Assignment 06: To Do List with Functions & Classes

# Introduction

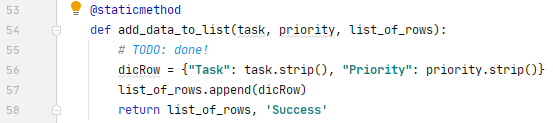
This week, students were introduced to functions and classes and used these to develop a user-inputted To Do list. The list was read from a .txt file and a menu was displayed to the user to add an item, remove an item, save, reload, and exit the program.

# Solution

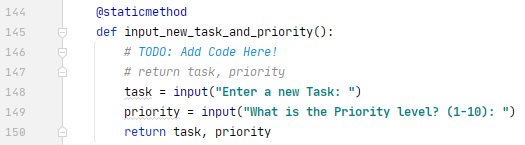
Students were instructed to complete the starter file by filing in the remainder of the Processor and IO classes. The script was ran successfully in PyCharm and Windows Command Processor.

# Adding Task and Priority

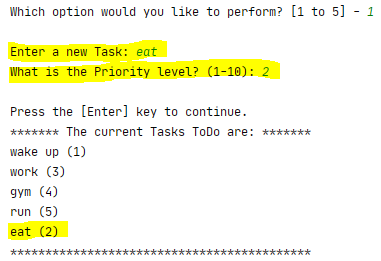
The processor class “add\_data\_to\_list” function passes through the parameters captured from the IO class “input\_new\_task\_and\_priority” function: task and priority. The Processor function appends the new row containing the IO inputs to the existing list\_of\_rows. Figure 3 displays the use of these functions.



***Figure 1: Processor function to add data***



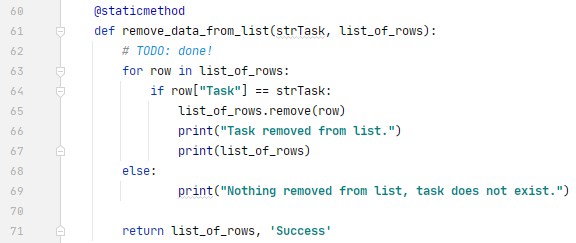
***Figure 2: IO function to receive user input***



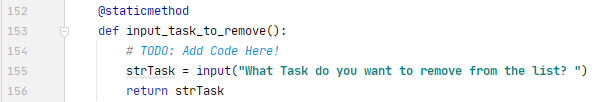
***Figure 3: Adding data to list***

# Removing Task from To Do List

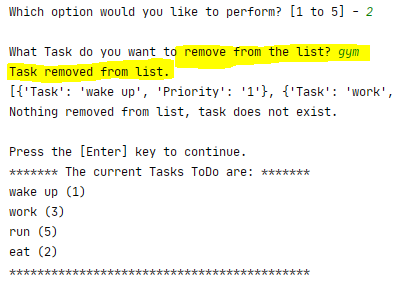
The processor class “remove\_data\_from\_list” function passes through the parameter captured from the IO class “input\_task\_to\_remove” function: strTask. The Processor function loops through each row of the list and checks if the Task matches strTask, if there is a match, then the row is removed and a message is displayed stating the task has been removed.. Figure 6 displays the use of these functions.



***Figure 4: Processor function to remove data***



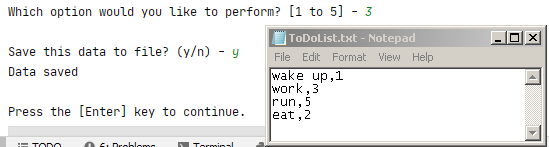
***Figure 5: IO function to remove data***



***Figure 6: Removing data from list***

# Saving Data

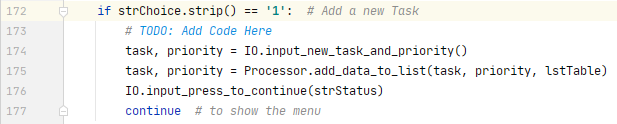
Data was saved via the write\_data\_to\_file function within the processor class. The file is written in Write mode and each row currently in memory is written via objFile.write. The file is closed a message is displayed, output seen in Figure 7 below.



***Figure 7: Saving data to .txt file***

# Menu Selection

For each menu user selection [1-5], the script calls the appropriate IO function to receive the input and then the Processor function to process the input. For selection #1, for example, input\_new\_task\_and\_priority function is called, and the user inputs the new task and new priority per the above Figure 1 and Figure 2. The script to call both functions is seen below in Figure 8. A similar approach is used for each menu selection [1-5].



***Figure 8: User Menu Selection #1***

# Summary

This method of keeping functions separated and then calling each as needed keeps the code organized and easier to maintain. This would be especially helpful for long scripts.

# Command Processor and .txt

