Emily Martin

02/27/2022

IT FDN 110

Assignment 05

# Introduction

The coursework for Module 04 introduces the concepts of \*\*\*. Module 05 also iterates \*\*\*. In this assignment, I used the knowledge gained in Module 05 to produce a program that allows the

user to enter CD data, view the current inventory, load data from a file, delete an entry in the program’s memory, save data to a CDInventory.txt data file, and exit the program.

# Approach

The following steps document the general approach taken to produce the source code referenced in the Appendix, as well as the reasoning behind a selection of notable syntactical decisions:

1. Based on the course recommendation for IDEs, I created a Python file in Spyder to write my code in. I added a header to capture administrative details.
2. Next, I modified the given code example to use dictionaries instead lists. These modifications were fairly straightforward, and I worked my way down the code line by line to make the necessary adjustments. I stopped to check that existing functionality was fully operational prior to moving forward.
3. I then moved on to adding code necessary to enable the functionality of loading data from file into memory. I began by referencing the materials in the learning module as a code structure starting point. I utilized the row.strip() and .split() methods introduced in Module 05 to take in the string data present in the file, remove the white space, and separate the values into a new list of strings. I then indexed into that list object by object and assigned keys to the indexed values to create a dictionary. I then appended each dictionary to my list of dictionaries.
4. After that, I addressed the ‘delete’ menu option. To delete an entry, I first instruct the user to input a known CD ID integer and set an index counter variable to 0. Next, I opened a for loop that examines each row, looking for the value assigned to the key ‘id’ that matches the user’s input. When a match is found, the CD list is indexed at the counter variable location and the dictionary is deleted. Otherwise, if the CD ID is not located in the list, the user is informed that the CD cannot be found.

# Summary

After testing this code by adding multiple CD entries, I was able to verify that CDInventory.txt was updated after every user entry. Additionally, I verified that I was able to load data in from the file, as well as delete dictionary entries from my list stored in memory. In summary, this assignment required knowledge from both current and previous modules to construct a solution to the required specifications – particularly loops, indexes, and writing to files.

# Outcome

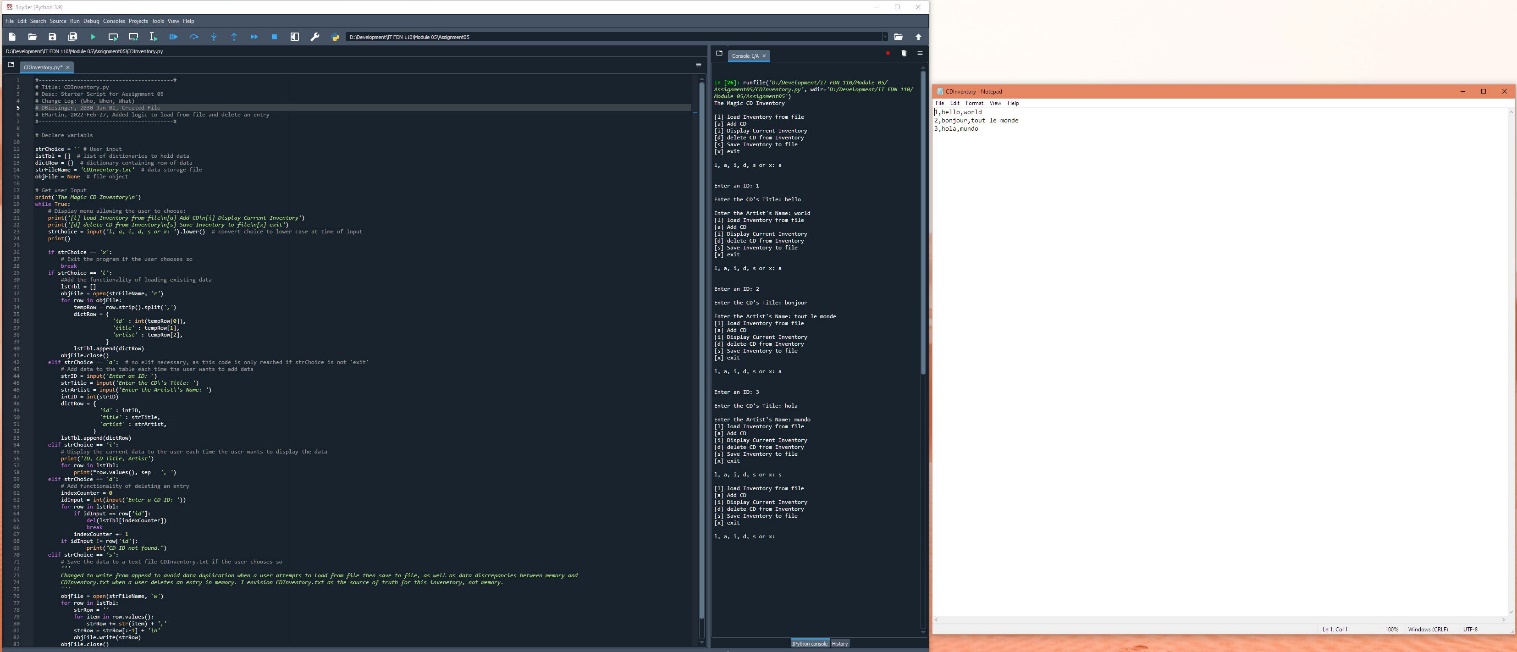


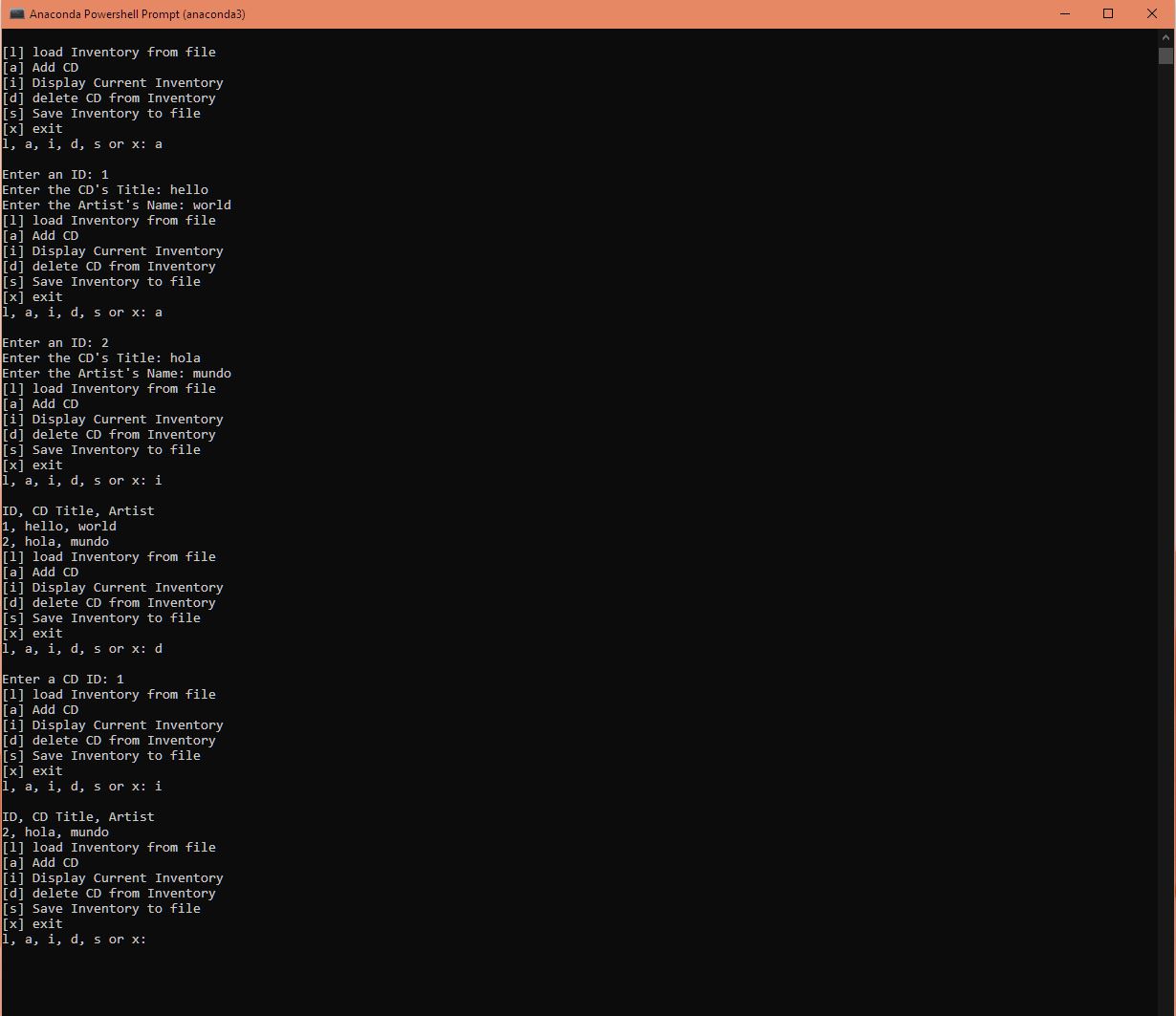
Figure : Capture of code running in Spyder

Figure : Capture of code running in a prompt