Arianna Morgan

November 20, 2022

Foundations of Programming: Python

Assignment 06

Classes and Functions

Introduction

In this module, we learned about classes and functions. Functions are segments of code that are grouped together and given a function name. They are defined first and then the code in a function is executed later when the function is called. They can have zero or many parameters (also referred to as arguments), which are data that are passed into the function and used for processing. Classes are groups of functions. Using classes helps create more clear organization of the code. This week, we improved upon last week’s script to better adhere to the separation of concerns design principle, separating our code into Data, Processing, and Presentation sections. We did this by creating functions, separated into classes, which accomplish tasks that fit clearly into the Processing or Presentation sections. We moved code from the main body of the code into separate functions and then called those new functions within the main body. This made each task easily identifiable in the code and made the main body more streamlined.

Details

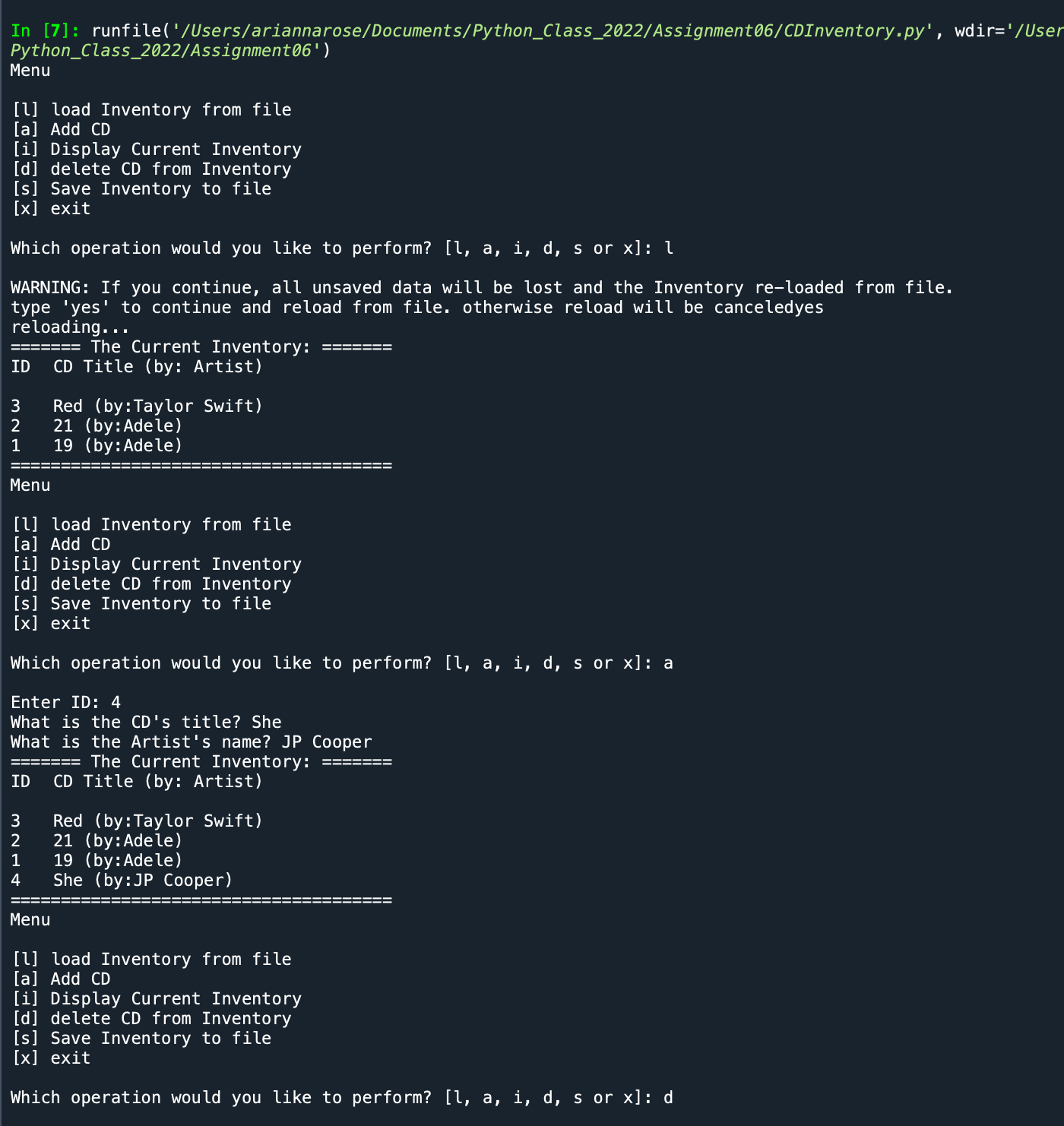
The first task was organizing the code that allows the user to add a new CD to the inventory. I created a new function under the IO class called add\_inventory\_menu and moved the code that asks for the new CD information from the main body into that new function. My function returns the user input (strID, strTitle, strArtist). When I call this function in the main body I assign it to those variables so that the user input values are available in the main body of the program. To process that data (i.e. to add it to the inventory table), I created a new function in the DataProcessor class called add\_inventory. I passed strID, strTitle, and strArtist into this function as parameters and moved the code that processes those inputs and adds them to the inventory into that new function. Lastly, I updated the main body code to call this function.

The next task was to clean up the section that allows the user to delete a CD from the inventory. I created another new function under the DataProcessor class, this one called del\_inventory. I moved the processing code from this section into my new function and called my function in the main code. My parameter for this function is the user’s selected ID to delete, intIDDel.

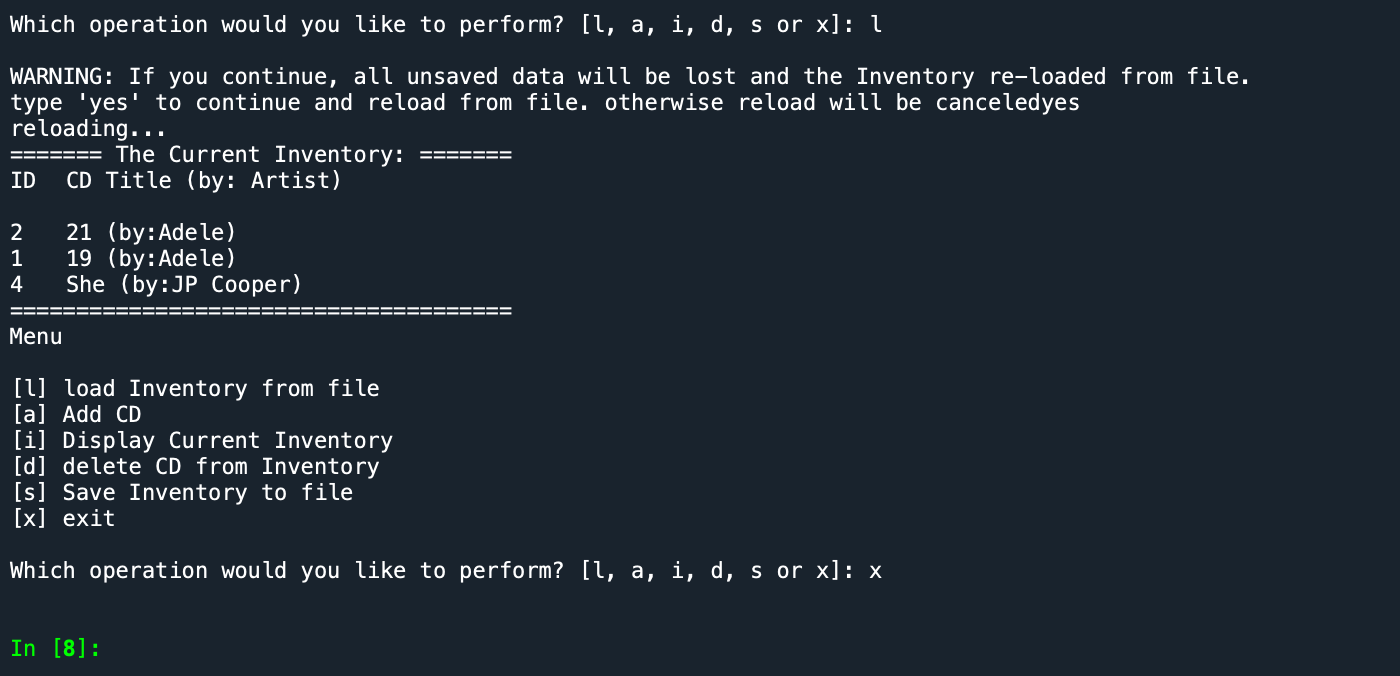
The last task was to update the section that saves the data back to the text file. The necessary function, write\_file, was already created under the FileProcessor class: I just needed to move the relevant code into this function and update the variables to match the already-included parameter names.

Below are screenshots of my code working:

In Spyder:



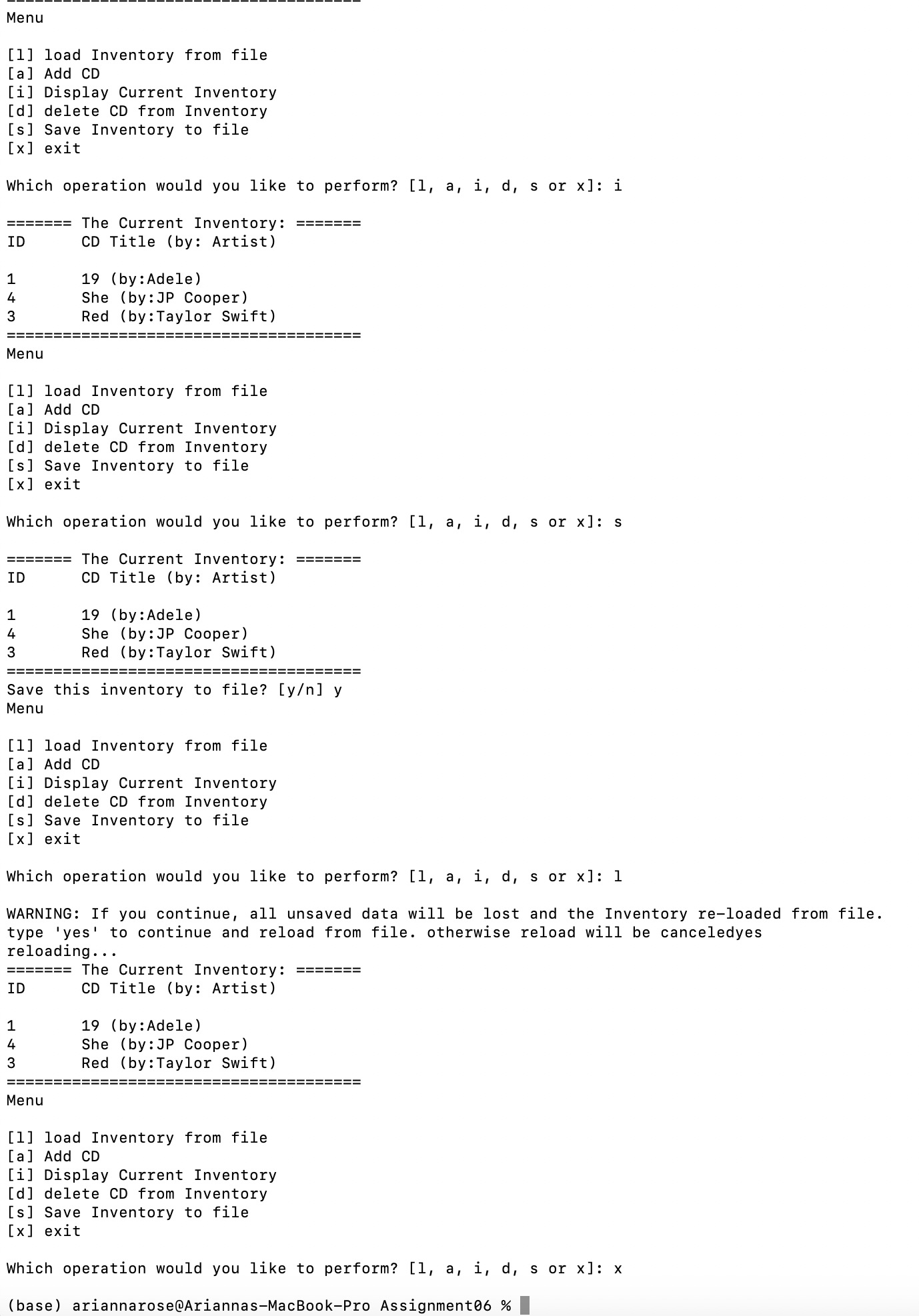




In Terminal:

Table

Description automatically generated with medium confidence



Summary

In this assignment, we again had the opportunity to work with existing code that we didn’t write and add value to it. I am enjoying the increase in the complexity of the code that we are working with each week. It is very useful to have practice creating code using best practices for readability. My previous experience learning Python did not deal with this very much, so I am learning a lot that I can apply to the code I am creating at work for process-automation tools.