\*\*\*\*\*\* The current Tasks ToDo are: \*\*\*\*\*\*\*

eat (low)

sleep (high)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Menu of Options

1) Add a new Task

2) Remove an existing Task

3) Save Data to File

4) Reload Data from File

5) Exit Program

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

Save this data to file? (y/n) - y

Success

Press the [Enter] key to continue.

\*\*\*\*\*\*\* The current Tasks ToDo are: \*\*\*\*\*\*\*

eat (low)

sleep (high)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Menu of Options

1) Add a new Task

2) Remove an existing Task

3) Save Data to File

4) Reload Data from File

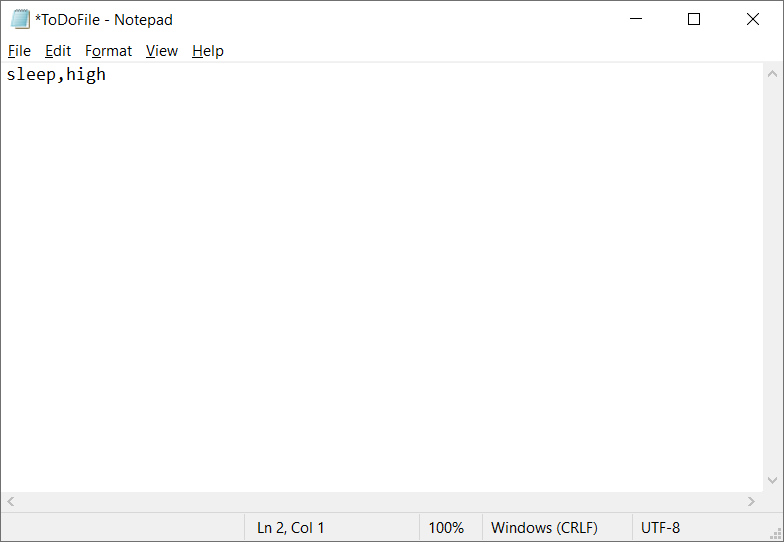
5) Exit Program

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

What task do you want removed: eat

Success

Press the [Enter] key to continue.

\*\*\*\*\*\*\* The current Tasks ToDo are: \*\*\*\*\*\*\*

sleep (high)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Menu of Options

1) Add a new Task

2) Remove an existing Task

3) Save Data to File

4) Reload Data from File

5) Exit Program

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

Save this data to file? (y/n) - y

Success

Press the [Enter] key to continue.

\*\*\*\*\*\*\* The current Tasks ToDo are: \*\*\*\*\*\*\*

sleep (high)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Menu of Options

1) Add a new Task

2) Remove an existing Task

3) Save Data to File

4) Reload Data from File

5) Exit Program

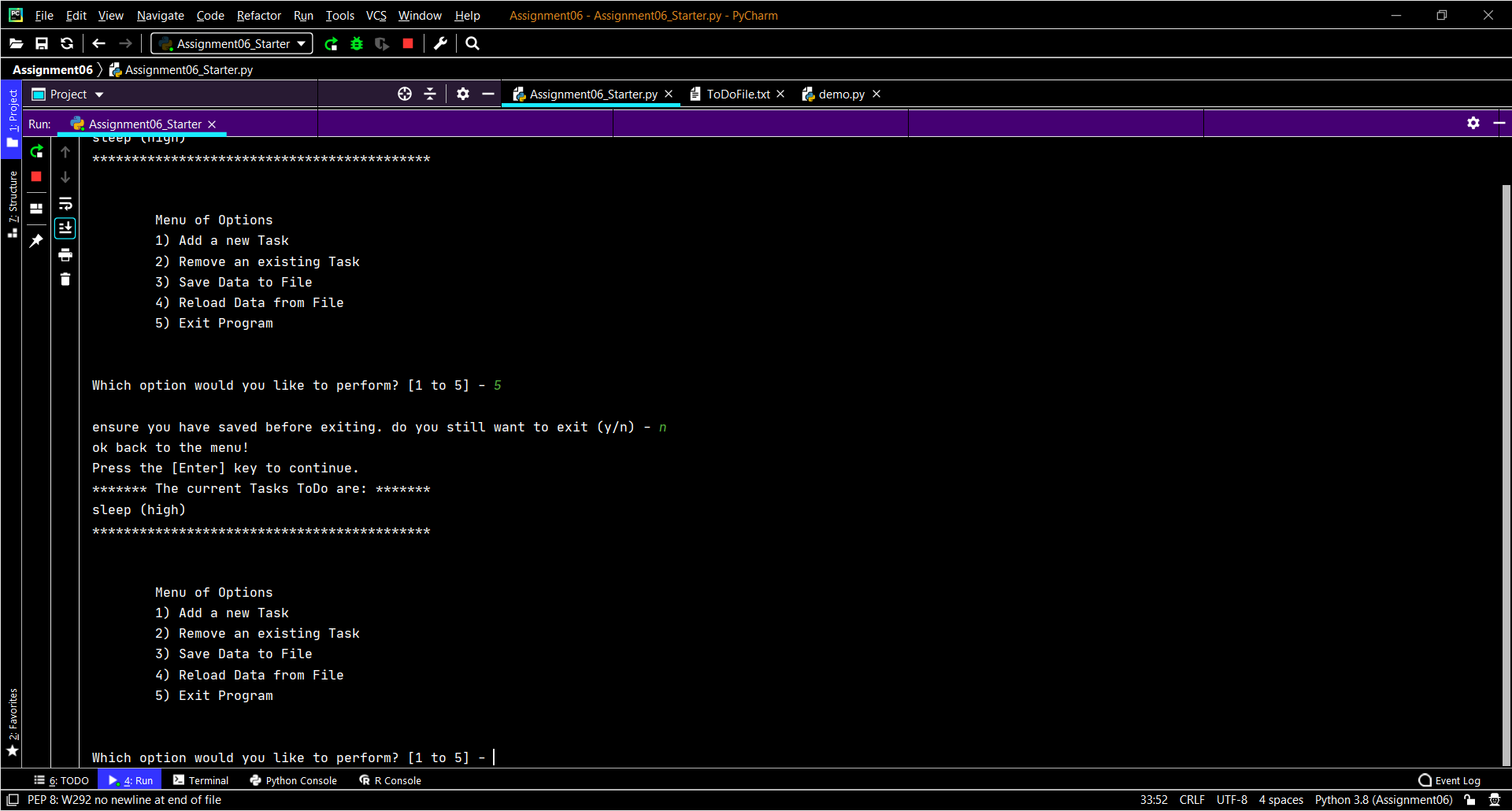
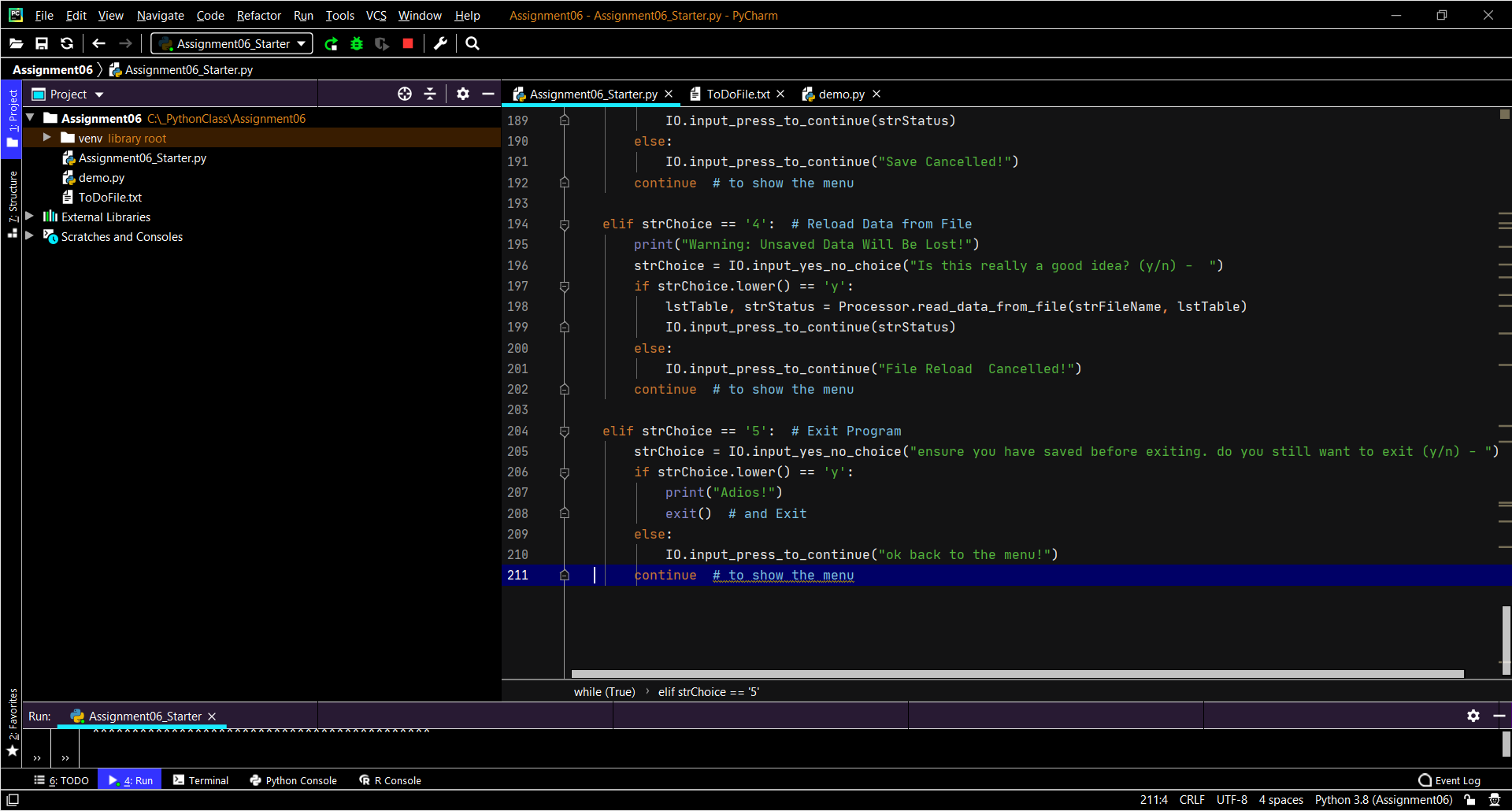
Which option would you like to perform? [1 to 5] -

**Fig 2: Running script in PyCharm, adding and removing items from list.**

**Removing Data from the list and saving to File.**

Sometimes task are no longer needed. For example, the user decided that food was no longer a task that was required. So, the user has chosen menu choice ‘2’ which will then ask for the user input and as ‘to type in a task that needs to be removed. The user has input ‘eat’ in lowercase as the script requires that the task matches a input that has been made into the dictionary under ‘dicRow’. If a entry has been found in the list then syntax ‘.remove’ is used and the task (and corresponding priority) is removed from the list. Once this has been done the script will then return the current items in the list to the user and the menu will be displayed again waiting for user input.

Once the item has been removed in the script the user must ‘save’ the list for the item to also be removed from the list. This is because the block of code under the save function will then write over the text file and remove the task that the user has requested to delete. This is important step as the text file will not correspond to the users request from the script. So whilst the script will be displaying only the current items that the user wants the text file will not reflect that. If the user request to exit without saving, then the user data is lost. This reminder is printed to the user before exiting the script to try and lessen human error…



**Fig 3: Please don’t leave! (exit choice and corresponding code)**

**Exit from the script and reloading previous save file:**

As seen in Fig 3 the code that was used has been poached from the ‘Reload’ input choice. Instead of then executing the save file function the script then will then instead either print ‘Adios’ and exit or else print ‘back to the menu’ and continue back to the menu. As stated then reason for this is just in case the user has forgotten to save the data or accidently input menu choice ‘5’ and exits from the script. As stated the ‘exit’ block of code was mimicked from the ‘reload’ block of code as shown in Fig 4. This block of code will reload the most recent file save to the script and display this to the user. If this menu choice is selected and the user chooses ‘y’ (for yes) then the current data that has been input will be deleted and not save and hence be lost. This is shown in Fig 4 as well.

C:\\_PythonClass\Assignment06\venv\Scripts\python.exe C:/\_PythonClass/Assignment06/Assignment06\_Starter.py

\*\*\*\*\*\*\* The current Tasks ToDo are: \*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Menu of Options

1) Add a new Task

2) Remove an existing Task

3) Save Data to File

4) Reload Data from File

5) Exit Program

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

What task is to be added: sleep

What is the priority: high

Success

Press the [Enter] key to continue.

\*\*\*\*\*\*\* The current Tasks ToDo are: \*\*\*\*\*\*\*

eat (low)

sleep (high)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Menu of Options

1) Add a new Task

2) Remove an existing Task

3) Save Data to File

4) Reload Data from File

5) Exit Program

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

Save this data to file? (y/n) - y

Success

Press the [Enter] key to continue.

\*\*\*\*\*\*\* The current Tasks ToDo are: \*\*\*\*\*\*\*

eat (low)

sleep (high)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Menu of Options

1) Add a new Task

2) Remove an existing Task

3) Save Data to File

4) Reload Data from File

5) Exit Program

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

What task is to be added: watch Tv

What is the priority: extremly low

Success

Press the [Enter] key to continue.

\*\*\*\*\*\*\* The current Tasks ToDo are: \*\*\*\*\*\*\*

eat (low)

sleep (high)

watch tv (extremly low)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Menu of Options

1) Add a new Task

2) Remove an existing Task

3) Save Data to File

4) Reload Data from File

5) Exit Program

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

What task is to be added: build a house

What is the priority: medium

Success

Press the [Enter] key to continue.

\*\*\*\*\*\*\* The current Tasks ToDo are: \*\*\*\*\*\*\*

eat (low)

sleep (high)

watch tv (extremly low)

build a house (medium)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Menu of Options

1) Add a new Task

2) Remove an existing Task

3) Save Data to File

4) Reload Data from File

5) Exit Program

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

Warning: Unsaved Data Will Be Lost!

Is this really a good idea? (y/n) - y

Success

Press the [Enter] key to continue.

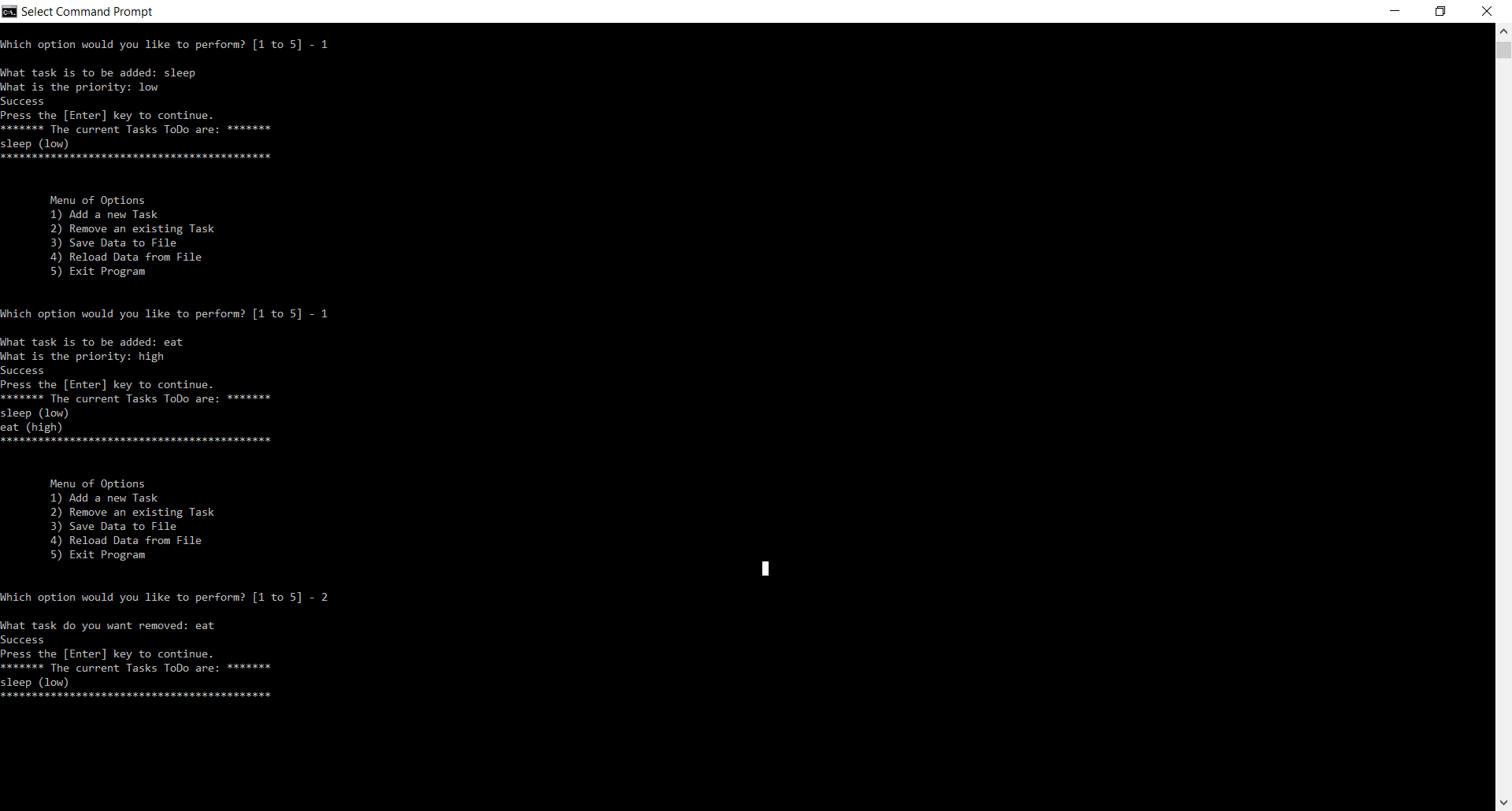
\*\*\*\*\*\*\* The current Tasks ToDo are: \*\*\*\*\*\*\*

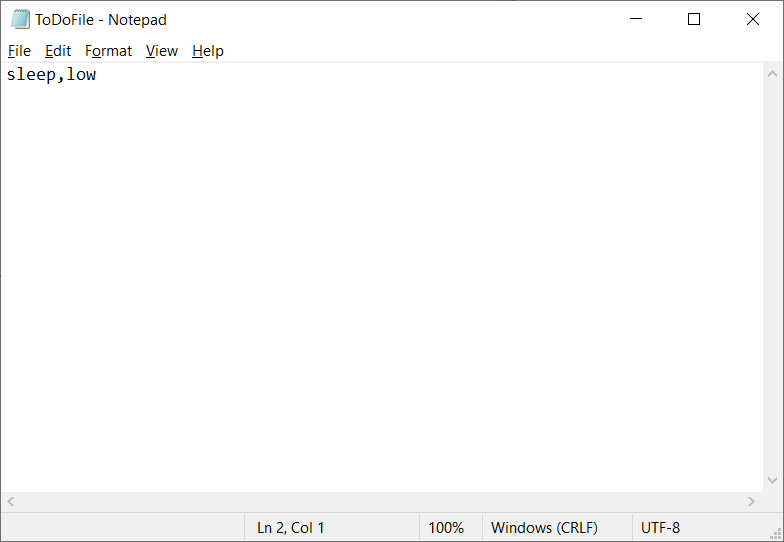
eat (low)

sleep (high)

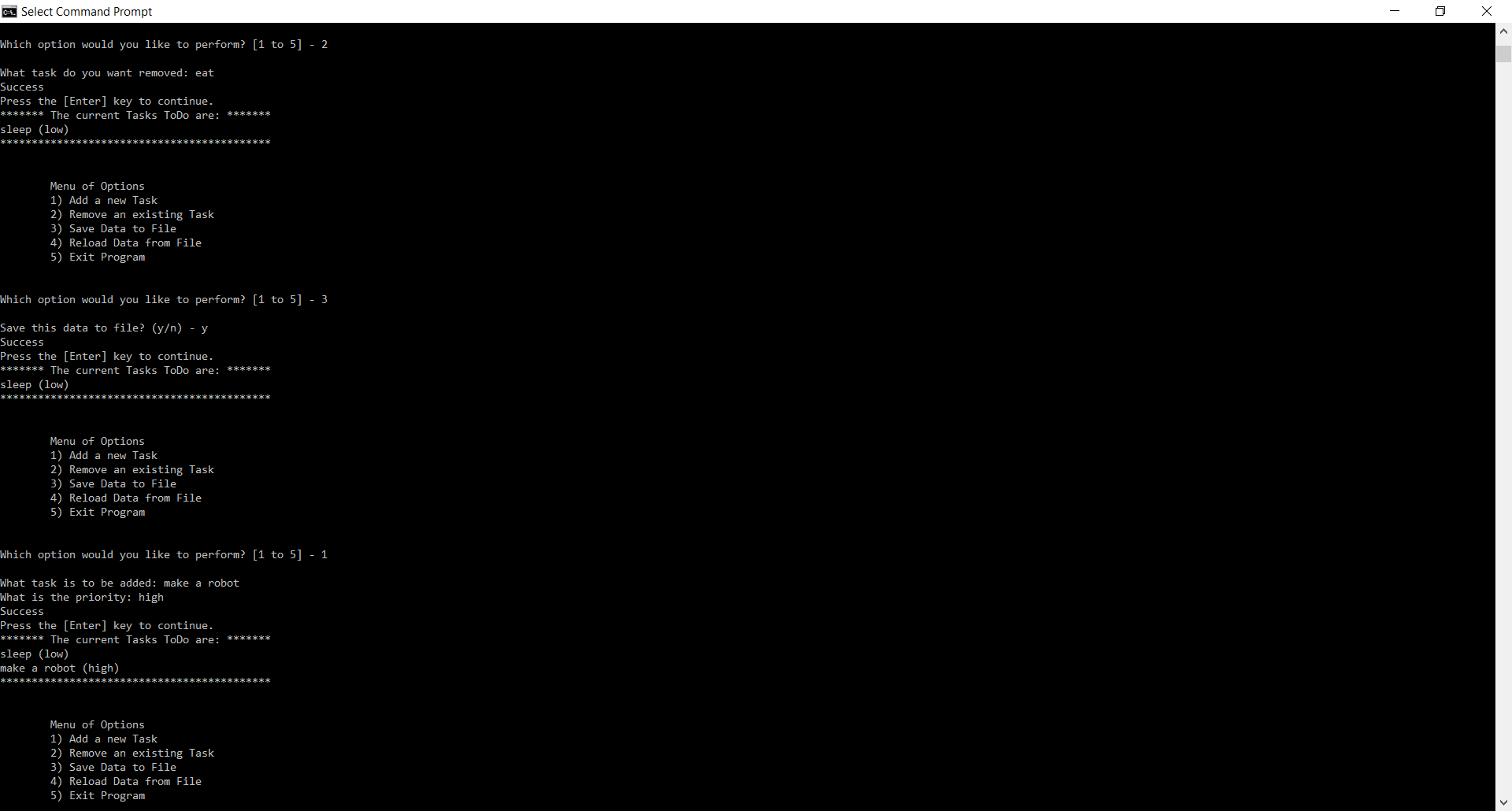
**Fig 4: Using Reload menu choice**

**Running script in command prompt!**

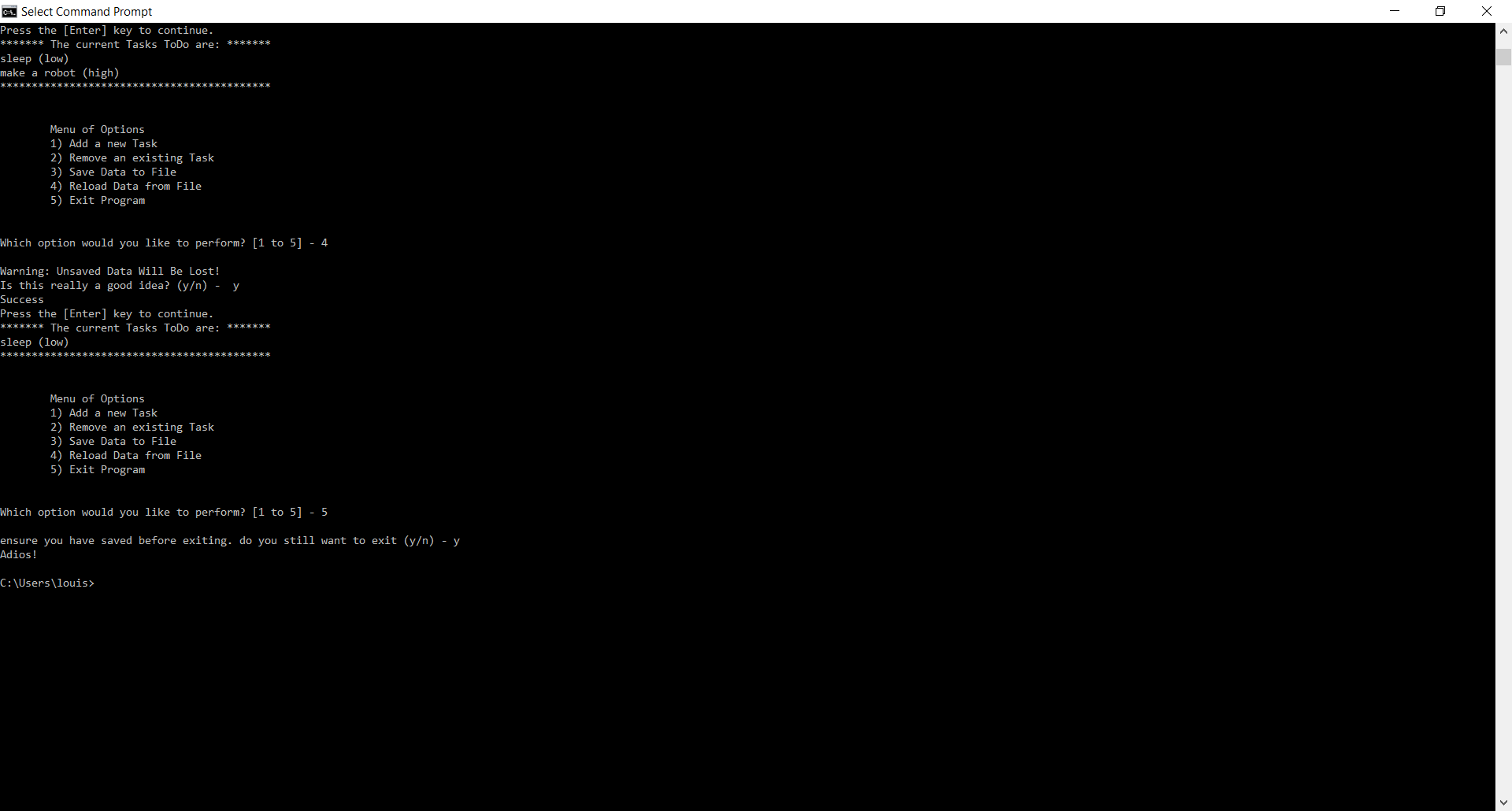




**Fig 5 Adding task and removing task in command prompt and the corresponding list once saved**



**Fig 6: Saving current data and inputting new task**



**Fig 7: Reloading previous save and exiting command prompt.**

**Summary:**

As seen through Fig 5-7 the script ran through command prompt seamlessly. Overall the script has functioned well and the only time there is a issue with the script it is caused by human error such as accidently placing a letter into the body of the script accidently and not realising… overall this use of classes and function shows that a script can execute different blocks of code as required through the use of ‘def’. These are useful for instead having to continuously write out large blocks of code that use the same function or class. This also ensures that there is less traceback errors and cuts down on the amount of lines needed drastically. Another useful tool that made this assignment in some regards easier was the use of a ‘demo’ script. I used this to be able to isolate my code and ensure that it functioned well. This was useful for that fact that there is a large amount of code for this script. This meant that rather then having to scroll up and down the script trying to find the issue creating the traceback or ‘#’ out large blocks of code I could simply isolate it separately.