Kevin LaJudice

2021-Aug-10

Foundations of Programming – Python

Assignment 05

Assignment 05 Write-Up

Introduction

In Assignment 05, we were tasked with modifying starter code that was an example solution for Assignment 04 in the previous module. The idea was to incorporate what we learned about dictionaries during module 5 as well as add some functionalities to the code like loading data from a file and deleting data.

Topic 1 – Replace list of lists with list of dictionaries

The first TODO involved replacing the list of lists with a list of dictionaries. In terms of lines of code written, this was pretty short and straightforward. I simply changed the “lstRow = []” code in line 14 to “dicRow = {}” (Figure 1) by modifying the variable name and using curly brackets instead of square brackets. This is fine since we’re starting with an empty dictionary that will get populated by the user as they user the program and select option “a”. I also parsed the code for other occurrences of the previous “lstRow” variable that needed to be updated to the “dicRow” variable by clicking on the the text in line 37 to highlight all the places where it occurs (Figure 2) and then go in and change them manually. In line 47, I also rewrote the list into a dictionary format (Figure 3).



Figure 1 – Allowing user to add data to the list

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

Description automatically generated*

Figure 2 – Highlighting all occurrences of “lstRow” variable to be changed to “dicRow” variable



Figure 3 – Re-writing list into dictionary format

Topic 2 – Add functionality of loading existing data

To accomplish the functionality of loading existing data, I copied in code that I learned in the module 5 material about reading and writing data to and from .txt files. The corresponding script is shown in Figure 4. The variable “strFileName” was defined in the #Data section of the code with a staring value of “CDInventory.txt” (Figure 5).

Text

Description automatically generated

Figure 4 – Script to read data from .txt file

**

Figure 5 – Defining the variable “strFileName”

Topic 3 – Add functionality of deleting an entry

To delete an entry, I declared the “deleteRow” and included an option for the user to indicate which row (CD entry) they want to delete (Figure 6). The script also converts their typed input into an integer to help with the following line of code.



*Figure 6*

I then made sure to subtract 1 from the variable integer (Figure 7) to get the proper index since the CD that the user would’ve saved under the ID “1” would’ve been the first row, and therefore actually the [0] value to the computer.

**

*Figure 7*

Summary

This homework assignment incorporated some of the new skills we learned in module 5 – namely using dictionaries, reading data from a file, and deleting data from a file.

Appendix A, below, includes screenshots of all the code working in Spyder and Terminal.

APPENDIX : Screenshots of code running in Spyder and Terminal

Text

Description automatically generated

Figure 1 – Code running in Spyder – option “[a] Add CD” has been completed successfully

*Text

Description automatically generated*

Figure 2 – Code running in Spyder – option “[i] Display Current Inventory” has been completed successfully

*A screenshot of a computer

Description automatically generated with medium confidence*Figure 3 – Code running in Spyder – option “[s] Save Inventory to File” has been completed successfully. Left of screen is .txt file. Middle of screen is program running in Spyder. Right of screen shows the file inside the Assignment05 folder.

*Text

Description automatically generated*

Figure 4 – Code running in Spyder – option “[l] Load Inventory from File” has been completed successfully.

Text

Description automatically generated

*Figure 5 – Code running in Spyder – option “[d] Delete CD from Inventory” has been completed successfully*

Text

Description automatically generated

*Figure 6 – Code running in Spyder – option “[x] exit” has been completed successfully*

Graphical user interface, application, Teams

Description automatically generated

Figure 7 – Code running in Terminal – options [a], [i], [s], all shown successfully running here. The middle of the screen is the .txt file opened to show its contents. The right side of the screen shows the file added into the Assignment05 folder.

Text

Description automatically generated

*Figure 8 - Code running in Terminal – option “[l] load Inventory from File” shown successfully running.*

Text

Description automatically generated

*Figure 9 - Code running in Terminal – option “[d] delete CD from inventory” and option “[x] exit” both shown successfully running.*

APPENDIX B: Writeup for Lab05\_A – Working with Files and Lists

In the module’s first lab, we were asked to complete the TODOs from the starter code. The first TODO in line 28 asked for code that would prompt the user to input data to store in a list. Lines 29-34 in Figure 1 show the associated code.

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

Description automatically generated

Figure 1 – Allowing user to add data to the list

The second TODO item (Figure 2) involved making writing the data to a file. This step was a good reminder about the importance of proper indentation since I kept getting the program to iterate over the data in ways other than what I wanted until I realized that my indentation was off.Text

Description automatically generated

Figure 2 – Writing data to a file

*A picture containing rectangle

Description automatically generated*

Figure 2B – Sample data in “Lab05\_A.txt” file

The third TODO item involved reading data from the file and printing it in the Spyder console. Lines 48-56 (Figure 3) show the associated code. Figure 3B shows the printout in the Spyder console.

*Text

Description automatically generated*

Figure 3 – Read data from the “Lab05\_A.txt” file

*Graphical user interface, text

Description automatically generated*

Figure 3B – Data printout in Spyder console

For the fourth and final TODO item, we needed to allow the program to display data entered by the user. Lines 57-63 (Figure 4) show the code for this and Figure 4B shows the printout in the Spyder console.

*A screenshot of a computer

Description automatically generated with medium confidence*

Figure 4 – Code to display user data

*A screenshot of a computer

Description automatically generated with low confidence*

Figure 4B – Console display of sample data