Emily Martin

03/06/2022

IT FDN 110

Assignment 06

# Introduction

The coursework for Module 06 introduces the concept of classes, static methods, and functions. Module 06 also iterates on previously introduced concepts of variable scope, as well as code commenting. In this assignment, I used the knowledge gained in Module 06 to streamline 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. First, I reviewed all of the to-dos in the existing code, and created a separate list of items to track. This allowed me to address each to-do more strategically, and enabled me to add notes to the list to capture additional follow up items.
3. I decided to start with Class FileProcessor, so I ported the logic in the code block into the static method write\_file. After doing that, I removed that logic from the main loop and added a call to my newly created function. This wrapped up my work on the FileProcessor class.
4. I then moved onto addressing the to-dos for the DataProcessor class. I separated this work into the add CD functionality, and the delete CD functionality. For each, I repeated the process of moving the logic into newly defined static methods add\_cd and delete\_cd. I then called those functions from the main loop, and checked to ensure that everything worked as expected.
5. Next, I addressed the IO class. I created three new static methods to handle CD addition, CD deletion, and CD data loading from file into memory. Like the steps above, after defining the functions and moving the logic into the functions, I then called those functions from the respective lines of the main loop. After each change, I ran the code to ensure that it functioned as intended.
6. As a last step, I surveyed all of the new static method functions I created during my work on this assignment and added docstrings to explain how each function was intended to perform, including specifying arguments and returns.
7. Upon completion, I uploaded my assignment to GitHub for review.

# Summary

After testing this code by adding multiple CD entries, I was able to verify that CDInventory.txt was updated after every user entry – now using static method functions containing the logic, and called from the main loop. 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 streamline the program.

# Outcome

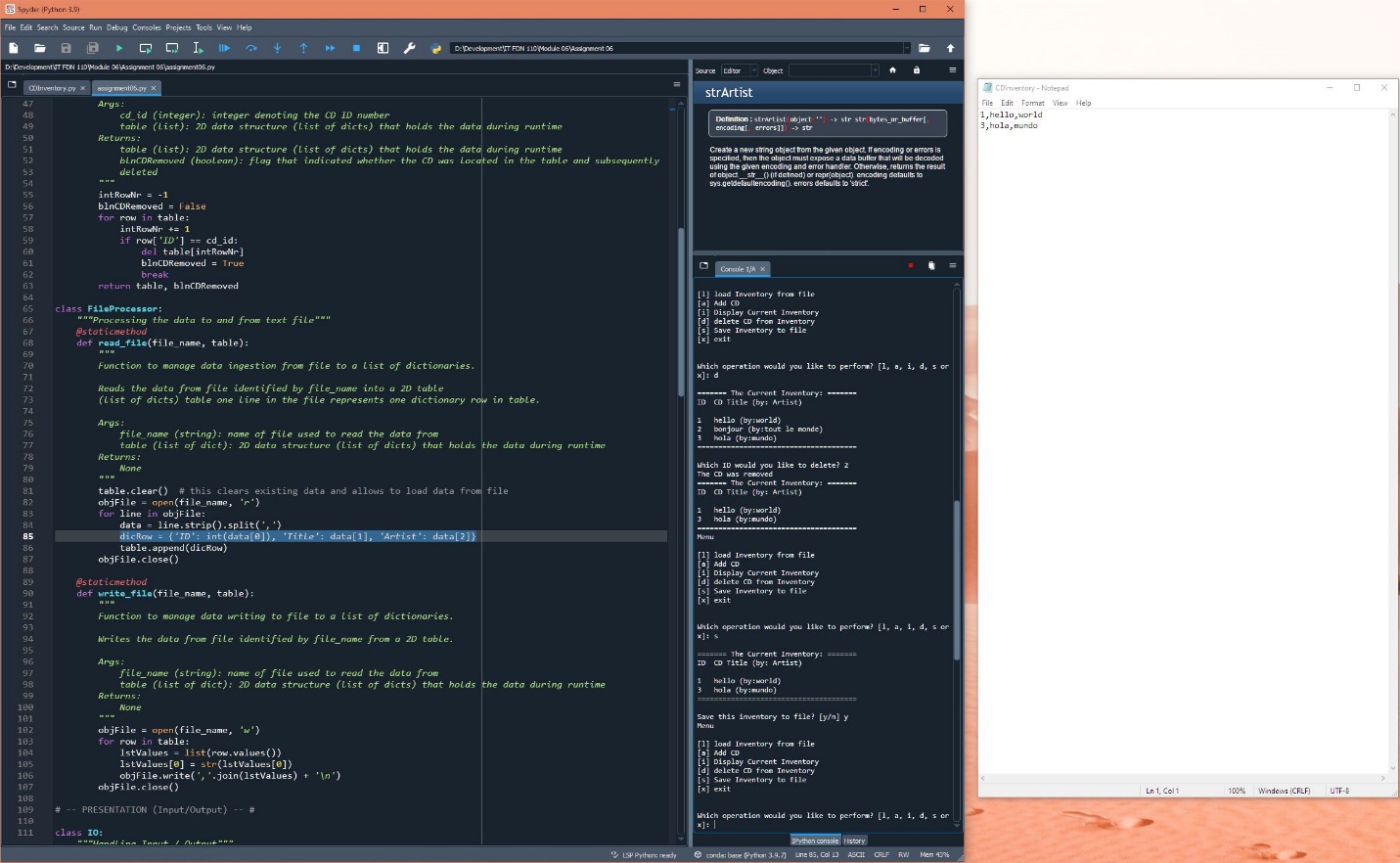


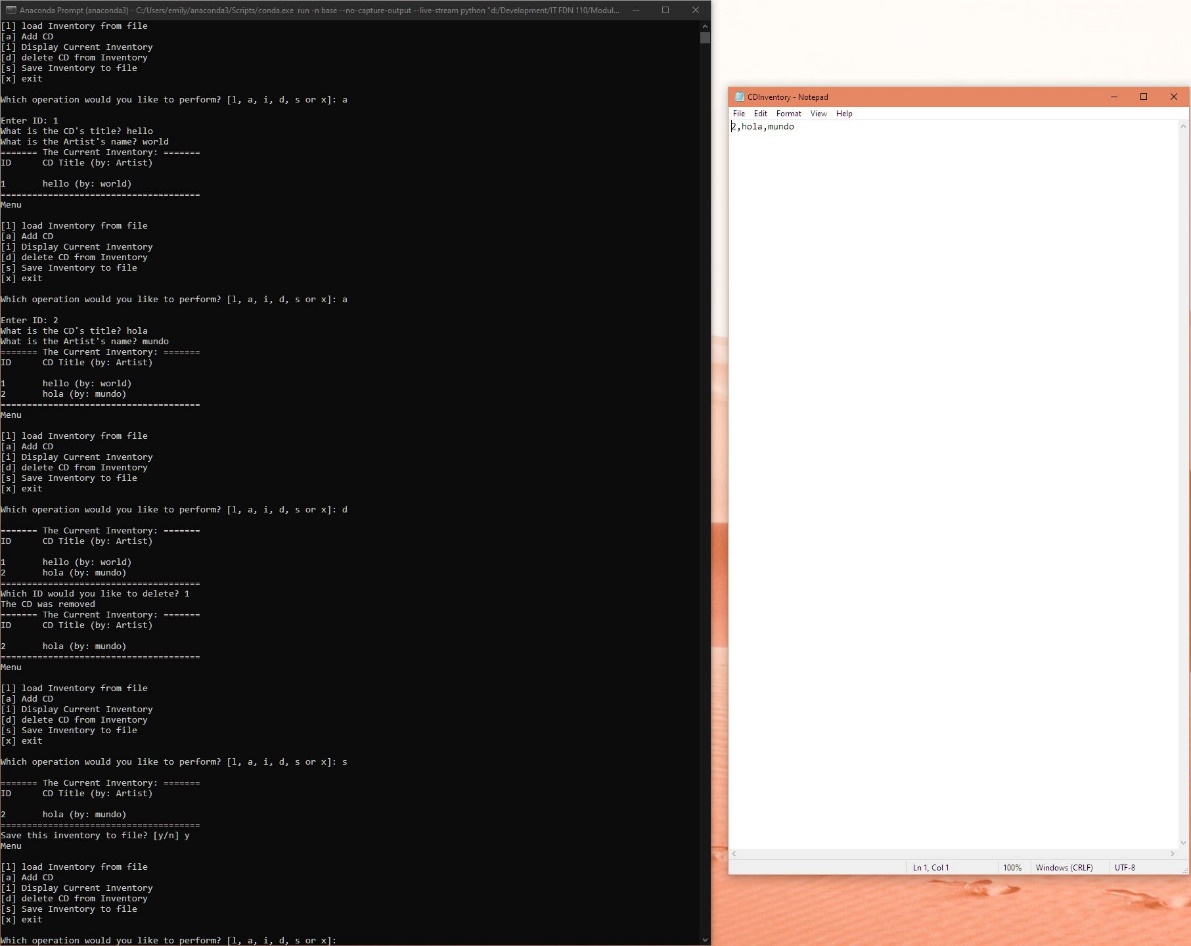
Figure 1: Capture of code running in Spyder

Figure 2: Capture of code running in a prompt