Jason Verberne

November 15, 2022

Foundations of Programming: Python

Assignment05

# To Do List – Task and Priority

## Introduction

This paper will discuss the steps taken in completing Assignment05 of the Foundations of Programming: Python Course. This assignment required the modification of a Python script that would allow the user to select from a menu to show task and priority data, add a new task and priority to the list, remove an existing dictionary from the list, save the information to a text file, and exit the program.

## Background

Prior projects and knowledge that contributed to the successful completion of this assignment include:

1. Assignment01: This assignment requested the creation of a Python script that would accept the first and last name of a user and, in return, would display the concatenated information provided as the full name.
2. Assignment02: This assignment requested the creation of a Python script that would independently request two numbers from the user and, in return, would display the sum, difference, product, and quotient of those two numbers.
3. Assignment03: This assignment requested the creation of a Python script that would gather a household item from the user, gather an estimated value of the item, and save that information in text file.
4. Assignment04: This assignment requested the creation of a Python script that would allow the user to select from a menu to 1) gather multiple household items and estimated costs, 2) display the information gathered from the user back to the user, and save that information in text file and exit the program.
5. Independent study prior to the course, including:
   1. YouTube: <https://www.youtube.com/watch?v=rfscVS0vtbw>, freeCodeCamp.org, Mike Dane (presenting), uploaded July 11, 2018 (External Site)
   2. Pierce County Library Book: Begin to Code with Python, Miles, Robert S., Microsoft/Pearson Education, Inc., 2018, ISBN 9781509304523

I chose to perform my coding in PyCharm instead of using IDLE.

## Header

Before writing code that will actually perform the task, it is important to provide supporting information to any programmer, including myself for later use, about the purpose and historical activities of the code. This information does not have to be lengthy, although it could be if needed. In this case, I provided a description of what the code would do, then provided the Who, What, When, and Why of the changes.

-----------------------------------------------------------------------------------

*Text

Description automatically generated*

-----------------------------------------------------------------------------------

Figure 1: Header of Python code

Because this information is not intended to be processed by Python, I started each line with a “#” symbol. In Python, any line that begins with a number sign (aka the pound sign) will not be processed by Python. Alternatively, I could have used two sets of three single quotes (‘’’) with the header information between the sets, which would have provided the same result.

## Pseudocode

As programs become more complicated, it is often advantageous to write pseudocode. This is not actual programming code PyCharm can execute. Instead, this is a road map or outline to assist the programmer in structuring and understanding the actual code. The details of the coding are omitted. From the Assignment04 feedback, going forward, I will place the pseudocode along with the relevant actual code to assist in the writing of code and for others to understand the code that was written. In Assignment05, the pseudocode was already written as part of the template.



-----------------------------------------------------------------------------------

Figure 2: Pseudocode

## Code

The basic elements of this code include:

1. variables - the variables I used to store the data are:
   1. objFile - This is a variable that represents a file
   2. strData - this variable is used to hold the row of data from the text file as a list. In this case, the comma is used to separate the row into indexes.
   3. dicRow - this variable is used to represent the dictionary, which has “task” as the key and “priority” as the value.
   4. lstTable - this variable represents a list. In this assignment, it is a list of dictionaries, but could contain many types of values.
   5. strMenu - this variable is defined as a string but wasn’t used.
   6. strChoice – this variable is used to hold the input of the user as a string.
   7. strTask – this variable is used to hold the task input by the user.
   8. strPriority - this variable is used to hold the priority of the task input by the user.
   9. count – this variable will be used to hold a number which will be incremented by 1 as part of the program.
   10. strRemoveTask - this variable is used to hold the dictionary task the user wants to remove
   11. strChoice3 – this variable captures the user choice in option 3 to remove an item by a number or type the task (‘N’ or ‘T’)

Text

Description automatically generated

-----------------------------------------------------------------------------------

Figure 3: Data elements of the code.

1. print() - this function displays information to the screen
2. input() - this function allows the user to provide information to be stored in Python
3. while loop - The behavior can change depending on how it is structured, but, generally, the while loop will continue to recycle the code until a set condition is no longer met, allowing the program to exit the loop.
4. if/elif/else - these statements evaluate a programmer created statement as either True or False. If the statement evaluates to True, the follow on action created by the programmer is performed. If False, the action is skipped. The elif statement performs the same actions as the if statement given the if statement originally evaluated to False. The else statement will be executed if no other condition in the if or elif statements are met.
5. try/except – the try statements attempts to execute code, but if an error occurs, the code under the except statement is executed instead.

-----------------------------------------------------------------------------------

## Separation of Concern

A new concept was introduced this week called “Separation of Concern,” which is defined as, “In computer science, separation of concerns (SoC) is a design principle for separating a computer program into distinct sections, so that each section addresses a separate concern. A concern is a set of information that affects the code of a computer program.” <https://en.wikipedia.org/wiki/Separation_of_concerns>, 2019. In our Assignment05 template, we had several notes idenfying the separate areas.



*-----------------------------------------------------------------------------------*

Figure 4, Separations of Concern from the template

Text

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generated

*-----------------------------------------------------------------------------------*

Figure 5: Python code for interacting with user

Lines 23 - 94

1. Line 28: This line is the Separation of Concern, described above. It’s the second section identified. The first being “Data.”
2. Line 29-30: These lines are the pseudocode for the following section of code.
3. Line 31: This line defines the variable objFile as the variable to represent the opened text file and as “r” (read only).
4. Line 32-35: These lines loop through each row of data from the text file (from objFile), split the data into elements via the .split method (by comma), then insert those elements into the dictionary (dicRow) by identifying the element’s index number. The lstTable then uses the append method to add the newly created dictionary into the list.
5. Line 36: This line closes the opened text file via the close method.
6. Line 37: This line was left blank to visually separate the blocks of code, making them easier to read by the programmer.
7. Line 38: This line is the third Separation of Concern, described above, identified as the “Input/Output” section.
8. Line 39: This line is the pseudocode for the following section of code.
9. Line 40: This line creates the while loop used for the program. The user will loop through this code until a ‘break’ is achieved.
10. Line 41-48: This line provides the user with the list of menu options, repeatedly, while using the program.
11. Line 49: This line uses the input() function to get user input on which of the menu options will be executed.
12. Line 50: This line provides a visual blank space for the user of the program, not the programmer.
13. Line 51: This line is the pseudocode for the following section of code.
14. Lines 52-57: These lines use the if() statement to evaluate if the user entered a 1 for the strChoice variable. The if statement uses the strip() method to eliminate hidden spaces and \n character line. If True, the for/in loop goes through each element of the lstTable list and prints the task and priority from the dictionary. Lines 48 and 51 are for aesthetics for the program user. The continue statement will cause the program to begin the while (True) loop again.
15. Line 58: This line is the pseudocode for the following section of code.
16. Line 59-65: These lines use the elif() statement to evaluate if the user entered a 2 for the strChoice variable. If True, the input() function gathers the task and priority and stores the information in the strTask and strPriority variables. The dicRow variable then assigns the input from the user as a ‘task’ and ‘priority’ of a dictionary. The dictionary is then appended as a list item to the lstTable list. The program prints a message to the user confirming the data has been added. The continue statement will cause the program to begin the while (True) loop again.
17. Line 66: This line is the pseudocode for the following section of code.
18. Lines 67-98: These lines use the elif() statement to evaluate if the user entered a 3 for the strChoice variable. If True, an if/elif/else statement is used to determine if the user wishes to eliminate a task/priority dictionary by a numbered list (choosing the number) or by typing the task. If by number, the user provides the number information. The count variable is used to create a row number for the user and is incremented by 1 using the += option. The strRemoveTask variable uses the input() function to gather which line (identified by the count variable) the user wants to remove. The continue statement will cause the program to begin the while (True) loop again. If the user chooses to remove the item by typing the task, the strInput variable is used to allow the user to type the information. If the strInput variable matches a task in one of the dictionaries, then the del lstTable[] function is used to remove the index. An error handler is used with the count vs len(lstTable).
19. Line 99: This line is the pseudocode for the following section of code.
20. Lines 100-106: These lines use the elif() statement to evaluate if the user entered a 4 for the strChoice variable. If True, the variable objFile is used to open the ToDoList text file with a “w” (write) option. The for/in loop then writes each list item on the lstTable list to the text file by identifying the ‘task’ and ‘priority’ of each dictionary in the list. The program then confirms to the user that the list has been saved. The continue statement will cause the program to begin the while (True) loop again.
21. Line 107: This line is the pseudocode for the following section of code.
22. Lines 108-115: These lines use the elif() statement to evaluate if the user entered a 5 for the strChoice variable. If True, the input() function is used to gather the user’s verification of wanting to leave by entering a ‘Y’ or ‘N’ and saving that information to variable strChoice. Another if/else statement performs data validation that the user has entered a ‘Y’ or ‘N.’ The program uses the lower() method to make any entry lowercase so a lowercase ‘y’ would be acceptable. If the if() statement evaluates as True, the loop will ‘break’ and end the program. If False, the program will ‘continue’ and loop back to the while True loop. The print() options provide information to the user on what they have selected and if an invalid entry has occurred.
23. Line 116: Same as line 36
24. Line 117: This line uses the input() function to pause the program until the user hits the ‘Enter’ button.

## Challenges and learning

The syntax for the if and elif statements were challenging when trying to input the information from the text document, so I had to review the lesson a few times. Secondly, I referenced <https://www.geeksforgeeks.org/python-ways-to-remove-a-key-from-dictionary/> (external link), [manjeet\_04](https://auth.geeksforgeeks.org/user/manjeet_04/articles), updated July 07, 2022, to review ways to remove items from a dictionary, but for option 3 of the menu, I realized I was removing an element from a list, which is an index, not a key.

## Results

The results of the program were as expected, with the user selecting from a menu to 1) gather multiple household items and estimated costs, 2) display the information gathered from the user back to the user and save that information in text file and exit the program.

-----------------------------------------------------------------------------------

*C:\\_PythonClass\Assignment05>python Assignment05\_Starter.py*

*Menu of Options*

*1) Show current data*

*2) Add a new item.*

*3) Remove an existing item.*

*4) Save Data to File*

*5) Exit Program*

*Which option would you like to perform? [1 to 5] - 1*

*Task | Priority*

*===============*

*Write Python Code | 10*

*Complete Homework | 9*

*Eat breakfast | 5*

*Finish painting | 2*

*===============*

*Menu of Options*

*1) Show current data*

*2) Add a new item.*

*3) Remove an existing item.*

*4) Save Data to File*

*5) Exit Program*

*Which option would you like to perform? [1 to 5] - 2*

*Please enter the new task: Read a book*

*Please enter the task priority: 6*

*Your Task and Priority have been added*

*Menu of Options*

*1) Show current data*

*2) Add a new item.*

*3) Remove an existing item.*

*4) Save Data to File*

*5) Exit Program*

*Which option would you like to perform? [1 to 5] - 3*

*Do you want to remove an item by a [N]umber list or [T]yping the task?*

*please type 'N' or 'T': t*

*What task would you like to remove: eat breakfast*

*'Eat breakfast' was removed from the list.*

*Menu of Options*

*1) Show current data*

*2) Add a new item.*

*3) Remove an existing item.*

*4) Save Data to File*

*5) Exit Program*

*Which option would you like to perform? [1 to 5] - 4*

*Your list has been saved*

*Menu of Options*

*1) Show current data*

*2) Add a new item.*

*3) Remove an existing item.*

*4) Save Data to File*

*5) Exit Program*

*Which option would you like to perform? [1 to 5] - 5*

*Are you sure you want to exit? Enter 'Y' or 'N': y*

*\*\* You have exited the program \*\**

*Prese 'Enter' to continue.*

-----------------------------------------------------------------------------------

Figure 5: Results after running the Python code and user providing input.

Graphical user interface, text, application

Description automatically generated

-----------------------------------------------------------------------------------

Figure 6: Saved Text File of the User Input

## Summary

This document described the steps taken and approach to completing Assignment05 of the Foundations of Programming: Python Course. It discussed the prior knowledge and background information considered when writing this script, and then the major sections of the script. Finally, it discussed the returned results the user would expect.