Ahson Butt

Foundations of Programming Python

Assignment 06

November 26, 2020

<https://ahsonb20.github.io/ITFnd100-Mod6/>

Working with Functions and Classes

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Introduction

This paper details the work that was completed for Assignment 06. I will review the things I needed to do to prepare for the assignment. Then I will discuss writing the script for the assignment. Finally, I will share what I learned and try to address some of the questions the instructor posed in the provided assignment overview.

Lecture and Study

After watching the lecture, I read through Chapter 6 of the text. Next I moved on to the supplied webpage and video.

In watching the lecture and completing the reading I answered the following:

What is a function? A function is a way of grouping one or more statements together. Calling a function executes the statements in the function.

What are parameters? Parameters allow you to pass values into the function for processing.

What are arguments? Values that are passed into parameters.

What is the difference between parameters and arguments? Arguments get passed to parameters, Parameters are variables that receive arguments. When you define a function, you include the parameters in the definition. The arguments that are included inside the function work with the parameters outside the function.

What are return values? Return values make a function act like an expression. You capture returning values of a function in variables or you can use them immediately without placing the result in a variable. Functions can return one or more values.

What is the difference between a global and a local variable? Local variables are defined inside of a function and can only be used inside the function that contains them. Global variables are defined outside functions at the top level of the program and can be used across the entire program.

How do you use functions to organize your code? Functions are a good way to ‘chunk’ your program. While the main loop runs, it takes advantage of this chunking by using the functions as sub-components to the main loop. Setting up functions allows the coder to write and test in segments instead of doing an integrated test of the program all at the end of completion.

What is the difference between a function and a class? A function is a small program, a class is a grouping of functions. A class can be used like a library of resources for the main loop.

How do functions help you program using the “Separations of Concerns" pattern? Portions of a program can be sectioned off into Concerns. The main loop will take advantage of this when the script is run by calling the functions. Each function can be tested individually because it should operate as stand-alone script.

How are the debugging tools use in PyCharm? A user can set breakpoints in the code to pause the program so the user can access the controls in the debugging window. The user can also use the ‘Step Into My Code’ button to walk through each line of the code.

What is a GitHub webpage? A webpage in GitHub that supports a GitHub Repository.

In using the provided script and the additional help for the assignment, I was able to complete the assignment. The script that was written is presented below:

*# ---------------------------------------------------------------------------- #  
# 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):  
# Ahson Butt,11-26-20,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):  
 row = {**"Task"**: str(task).strip(), **"Priority"**: str(priority).strip()}  
 list\_of\_rows.append(row)  
 **return** list\_of\_rows, **'Success'** @staticmethod  
 **def** remove\_data\_from\_list(task, list\_of\_rows):  
 **for** row **in** list\_of\_rows:  
 **if** row[**"Task"**].lower() == task.lower():  
 lstTable.remove(row)  
 *# print("row removed")* **return** list\_of\_rows, **'Success'** @staticmethod  
 **def** write\_data\_to\_file(file\_name, list\_of\_rows):  
 file = open(file\_name, **"w"**)  
 **for** row **in** list\_of\_rows:  
 file.write(row[**"Task"**] + **","** + row[**"Priority"**] + **"\n"**)  
 file.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** row **in** list\_of\_rows:  
 print(row[**"Task"**] + **" ("** + row[**"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 is the task? - "**)).strip()  
 priority = str(input(**"What is the priority? - "**)).strip()  
 **return** task, priority  
  
 @staticmethod  
 **def** input\_task\_to\_remove():  
 task = str(input(**"What is the task? - "**)).strip()  
 print() *# Add an extra line for looks* **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* strTask, strPriority = IO.input\_new\_task\_and\_priority()  
 lstTable, strStatus = Processor.add\_data\_to\_list(strTask, strPriority, lstTable)  
 IO.input\_press\_to\_continue(strStatus)  
 **continue** *# to show the menu* **elif** strChoice == **'2'**: *# Remove an existing Task* strTask = IO.input\_task\_to\_remove()  
 lstTable, strStatus = Processor.remove\_data\_from\_list(strTask, 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(**"Are you sure you want to reload data from file? (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(**"Goodbye!"**)  
 **break** *# and Exit*

Running this script produced the following results in PyCharm:

/usr/local/bin/python3.8 /Users/ahsonbutt/Documents/\_PythonClass/Assignment06/Assigment06\_Starter.py

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

update starter code (high)

write about the how I did it (high)

upload to new GitHub repository (high)

create repository web page (high)

submit to Canvas (high)

do a peer review (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 is the task? - eat

What is the priority? - medium

Success

Press the [Enter] key to continue.

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

update starter code (high)

write about the how I did it (high)

upload to new GitHub repository (high)

create repository web page (high)

submit to Canvas (high)

do a peer review (low)

eat (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] - 2

What is the task? - eat

Success

Press the [Enter] key to continue.

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

update starter code (high)

write about the how I did it (high)

upload to new GitHub repository (high)

create repository web page (high)

submit to Canvas (high)

do a peer review (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] - 3

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

Success

Press the [Enter] key to continue.

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

update starter code (high)

write about the how I did it (high)

upload to new GitHub repository (high)

create repository web page (high)

submit to Canvas (high)

do a peer review (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] - 4

Warning: Unsaved Data Will Be Lost!

Are you sure you want to reload data from file? (y/n) - y

Success

Press the [Enter] key to continue.1

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

update starter code (high)

write about the how I did it (high)

upload to new GitHub repository (high)

create repository web page (high)

submit to Canvas (high)

do a peer review (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 is the task? - eat

What is the priority? - medium

Success

Press the [Enter] key to continue.

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

update starter code (high)

write about the how I did it (high)

upload to new GitHub repository (high)

create repository web page (high)

submit to Canvas (high)

do a peer review (low)

eat (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!

Are you sure you want to reload data from file? (y/n) - y

Success

Press the [Enter] key to continue.

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

update starter code (high)

write about the how I did it (high)

upload to new GitHub repository (high)

create repository web page (high)

submit to Canvas (high)

do a peer review (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] - 5

Goodbye!

Process finished with exit code 0

Running the script through the Terminal produces the same results to running the script in PyCharm see Figure 1:

A picture containing text

Description automatically generated

Figure 1: Script Run in Terminal

Learnings

I feel like we made a big leap this week but upon reflection, it seems the assignment was inline with the lecture. Having struggled with Assignment 5, I was even more lost in this assignment. I understood the basic principal of what the exercise was supposed to teach us, encapsulation, building functions, using these functions between the different classes and the main loop but I didn’t have the right pieces in front of me to assemble the project. The additional lecture provided by Prof. Root helped immensely with putting all this together.

Summary

In this week’s project, we defined and used functions in two classes that was used by our main loop. Coding can be streamlined and frustration of troubleshooting can be reduced using functions. Each function is an independent program, so if a problem occurs in the main loop the source of the problem can be determined using breaks and debugging to pin point which function failed to execute. Classes help us organize these functions in blocks that address our psuedocode.