Name: Mitsuyo Kawano
Date: May 24, 2023
Course: Foundations of Programming Python
Assignment 06
GitHubURL: https://github.com/mitsuyojp/IntroToProg-Python-Mod06

The Steps in Performing the Assignment Script

Introduction

In Module 6, we learned about Functions, Parameters, Arguments, Return Values, Global and Local variables and Classes. Assignment 06 is to modify a script that manages a "ToDo list." The starter Python file was written with classes and functions that we learned in this module.
I'd like to explain the steps I took in performing this assignment.

What are steps?

1. **Modify Script Header**
   As always, I wrote (modified) this script header first. I added my name and date for the Changelog section.

```
# ------------------------------------------------------------------------ #
# 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 (MKawano,05/23/2023,add function script):
# RRoot,1.1.2030,Created started script
# MKawano,05/23/2023,Modified code to complete assignment 06
# ------------------------------------------------------------------------ #
```

Figure 1 A Screenshot of Header of Python Script (PyCharm)

2. Understand the structure of this script.

Next, I read the entire starter Python script and understand the structure of this script. There are four sections in this script.

1) Data Declaration
2) Processor Class
3) Input /Output Class
4) Main script which calls functions from two classes.

1) Data Declaration

```
12  # Data ---------------------------------------------------------------------- #
13  # Declare variables and constants
14  file_name_str = "ToDoFile.txt"  # The name of the data file
15  file_obj = None  # An object that represents a file
16  row_dic = {}  # A row of data separated into elements of a dictionary {Task,Priority}
17  table_lst = []  # A list that acts as a 'table' of rows
18  choice_str = ""  # Captures the user option selection
19
```

Figure 2 A Screenshot of Data Declaration Script (PyCharm)

2) Processor Class (has 4 functions)

```
21  # Processing  ----------------------------------------------------------
22  class Processor:
23      """  Performs Processing tasks """
24
25      @staticmethod
26  >   def read_data_from_file(file_name, list_of_rows):...
41
42      @staticmethod
43  >   def add_data_to_list(task, priority, list_of_rows):...
56
57      @staticmethod
58  >   def remove_data_from_list(task, list_of_rows):...
68
69      @staticmethod
70  >   def write_data_to_file(file_name, list_of_rows):...
```

Figure 3 A Screenshot of Processor Class (PyCharm)

3) Input /Output Class (has 5 functions)

```
# Presentation (Input/Output)  ---------------------------------------- #


class IO:
    """ Performs Input and Output tasks """

    @staticmethod
    def output_menu_tasks():...

    @staticmethod
    def input_menu_choice():...

    @staticmethod
    def output_current_tasks_in_list(list_of_rows):...

    @staticmethod
    def input_new_task_and_priority():...

    @staticmethod
    def input_task_to_remove():...
```

Figure 4 A Screenshot of Input/Output Class (PyCharm)

4) Main script which calls functions from two classes.

```
# Main Body of Script  ------------------------------------------------- #


# Step 1 - When the program starts, Load data from ToDoFile.txt.
Processor.read_data_from_file(_file_name=file_name_str, list_of_rows=table_lst)  # read file data

# Step 2 - Display a menu of choices to the user
while (True):
    # Step 3 Show current data
    IO.output_current_tasks_in_list(list_of_rows=table_lst)  # Show current data in the list/table
    IO.output_menu_tasks()  # Shows menu
    choice_str = IO.input_menu_choice()  # Get menu option

    # Step 4 - Process user's menu choice
    if choice_str.strip() == '1':...

    elif choice_str == '2':...

    elif choice_str == '3':...

    elif choice_str == '4':...
```

Figure 5 A Screenshot of Main Script (PyCharm)

3. Processor Class
Add code in add_data_to_list function.

I read Processor Class. There are four functions in this class.
I have to write a script in add_data_to_list function. I wrote a script as below.

```
@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 to add more data to:
    :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 6 A Screenshot of add_data_to_list function (PyCharm)

4. *Processor Class*
remove_data_from_list function.

Now I have to add some code to remove data function. Here is the script I wrote first.
(When I tested this code, it didn't work well. I'll explain about the errors later.)

```
@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:
        task, priority = dict(row).values()
        if task == strRemoveTask:
            lstTable.remove(row)

    return list_of_rows
```

Figure 7 A Screenshot of remove_data_from_list function. (PyCharm)

5. Processor Class
write_data_to_file function.

Now I need to finish writing the data function. Here is the script of this step.

```
@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!
    objFile = open(file_name, "w")
    for dicRow in list_of_rows:
        objFile.write(dicRow["Task"] + "," + dicRow["Priority"] + "\n")
    objFile.close()

    return list_of_rows
```

Figure 8 A Screenshot of write_data_to_file function (Pycharm)

6. Input / Output Class
input_new_task_and_priority function.

Now we moved on to next class! I wrote this function as below.

```
@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!
    strTask = input("Task:")
    strPriority = input("Priority:")
    dicRow = {"Task": strTask, "Priority": strPriority}
    list_of_rows.append(dicRow)
```

Figure 9 A Screenshot of input_new_task_and_priority Function (PyCharm)

After writing a code, I read the description comments and realized I don't need to append the list here. Simply return (string, string). I revised the code as below.

```
@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!
    strTask = input("Task:")
    strPriority = input("Priority:")
    return strTask,strPriority
```

Figure 10 A Screenshot of revised code of input_new_task_and_priority Function (PyCharm)

7. Input / Output Class input_task_to_remove function.

This function is getting the user input data to remove the item, and simply return the input data.

```
@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!
    strRemoveTask = input("Which Task would you like to remove?:")
    return strRemoveTask
```

Figure 11 A Screenshot of input_task_to_remove Function (PyCharm)

8. Test the code and fix issues

Now it's time to test the code. Here is the first error.
Simple error that says "ToDoFile.txt" doesn't exist. Created the file and fixed the issue.

```
  ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
 File "C:\_PythonClass\Assingment06\Assigment06_MKawano.py", line 35, in read_data_from_file
  file = open(file_name, "r")
         ^^^^^^^^^^^^^^^^^^^^
FileNotFoundError: [Errno 2] No such file or directory: 'ToDoFile.txt'
```

Figure 12 A Screenshot of first error (PyCharm)

Choice 1 (Add a new Task) worked successfully.

```
        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

Task:Window Cleaning
Priority:High
******* The current tasks ToDo are: *******
Dishes (High)
Laundry (Low)
Window Cleaning (High)
*******************************************
```

Figure 13 A Screenshot of Choice 1 working! (PyCharm)

Choice 3 (Save Data to File) worked successfully as well.

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

Data Saved!
******* The current tasks ToDo are: *******
Dishes (High)
Laundry (Low)
Window Cleaning (High)
```

Figure 14 A Screenshot of Choice 3 working! (PyCharm)

Choice 4 (Exit Program) worked successfully as well.

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

Goodbye!

Process finished with exit code 0
```

Figure 15 A Screenshot of Choice 4 working! (PyCharm)

Choice 2 (Remove existing task) is not working. Here is the error message.

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

Which Task would you like to remove?:Laundry
Traceback (most recent call last):
  File "C:\_PythonClass\Assingment06\Assigment06_MKawano.py", line 177, in <module>
    table_lst = Processor.remove_data_from_list(task=task, list_of_rows=table_lst)
                ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\_PythonClass\Assingment06\Assigment06_MKawano.py", line 69, in remove_data_from_list
    if task == strRemoveTask:
               ^^^^^^^^^^^^^
NameError: name 'strRemoveTask' is not defined

Process finished with exit code 1
```

Figure 16 A Screenshot of error message. (PyCharm)

I checked three parts of the script that are related to removing the item.
Here is the script that causes the error. In this function, they don't know what strRemoveTask is since this parameter is declared later in the different function in different class.

```
@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:
        task, priority = dict(row).values()
        if task == strRemoveTask:
            list_of_rows.remove(row)

    return list_of_rows
```

Figure 17 A Screenshot of remove_data_from_list Function. (PyCharm)

Here is two other parts of scripts that related to removing item.
This function is getting user input data.

```
@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!
    strRemoveTask = input("Which Task would you like to remove?:")
    return strRemoveTask
```

Figure 18 A Screenshot of input_task_to_remove Function (PyCharm)

This script is calling functions to remove a task that the user input.

```
elif choice_str == '2':  # Remove an existing Task
    task = IO.input_task_to_remove()
    table_lst = Processor.remove_data_from_list(task=task, list_of_rows=table_lst)
    continue  # to show the menu
```

Figure 19 A Screenshot of calling functions to remove a task (PyCharm)

To fix this issue, I simply changed **strRemoveTask** to **task** as below. Let's see what happens.

```
@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:
        task, priority = dict(row).values()
        if task == task:
            list_of_rows.remove(row)

    return list_of_rows
```

Figure 20 A Screenshot of revised script (PyCharm)

It runs but this is not right. It saved Laundry, but I wanted it removed. And removed all other data that were supposed to be saved!!

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

Which Task would you like to remove?:Laundry
******* The current tasks ToDo are: *******
Laundry (Low)
*****************************************
```

Figure 21 A Screenshot of the test result (PyCharm)

This is the code processing error. The script needs to go through the list_of_rows and find the matching task that the user input. To do that, I changed the processing function as below.

```
    @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 dicRow in list_of_rows:
            if dicRow["Task"] == task:
             list_of_rows.remove(dicRow)

        return list_of_rows
```

Figure 22 A Screenshot of revised script. (PyCharm)

It runs successfully!

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

Which Task would you like to remove?:Dishes
******* The current tasks ToDo are: *******
Laundry (Low)
Window (HIgh)
Cleaning (Low)
*****************************************
```

Figure 23 A Screenshot of the test result (PyCharm)

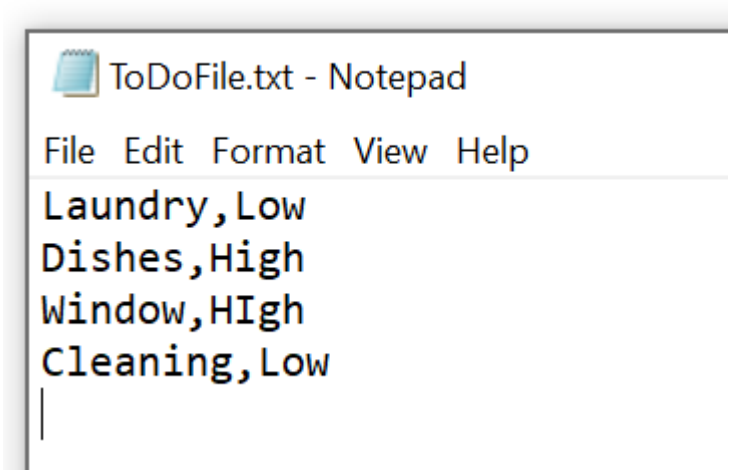

Figure 24 A Screenshot of test result (ToDoFile.txt)

I also ran the script in Windows command line. It ran successfully.

```
Python: can't open file 'C:\\_PythonClass\\Assingment06\\Assingment06_Mkawano.py': [Errno 2] No such file or directory

C:\_PythonClass\Assingment06>Python Assigment06_MKawano.py
******* The current tasks ToDo are: *******
Laundry (Low)
Dishes (High)
Window (HIgh)
Cleaning (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

Task:Garbage
Priority:High
******* The current tasks ToDo are: *******
Laundry (Low)
Dishes (High)
Window (HIgh)
Cleaning (Low)
Garbage (High)
*****************************************
```

Figure 25 A Screenshot of the script running in Command Line. (CMD)

Summary

Here is the summary of steps to complete this assignment.

1. Modify Script Header

2. Understand the structure of this script.
   1) Data Declaration
   2) Processor Class (has 4 functions)
   3) Input/Output Class (has 5 functions)
   4) Main script which calls functions from two classes

3. Processor Class
   code in add_data_to_list function.

4. Processor Class
   remove_data_from_list function.

5. Processor Class
   write_data_to_file function.

6. Input / Output Class
   input_new_task_and_priority function.

7. Input / Output Class
   input_task_to_remove function.

8. Test the code and fix issues