Kapkyi Lwai

Foundation Of Programing: Python

Assignment 6: Functions

November 24, 2022

# Functions (To Do File)

## Introduction:

This assignment is similar to the previous assignment 5. In this assignment, a new project is created in PyCharm for a program to create a 'ToDoFile.txt' file. The program will allow user to input data, view their data, save their data, and also remove their data. The program scripted using functions on top of the starter code given for the assignment to allow reading and writing to a text file. This project will also be uploaded to GitHub repository and also, we are to create a web page.

## Getting Started:

I created a new project on PyCharm and I named it 'Assignment06' and open a python file called 'Assignment06_starter.py'. The codes from this file were added from Module 6 file to be modified and add necessary codes for the program to process. Again, the program contains two columns of data, 'Task' and 'Priority'. These columns will be loaded into a Python dictionary object as each object represents as one row of data. The row data were then added as lists and create a table of data.

## Headers and Variables:

After the starter script have been added to the new Python project, I modified the headers by adding my information and declared my variables (Figure 1). This project also contains many

pseudo codes and comments from top to bottom with '#' at beginning followed by comments. The program is started by declaring variables and constants. (Figure 1)

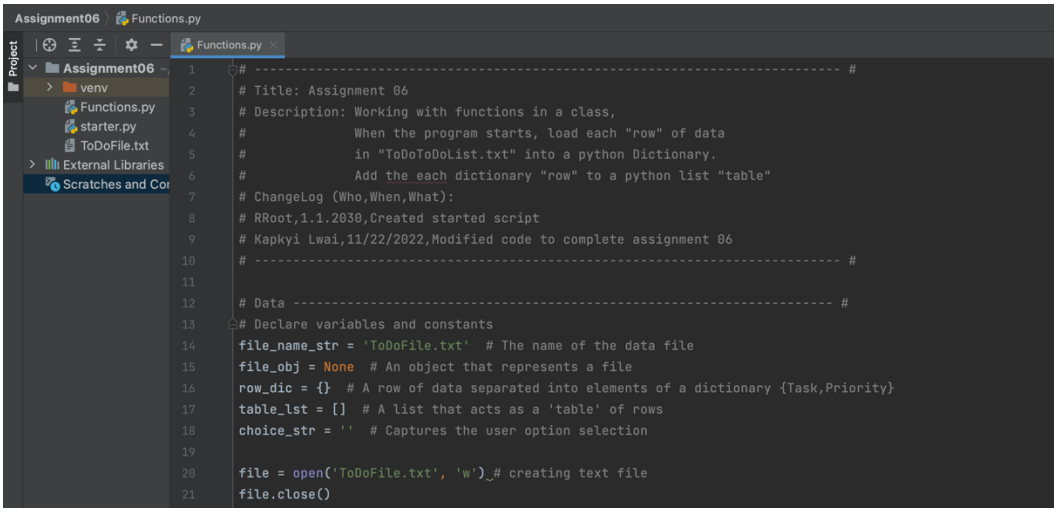



*Figure 1: headers and variables with comments #*

## **Writing The Script:** (Load data in text file ToDoFile.txt)

The first step for writing the script was to create a text file 'ToDoFile.txt' for data to be added. Similarly, from assignment 5, I created a code that allows to load data in the file into list of dictionaries rows. In this assignment as shown in above Figure 1, I created a code to create text file with 'w' mode to avoid error and allow my program to start.

## **Writing The Script:** (main body script)

The main body of script was given on the starter script (reference to Figure 2). This part of the scripts is written to allow the program to read and write file, shows current data and display menu of choices, process data and save data, removing data from list, and exit program.

```
# 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':  # Add a new Task
        task, priority = IO.input_new_task_and_priority()
        table_lst = Processor.add_data_to_list(task=task, priority=priority, list_of_rows=table_lst)
        continue  # to show the menu

    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

    elif choice_str == '3':  # Save Data to File
        table_lst = Processor.write_data_to_file(file_name=file_name_str, list_of_rows=table_lst)
        print('Data Saved To File!: Press Enter To Continue')
        continue  # to show the menu

    elif choice_str == '4':  # Exit Program
        print('GoodByE!')
        break  # by exiting loop
```

*Figure 2: Main Body of Script*

## **Writing The Script:** (processing read and add data)

For this assignment, a new method is used to process the program. The program is coded by defining functions and call the functions. Functions are writing with the key code ‘def’ to declare its name applicable to the action it will perform. The first part of modifying/adding code to the starter script is defining function under class ‘processor’ (Figure 3). The new added codes are written under the line *‘# ToDo: Add Code Here’*. The first added code was to add data to list of dictionary rows (list_of_rows). The program is also coded to display current data by using ‘return list_of_rows’ after each choice from menu.

```python
# 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

    @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)        # adding new row to list_of_rows
        return list_of_rows
```

*Figure 3: processing data (read and add data to list)*

## **Writing The Script:** processing (remove, write, and save)

The next added scripts are for removing data (task), writing data to text file, and saving data to file. For removing a data (task), I gave a variable name ‘taskRemove’ with ‘if’ and ‘else’ functions with True and False statements. (Figure 4When the user enters the matching task, they want to remove from the list, the program will remove the task and display a message back to user it has been remove. When user enter data that was not in the list, the program will deny the removal by displaying message that the task enter was not in the list.

The 'for' code is used to write data from list of rows to the text file 'ToDoFile.txt'. The data entered were then printed in the text file. The data were then saved to file with code seen in last part of Figure 4 below. A display message will then show to user their data is saved to file and a direction to press enter to continue. The data are scripted to convert to string with lower and strip methods.

```
    @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!
        taskRemove = False  # variable used to find 'task' for removal
        for item in list_of_rows:
            if item['Task'].lower().strip() == task.lower().strip():
                list_of_rows.remove(item)       # deleting/removing a task
                taskRemove = True
        if taskRemove == True:  # user input correct data
            print(task.lower().strip() + ' has been removed')
        else:                   # use input data not found in list
                print(task.lower().strip() + ' is not on the list')
        return list_of_rows

    @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!
        print("\n******* The current Task and Priority are: *******")
        for row in list_of_rows:
            print(row['Task'] + ' (' + row['Priority'] + ')')
        print("*************************************************")
```

```

        #saving data to file
        file = open(file_name, 'w')
        for row in list_of_rows:
            file.write(row['Task'] + ',' + row['Priority'] + '\n')
        file.close()
        input('Data Saved To File!: press ENTER to continue')
        return list_of_rows
```

*Figure 4: processing data (remove, write to file, save to file)*

## **Writing The Script:** presentation (input/output)

This part of the program scripts in Figure 5 below are already given on starter code. The codes will allow the program to display menu of choices for user and user input/output data in list of dictionary rows.

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

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

    @staticmethod
    def output_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) Exit Program
        ''')
        print()  # Add an extra line for looks

    @staticmethod
    def input_menu_choice():
        """ Gets the menu choice from a user
        :return: string
        """
        while (True):
            choice = str(input('Which option would you like to perform? [1 to 4] - ')).strip()
            if choice in ('1', '2', '3', '4'):
                print()  # Add an extra line for looks
                return choice
            else:
                print('Invalid Entry: enter number 1-4'+'\n')
```

```
    @staticmethod
    def output_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 row in list_of_rows:
            print(row['Task'] + ' (' + row['Priority'] + ')')
        print('*******************************************')
        print()  # Add an extra line for looks
```

*Figure 5: starter code for presenting user of menu of choices and current data*

## **Writing The Script:** user input task and remove task

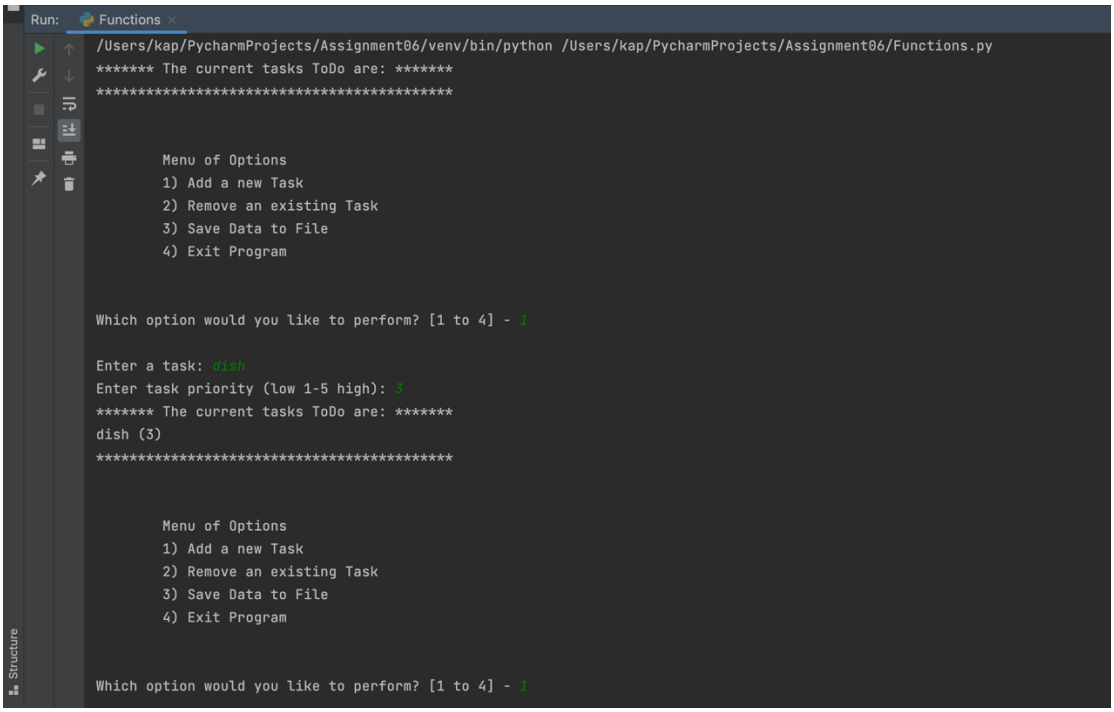
The last part of modifying/adding code is to use function to get user input task and priority data to the program and request user to enter data to be removed. Variables are 'task' and 'priority' with 'input' function to allow user to enter data. The added codes are found under *'# ToDo: Add Code Here'* (Figure 6).

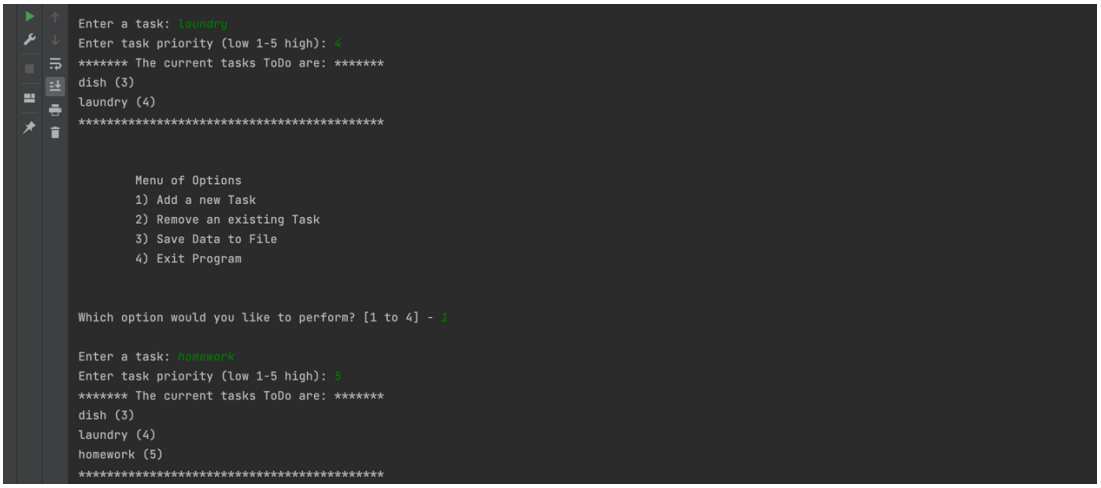
```
144     @staticmethod
145     def input_new_task_and_priority():
146         """  Gets task and priority values to be added to the list
147 
148         :return: (string, string) with task and priority
149         """
150         # TODO: Add Code Here!
151         task = str(input('Enter a task: ')).strip() # get user input data
152         priority = (input('Enter task priority (low 1-5 high): ')).strip() #user input data
153         return task, priority
154 
155     @staticmethod
156     def input_task_to_remove():
157         """  Gets the task name to be removed from the list
158 
159         :return: (string) with task
160         """
161         # TODO: Add Code Here!
162         task = str(input('Enter a task name you wish to remove: ')).lower().strip() # request user to input data
163         return (task)
164         print()  # extra line for looks
```

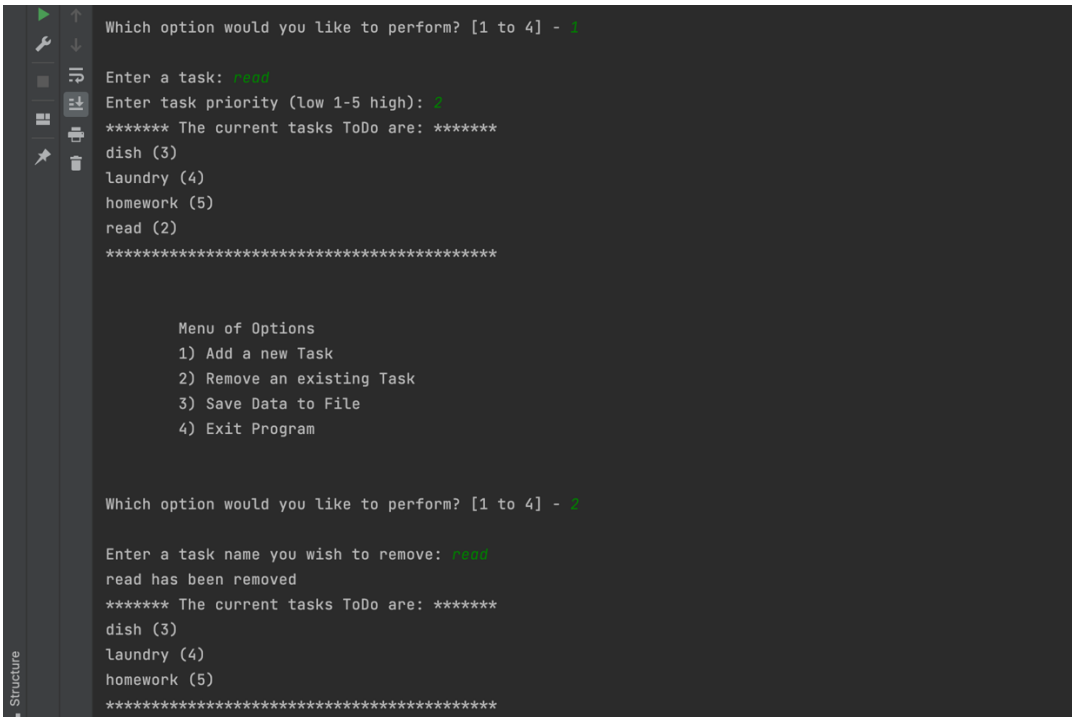
*Figure 6: get user input task, priority, and task to be removed*

## **Running The Script:**

Lastly, I ran the program in PyCharm as well as on Shell window to double check it is working. The program ran smoothly in PyCharm as shown in screenshots (Figure 7) below as well as on Shell window (Figure 8). The text file 'ToDoFile.txt' was also created successfully and data were saved as list of rows.


Run:
Functions
/Users/kap/PycharmProjects/Assignment06/venv/bin/python /Users/kap/PycharmProjects/Assignment06/Functions.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) Exit Program
Which option would you like to perform? [1 to 4] - 1
Enter a task: dish
Enter task priority (low 1-5 high): 3
******* The current tasks ToDo are: *******
dish (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] - 1
Structure


Enter a task: laundry
Enter task priority (low 1-5 high): 4
******* The current tasks ToDo are: *******
dish (3)
laundry (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] - 1
Enter a task: homework
Enter task priority (low 1-5 high): 5
******* The current tasks ToDo are: *******
dish (3)
laundry (4)
homework (5)
*****************************************



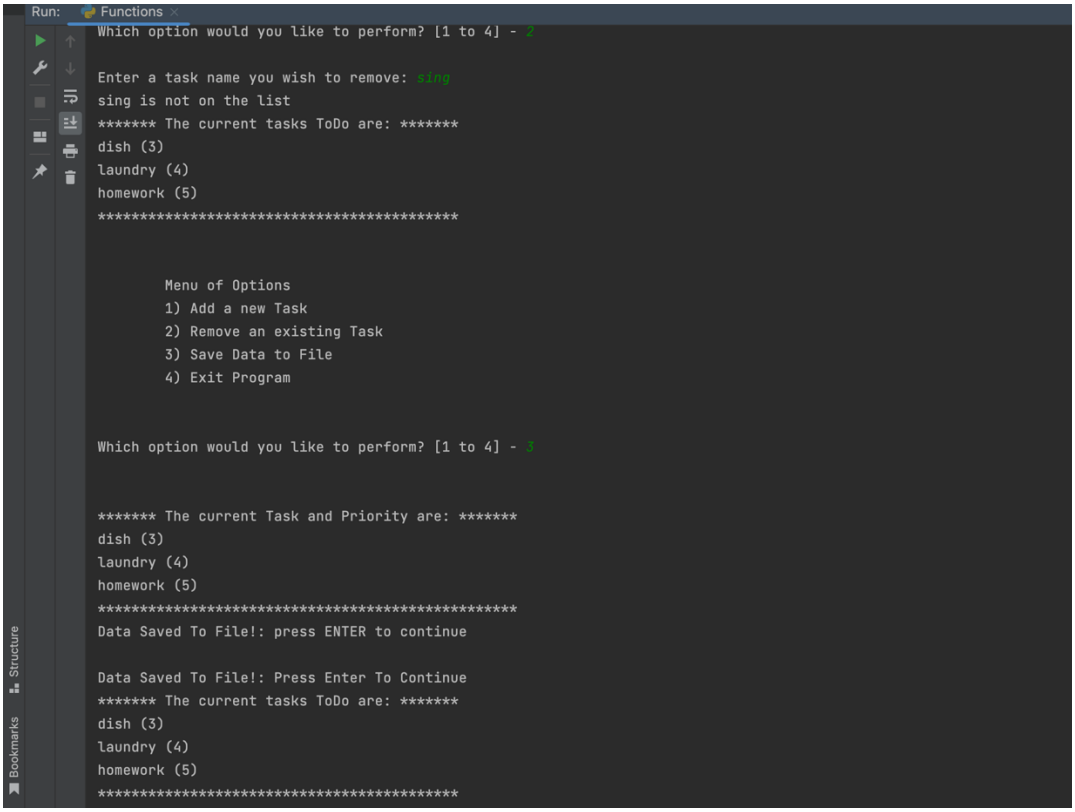
```
Which option would you like to perform? [1 to 4] - 1

Enter a task: read
Enter task priority (low 1-5 high): 2
******* The current tasks ToDo are: *******
dish (3)
laundry (4)
homework (5)
read (2)
****************************************


        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] - 2

Enter a task name you wish to remove: read
read has been removed
******* The current tasks ToDo are: *******
dish (3)
laundry (4)
homework (5)
****************************************
```



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

Enter a task name you wish to remove: sing
sing is not on the list
******* The current tasks ToDo are: *******
dish (3)
laundry (4)
homework (5)
****************************************


        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


******* The current Task and Priority are: *******
dish (3)
laundry (4)
homework (5)
************************************************
Data Saved To File!: press ENTER to continue

Data Saved To File!: Press Enter To Continue
******* The current tasks ToDo are: *******
dish (3)
laundry (4)
homework (5)
****************************************
```

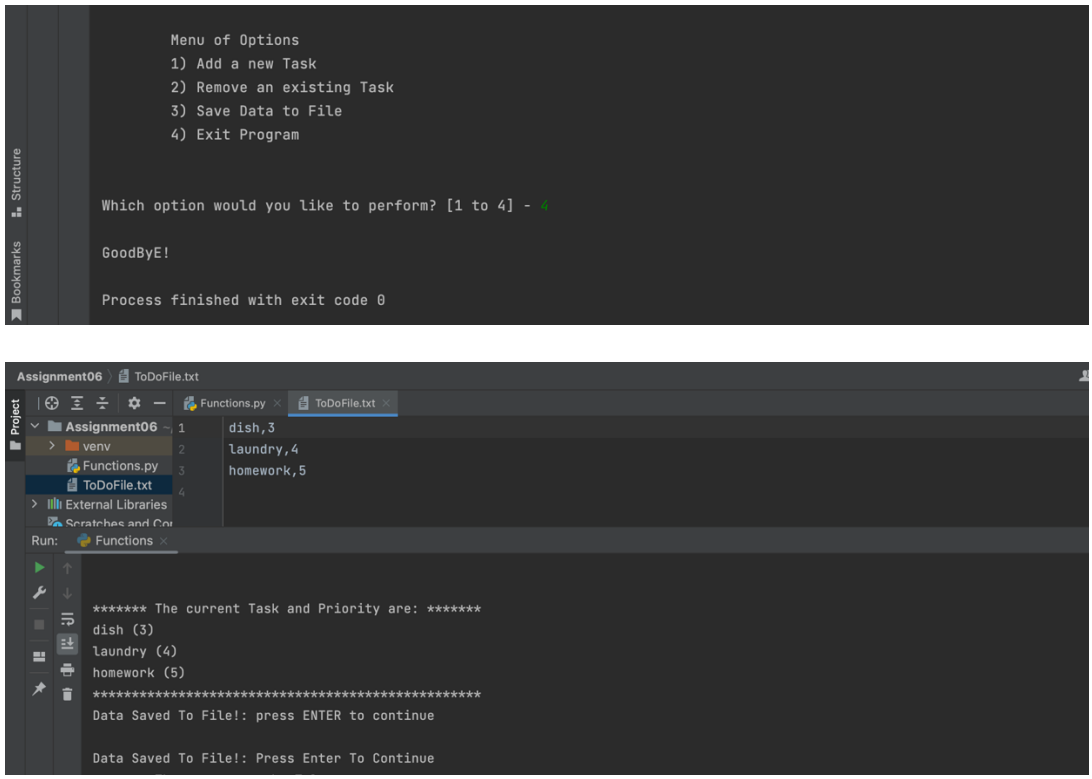



*Figure 7: Program running successful in PyCharm*

```
Python 3.10.8 (v3.10.8:aaaf517424, Oct 11 2022, 10:14:40) [Clang 13.0.0 (clang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======= RESTART: /Users/kap/Desktop/PythonClass/Assignment06/Functions.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) Exit Program


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

Enter a task: homework
Enter task priority (low 1-5 high): 5
******* The current tasks ToDo are: *******
homework (5)
*********************************************


        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

Enter a task: read
Enter task priority (low 1-5 high): 2
******* The current tasks ToDo are: *******
homework (5)
read (2)
*********************************************


        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] - 2

Enter a task name you wish to remove: read
read has been removed
******* The current tasks ToDo are: *******
homework (5)
*********************************************
```

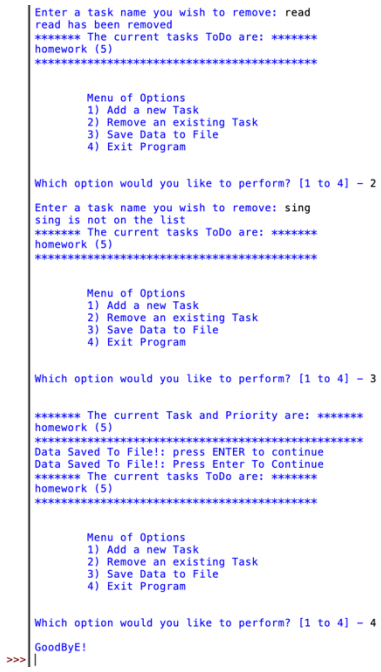



*Figure 8: program running in Shell window*

## Upload to GitHub:

This project was uploaded to my GitHub account as public. The project was uploaded for others to view and provide comments or edits. In addition, a webpage for my assignment was also created. (Figure 9)

**Assignment link:** https://github.com/kapkyi/IntroToPro-Python-Mod06

**Webpage link:** https://kapkyi.github.io/IntroToPro-Python-Mod06/

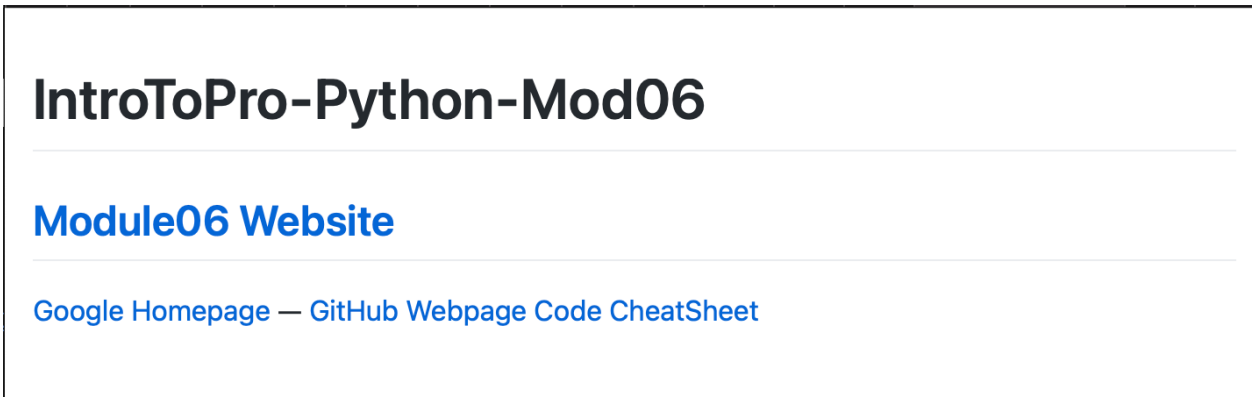



***Figure 9:*** *GitHub link and webpage created*

## Summary:

In conclusion, this project was created from already made outline and it was modified with added codes to complete the program. The assignment is similar to previous assignment and in addition, the function method was introduced. From this assignment, I learned how to edit and modify Python script and upload my project to GitHub as well as how to created web page. The assignment allows me to demonstrate my knowledge of how to use functions and separation of codes. From Chapter 6 of the textbook (Python Programing, 3rd edition, Michael Dawson, 2010), looking at the Module 6 documentation, lecture video and demonstrations, and the starter Python script outline provided for the assignment, I was able to successfully complete the program.