Lantian Qin

November 24, 2020

IT FDN 110 A

Assignment 06

<https://github.com/NoraQin/IntroToProg-Python-Mod06>

Functions and Classes

Introduction

In this assignment, we work with a start file provided by professor Randall Root and modify to make the program work and also add comments as well as new components to make it more structured. This document will walk through how I modified the file, my thinking process, and review some important knowledge about functions and classes.

Review

By the end of the course, we are asked to think about the answer to the questions below. I will answer them to my best knowledge as a way to review the things I’ve learned.

* What is a function?
  + Functions are a way of grouping one or more statements
* What are parameters?
  + Parameters are placeholders for temporary values that get passed into a function for processing
* What are arguments?
  + Arguments are values you passed into the function when you call the function
* What is the difference between parameters and arguments?
  + Parameters are placeholders that allow you to pass values into a function. Arguments are the values that are being passed into parameters
* What are return values?
  + The output value of a function, which can be captured in a variable
* What is the difference between a global and a local variable?
  + A global variable is defined in the main body of the script and can be used anywhere in the script. A local variable is defined in the definition of a function and can only be used in that function. Global variables are often used to access function results whereas local variables are often used to return results.
* How do you use functions to organize your code?
  + You can group a set of operations that gets used repeatedly (such as print current file content, ask user for password) into a function, so that you can refer to it whenever you need to perform that set of operations
* What is the difference between a function and a class?
  + Functions are a way of grouping one or more statements, while classes are a way of grouping functions, variables and constants. Functions, variables and constants in a class can be called by running classname.functionname()
* How do functions help you program using the “Separations of Concerns" pattern?
  + Ideally, a function is built to do either grab data, process data and present/collect data, so the use of function also makes it easier to enforce separation of concern
* How are the debugging tools use in PyCharm?
  + You set a breakpoint, ask your computer to run the script step by step until the breakpoint, and check to see how the variables have changed and/or error messages to identify where the bug is.
* What is a GitHub webpage?
  + Github webpage is a page you can create in your Github repository to provide some instructions, some useful information and make your work more presentable etc.

Change task removal component

First off, reading the part of the code that asks user to remove an item from the list, I feel that asking users to copy and paste the entire task (which can get pretty long) to get it deleted is not user-friendly. Instead, I want it to show the users all the tasks in their list first, assign each of them an index, and then ask the user which one they want to delete by typing in its index. In order for this this to happen, the Processor.remove\_data\_from\_list function needs to take an index as input. I’m assuming strTask in the data section is used to hold the task to-be-removed, so I changed it to an integer named intIndex, which is a more suitable name and data type. For the same reason, I renamed IO.input\_task\_to\_remove to IO.input\_index\_to\_remove, and added a new function called IO.print\_current\_Tasks\_in\_list\_for\_removal that prints out the list of tasks in the file along with their indices, for user’s reference when they choose to delete a task.

Text

Description automatically generated

*Figure 1. changed strTask to intIndex*

Graphical user interface, application

Description automatically generated

*Figure 2. changed input\_task\_to\_remove to input\_index\_to\_remove*

Graphical user interface, text, application

Description automatically generated

*Figure 3. print\_current\_Tasks\_in\_list\_for\_removal*

Remove unnecessary function returns

In the starter file, every function in the Processor class returns the list\_of\_rows which is the list of dictionaries that hold the data. As I was working my way through the script, I found that since these functions are for data processing, when you finish calling these functions the list is already updated, so there’s no need to return list\_of\_rows for capturing in a variable. Thus, I remove all the list\_of\_rows from the function return.

Graphical user interface, application

Description automatically generated

*Figure 4. Remove list\_of\_rows from return*

Handle exceptions and update run status in functions

In all the functions in Processor class, the statements either runs successfully and the function returns a “Success” status or the program throws an error. I want to make sure the program handles errors in the function correctly and tells the user when a function didn’t go through. Then it will take user to the beginning of the loop in the main body of the script and ask them to select an option from the menu again. To do that, for each function in Processor class I wrapped the statements inside a try-except construct. Also, instead of always returning “Success”, I changed the codes so that it returns a local variable called run\_status. Run\_status is used to capture the status of the code run. When the run is successful, its value will be set to “Success”. When there’s an error, it would be set to “Fail” in the except statement. This local variable then gets returned to the global variable called strStatus for output.

Graphical user interface, text, application

Description automatically generated

*Figure 5. Exception handling and run status update*

Capture invalid menu selection

I want to make sure that the program prints an informational message when user enters an option that’s out of range of the menu (1-6), and send them back to the beginning and prompts them for entry again. So at the end of the loop in the main body of the script, I added an else block with code as follows.

Graphical user interface, text, application

Description automatically generated

*Figure 6. Else…*

Complete functions and main body of script

With those changes planned, I completed the functions with working statements and added docstrings for documentation and consistency. The contents of these functions are quite similar to what we did it Assignment 05, so I will skip over the details. I also filled in each code block in the loop in the main body of the script, but instead of writing statements that directly process data, I called functions whenever possible, and tested each code block separately to make sure it works smoothly, and see if there’s any update I need to make to the functions.

Text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

*Figure 7. Main body of script*

Results

Here’s what happens when you run the code on PyCharm. As shown in the screenshots, the program works smoothly and behaves as expected. The result, also attached below, shows that the list has been updated accordingly.

A picture containing text

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

Text

Description automatically generated

Table

Description automatically generated

*Figure 8. Running on PyCharm*

*Graphical user interface, text, application, chat or text message

Description automatically generated*

*Figure 9. ToDoFile.txt*

Summary

This assignment is a great exercise the helped me get familiarized with the concept of classes and functions, as well as how to use them effectively. Interestingly, although we were using a starter script with a lot of code written already, it actually took longer than the previous assignment because of the time spent trying to figure out what each segment is doing and how to make it work better. It represents a real-world challenge that programmers often face which is to work off of someone else’s completed or partially completed code. This shows us how important it is to document your code well, as there’s a high chance that someone else might need to revisit and update it and the process would be a lot less painful if it has a lot of useful comments.