Sanghee Kang
February 22, 2022
IT FND 110 Foundations of Programming: Python
Assignment 06
GitHub URL: https://github.com/bsb218218/IntroToProg-Python-Mod06

Modifying the To Do List Python Script

1. Introduction

In this report, I will explain how I modified the To Do List Python script by adding functions. Also, I will describe my observation of the performance of the code in both PyCharm and the command shell.

2. Modifying the To Do List Python Script

2.1. Creating a sub-folder

Like I did for the previous assignment, I created a folder (Assignment06_SangheeKang) inside of the "_PythonClass" folder.

2.2. Creating a new Project in PyCharm

I created a new Project in Pycharm (File-New Project). For the location, I used "C:_PythonClass\Assignment06_SangheeKang" and selected "New environment using Virtualenv".

2.3. Adding the starter Python file (Assignment06_Starter.py) and modifying the script

In the new project, I added the starter Python file and renamed it as "Assignment06_SangheeKang". In the script header, I added my name and the date to indicate that I modified the file.

2.3.1. Data section

As the instructor provided a starting template which contained some functions and codes in it, I did not have to write the code from scratch. To complete the assignment, I first tried to study what is already added in the file, which was written by someone else. It took a while to make sense of how things are organized in the script. In particular, as I have not seen many codes that are organized using functions yet, it was a challenging task. As for the data section, I kept what was provided in the starting script and only changed my text file name (existing file) to ToDoFile.txt" for consistency.

```
# ------------------------------------------------------------------------ #
# 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
# SKang,2.22.2022,Modified code to complete assignment 06
# ------------------------------------------------------------------------ #

# Data ------------------------------------------------------------------- #
# Declare variables and constants
file_name_str = "ToDoFile.txt"  # The name of the data file
file_obj = None  # An object that represents a file
row_dic = {}  # A row of data separated into elements of a dictionary {Task,Priority}
table_lst = []  # A list that acts as a 'table' of rows
choice_str = ""  # Captures the user option selection
```

Figure 1. A screenshot of the content of the script header and the data section

2.3.2. Processing section

In the processing section, three codes/functions need to be added: read_data_from_file(file_name, list_of_rows); add_data_to_list(task, priority, list_of_rows); remove_data_from_list(task, list_of_rows). To figure out what part of the code I should take and put from my assignment05 code, I reviewed other functions in the starting template thoroughly. To separate codes into sections (processing and presentation; as the data section was already created, I focused on the two sections for this assignment), I had to figure out which codes belong to which section. Codes for displaying data (e.g., menu of options, current data) and capturing user's input (e.g., new tasks and their priority, tasks to be removed) were placed in the presentation section and codes for processing data were all located in the processing section. So, for the three functions (read_data_from_file(file_name, list_of_rows); add_data_to_list(task, priority, list_of_rows); remove_data_from_list(task, list_of_rows)) in the processing section, I only added codes used to process data. But, first, as my Assignment05 codes had a couple of issues, I had to fix them in advance. I added relevant codes to each function as shown in Figures 2, 3, and 4. For adding a new task, I used the append() method. For removing data from the list, I added the for loop and the if function, and the remove() method. Lastly, For writing data to a file, I used functions such as open(), file.write(), and file.close().

```python
@staticmethod
def add_data_to_list(task, priority, list_of_rows):
    """ Adds data to a list of dictionary rows

    :param task: (string) with name of task:
    :param priority: (string) with name of priority:
    :param list_of_rows: (list) you want filled with file data:
    :return: (list) of dictionary rows
    """
    row = {"Task": str(task).strip(), "Priority": str(priority).strip()}
    # TODO: Add Code Here!
    list_of_rows.append(row)
    return list_of_rows
```

Figure 2. A function of add_data_to_list in PyCharm

```python
@staticmethod
def remove_data_from_list(task, list_of_rows):
    """ Removes data from a list of dictionary rows

    :param task: (string) with name of task:
    :param list_of_rows: (list) you want filled with file data:
    :return: (list) of dictionary rows
    """
    # TODO: Add Code Here!
    for row in list_of_rows:
        if row["Task"].lower() == task.lower():
            list_of_rows.remove(row)
    return list_of_rows
```

Figure 3. A function of remove_data_from_list in PyCharm

```python
@staticmethod
def write_data_to_file(file_name, list_of_rows):
    """ Writes data from a list of dictionary rows to a File

    :param file_name: (string) with name of file:
    :param list_of_rows: (list) you want filled with file data:
    :return: (list) of dictionary rows
    """
    # TODO: Add Code Here!
    file = open(file_name, "w")
    for row in list_of_rows:
        file.write(str(row["Task"]) + "," + str(row["Priority"]) + "\n")
    file.close()
    return list_of_rows
```

Figure 4. A function of write_data_to_file in PyCharm

2.3.3. Presentation section

For the presentation section, because functions for output_menu_tasks(), input_menu_choice(), output_current_tasks_in_list() were already added in the script, I only added codes for two functions: input_new_task_and_priority() and input_task_to_remove(). Figures 5 and 6 show what codes I added to the script file. For capturing what task (along with its priority) the user wants to add, I used print() for the directions and assigned string variables by using str(), input(), and strip(). For getting tasks that the user wants to remove, I used the same methods as input_new_task_and_priority().

```
@staticmethod
def input_new_task_and_priority():
    """  Gets task and priority values to be added to the list

    :return: (string, string) with task and priority
    """
    #pass  # TODO: Add Code Here!
    print("Type in 'Task' and 'Priority' for your To-Do List")
    task = str(input(" Enter a Task: ")).strip()
    priority = str(input(" Enter Priority: ")).strip()
    return task, priority
```

Figure 5. A function input_new_task_and_priority() in PyCharm

```
@staticmethod
def input_task_to_remove():
    """  Gets the task name to be removed from the list

    :return: (string) with task
    """
    #pass  # TODO: Add Code Here!

    task = str(input("Item to Remove:  ")).strip()
    print()
    return task
```

Figure 6. A function of input_task_to_remove() in PyCharm

2.3.4. Main Body of Script

As the main body of the script was already well organized, I did not modify the script. But I spent quite some time to understand how it is structured, especially focusing on how different

functions were added. Using functions in the script helped making sense of the codes a lot easier (particularly, using class in each function).

2.4. Running the modified ToDoList Python script and verifying that it worked

To run the modified Python script in PyCharm, I right-clicked and selected “Run Assignment06_SangheeKang.” I typed in four options one by one to check whether all of them perform its task successfully. In the existing text file, I had two tasks (cleaning: high; ordering a dresser: mid; see Figure 7) and added two tasks (doing laundry: low; grocery shopping: mid) using the program and removed the task “cleaning”. When I am done with running all functions in the code, I checked output (Figures 8-13) and the text file saved in the same folder. As shown in Figure 14, the file had the three remaining tasks. Therefore, my python code completed the task successfully.

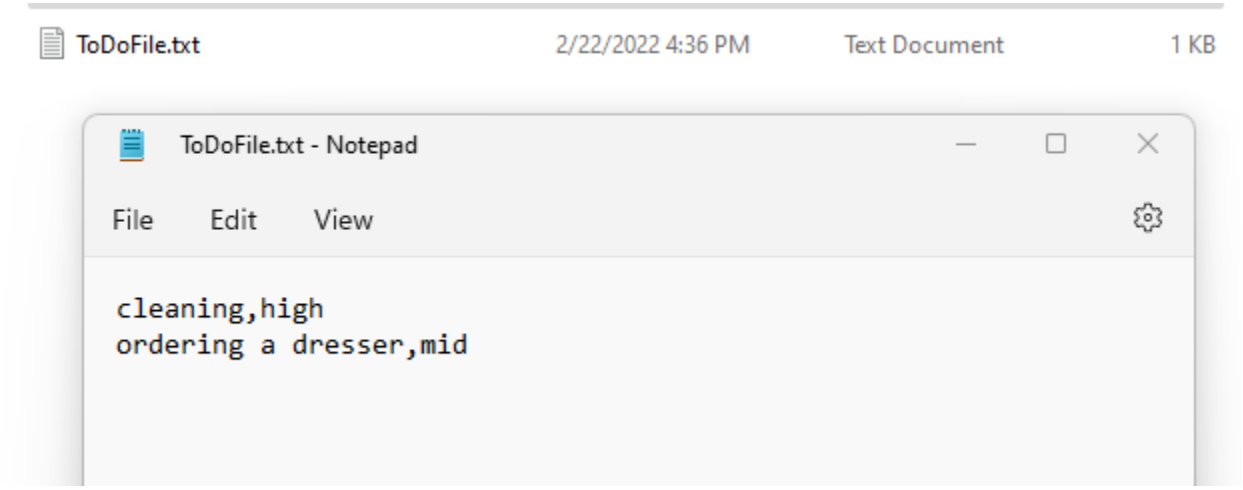

Figure 7. Content of the exiting text file

```
C:\_PythonClass\Assignment06_SangheeKang\venv\Scripts\python.exe C:/_PythonClass/Assignment06_SangheeKang/Assigment06_SangheeKang.py
******* The current tasks ToDo are: *******
cleaning (high)
ordering a dresser (mid)
*****************************************
```

Figure 8. Output: Showing the current Tasks in the list of dictionaries rows

```
        Menu of Options
        1) Add a new Task
        2) Remove an existing Task
        3) Save Data to File
        4) Exit Program


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

Type in 'Task' and 'Priority' for your To-Do List
 Enter a Task: doing laundry
 Enter Priority: low
******* The current tasks ToDo are: *******
cleaning (high)
ordering a dresser (mid)
doing laundry (low)
*****************************************
```

Figure 9. Final output of the modified Python script in PyCharm (Option 1)

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

Type in 'Task' and 'Priority' for your To-Do List
 Enter a Task: grocery shopping
 Enter Priority: mid
******* The current tasks ToDo are: *******
cleaning (high)
ordering a dresser (mid)
doing laundry (low)
grocery shopping (mid)
*****************************************
```

Figure 10. Final output of the modified Python script in PyCharm (Option 2)

```
Which option would you like to perform? [1 to 4] - 2

Item to Remove:  cleaning

******* The current tasks ToDo are: *******
ordering a dresser (mid)
doing laundry (low)
grocery shopping (mid)
*****************************************
```

Figure 11. Final output of the modified Python script in PyCharm (Option 3)

```
        Menu of Options
        1) Add a new Task
        2) Remove an existing Task
        3) Save Data to File
        4) Exit Program


Which option would you like to perform? [1 to 4] - 3

Data Saved!
******* The current tasks ToDo are: *******
ordering a dresser (mid)
doing laundry (low)
grocery shopping (mid)
*****************************************
```

Figure 12. Final output of the modified Python script in PyCharm (Option 4)

```
        Menu of Options
        1) Add a new Task
        2) Remove an existing Task
        3) Save Data to File
        4) Exit Program


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

Goodbye!

Process finished with exit code 0
```

Figure 13. Final output of the modified Python script in PyCharm (Option 5)

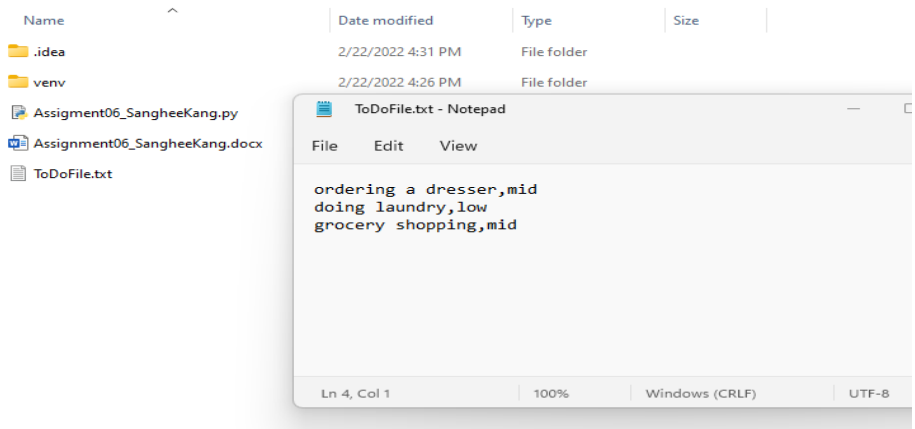

Figure 14. A screenshot of the content of ToDoFile.txt (final output) and its location

Afterwards, I ran the Python code in a command shell. I followed the same steps as I did in the PyCharm and the same output was yielded including the content of the text file. (see Figures 15 and 16).

```
Microsoft Windows [Version 10.0.22000.493]
(c) Microsoft Corporation. All rights reserved.

C:\Users\sangh>CD C:\_PythonClass\Assignment06_SangheeKang

C:\_PythonClass\Assignment06_SangheeKang>Python.exe Assignment06_SangheeKang.py
******* The current tasks ToDo are: *******
cleaning (high)
ordering a dresser (mid)
*****************************************


        Menu of Options
        1) Add a new Task
        2) Remove an existing Task
        3) Save Data to File
        4) Exit Program


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

Type in 'Task' and 'Priority' for your To-Do List
 Enter a Task: doing laundry
 Enter Priority: low
******* The current tasks ToDo are: *******
cleaning (high)
ordering a dresser (mid)
doing laundry (low)
*****************************************


        Menu of Options
        1) Add a new Task
        2) Remove an existing Task
        3) Save Data to File
        4) Exit Program


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

Type in 'Task' and 'Priority' for your To-Do List
 Enter a Task: grocery shopping
 Enter Priority: mid
******* The current tasks ToDo are: *******
cleaning (high)
ordering a dresser (mid)
doing laundry (low)
grocery shopping (mid)
*****************************************
```

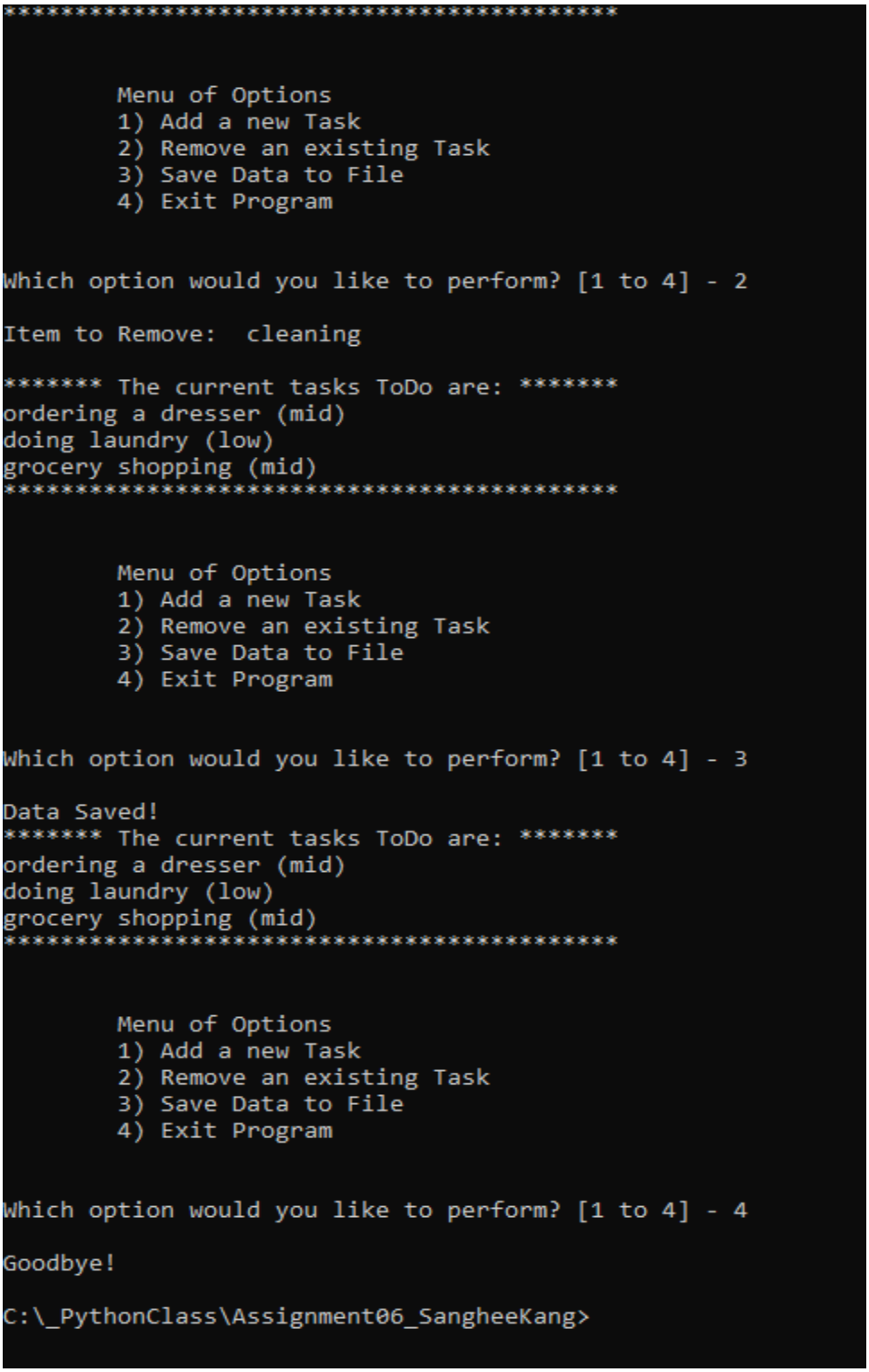

Figure 15. Final output of the modified Python script in a command shell (top and bottom)

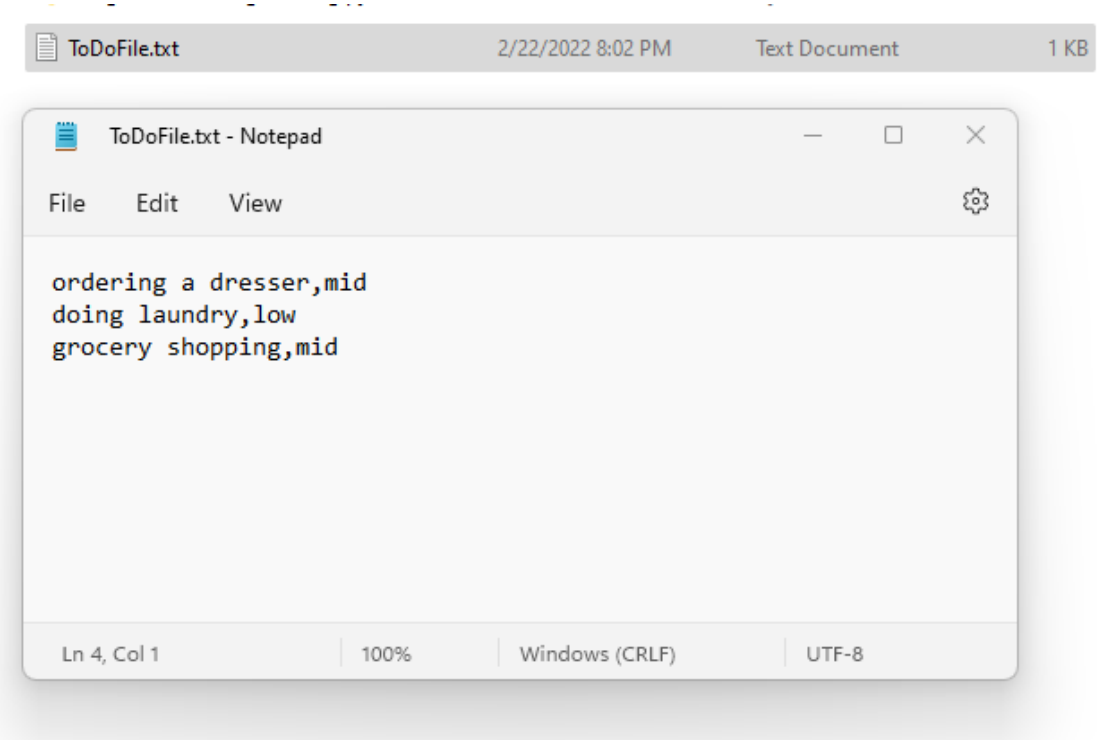

Figure 16. A screenshot of the content of ToDoFile.txt (final output) and its location after running the code in the command shell

3. Summary

For this assignment, I modified a code that is created by someone else by adding functions so that the code can be more organized. At first, it took time to make sense of how the script is structured and how I can add functions. But after completing the code and running it, I found that using functions really helped make the code look neater and easier to make sense. I hope I can gain more skills and get more practice on how to create codes containing functions.