Jesse Kim

11/20/2022

UW PCE – IT FDN 110

Assignment 06

Assignment 6 - CD Inventory with Classes and Functions

## Introduction

This week’s lesson and modules introduced the concept of functions and classes as well as best practices for working with and passing arguments and parameters. We also learned how to use doc strings to provide comments on what the function does and describe the arguments passed and value(s) returned, if there are any.

## Assignment 6

This week’s assignment covered the previous code where we updated our script to allow a user to access and edit a list of stored compact discs (CDs) as well as write the data to a file. Users should have the options to load data, see inventory, add new inventory, and save the inventory to a file.

### Creating the program

I pulled in some of my code from last week to the starter file and fixed a couple of things along the way that I wasn’t able to figure out last week.

### Adding Functions to the Data Processing Class

First I added a function to append a new dictionary row to the list of dictionaries used to store the values while the program was running and set those values equal to what would be provided by the user inputs from a related IO Class function.

Then I added a function to delete a row if the ID input by a user matched the value from the ID input by the user later in the loop. I used the enumerate function to iterate through the list of items in the list of dictionaries and if the user input ID matched the value for the ‘ID’ key, it would use that to remove that dictionary row from the list.

### Adding Functions to the File Processing Class

After the read function, I added a function to save the data to a text file, CD Inventory.txt, by iterating through the values stored in the list of dictionaries. I assigned each row value in the dictionary to a new list value, setting the zero index value for ‘ID’ to a string, then joined them and wrote them to the file.

### Adding Function for Input to the IO Class

I added a user input function that I would call later for adding a CD to the inventory, the id, album, and artist were requested from the user and returned.

### Updating the TODOs in the While loop

3.3.1 - I moved the existing input code into the user\_input method that would ask the user to input that data and called that method from the IO class in the step to add a new row in line 29 so there was nothing to add here.

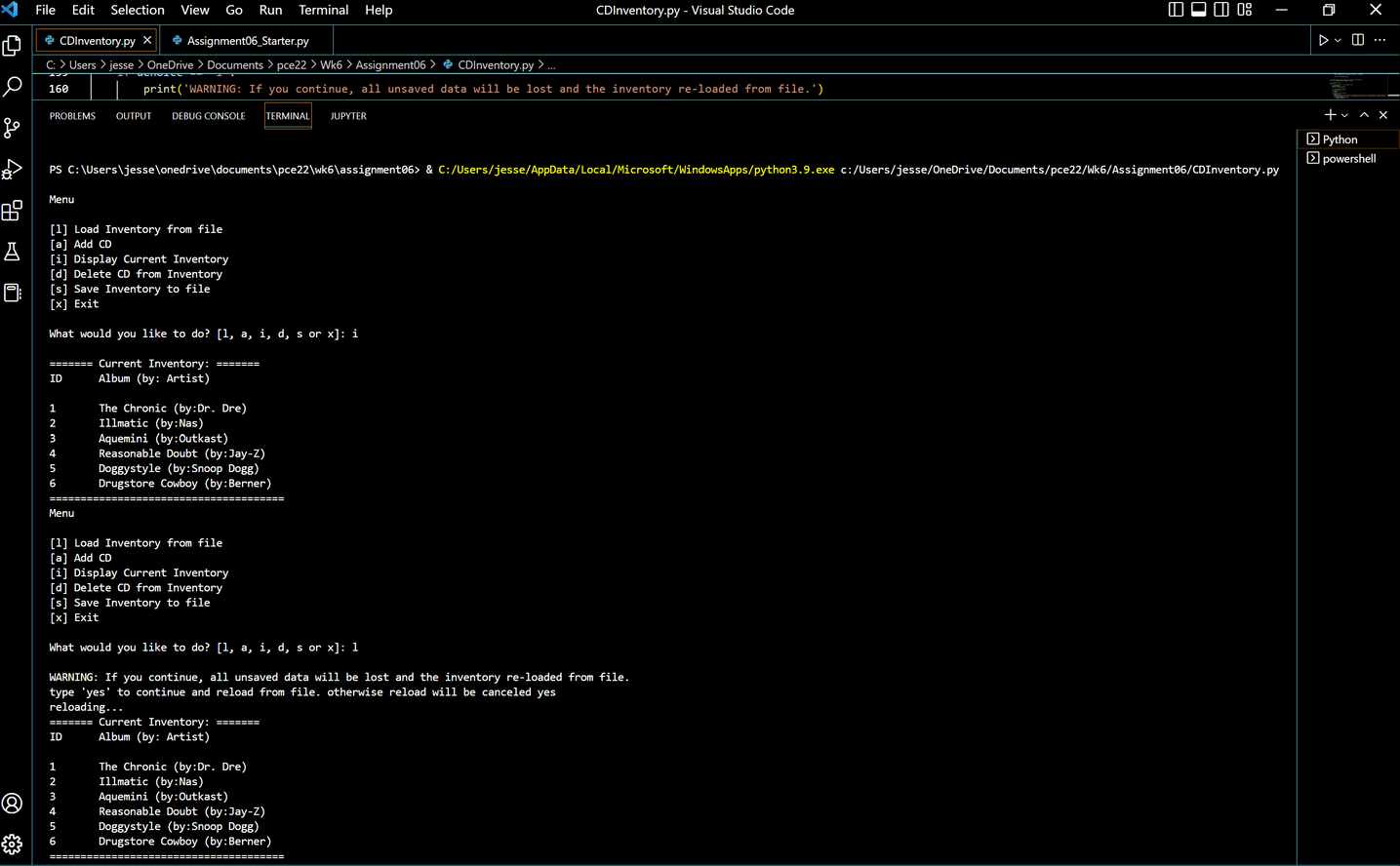
3.3.2 - I moved the existing code to the new\_row method I created in theData Processor class and called it from there.

3.5.2 - I called the cd\_delete method I created in the Data processing class where it takes the ID to delete that is input by the user (deliD) and iterates through the list until it finds the matching value then removes that row.

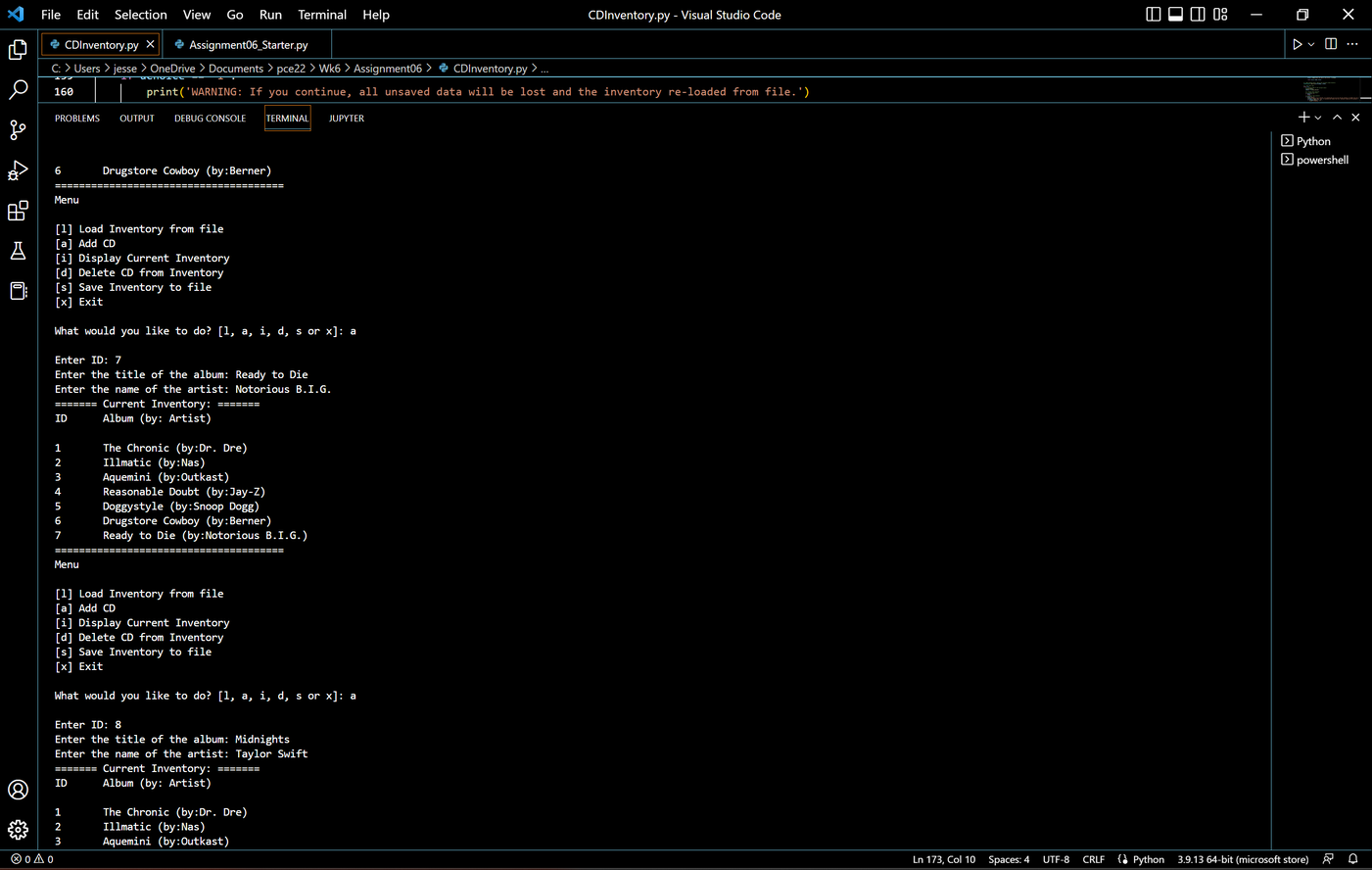
3.6.2 - I moved this code to a method in the File Processor class called save\_inventory and called it from there.

### Testing in the VS Code Terminal

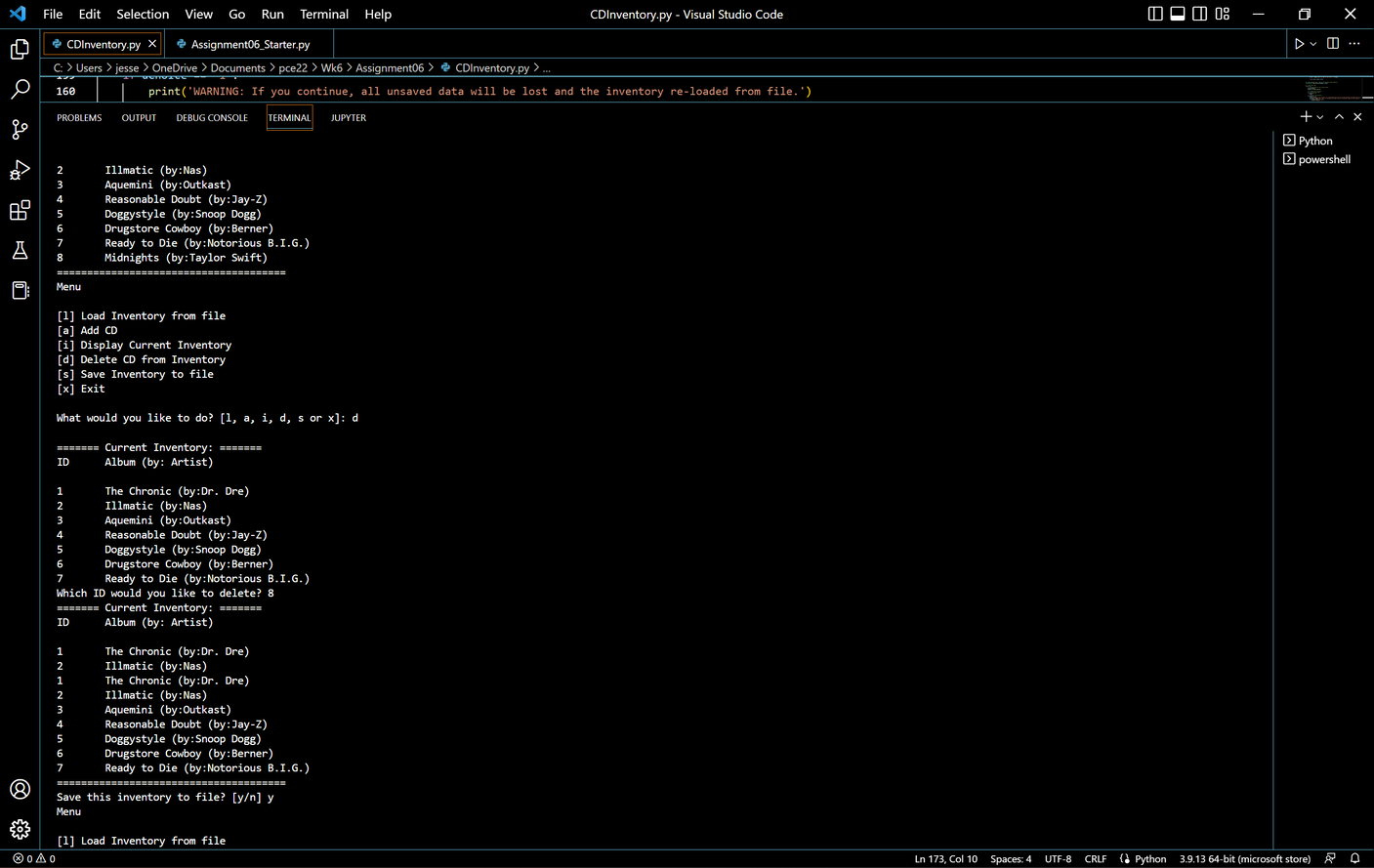
#### *Loading the program checking inventory and loading inventory from the file*



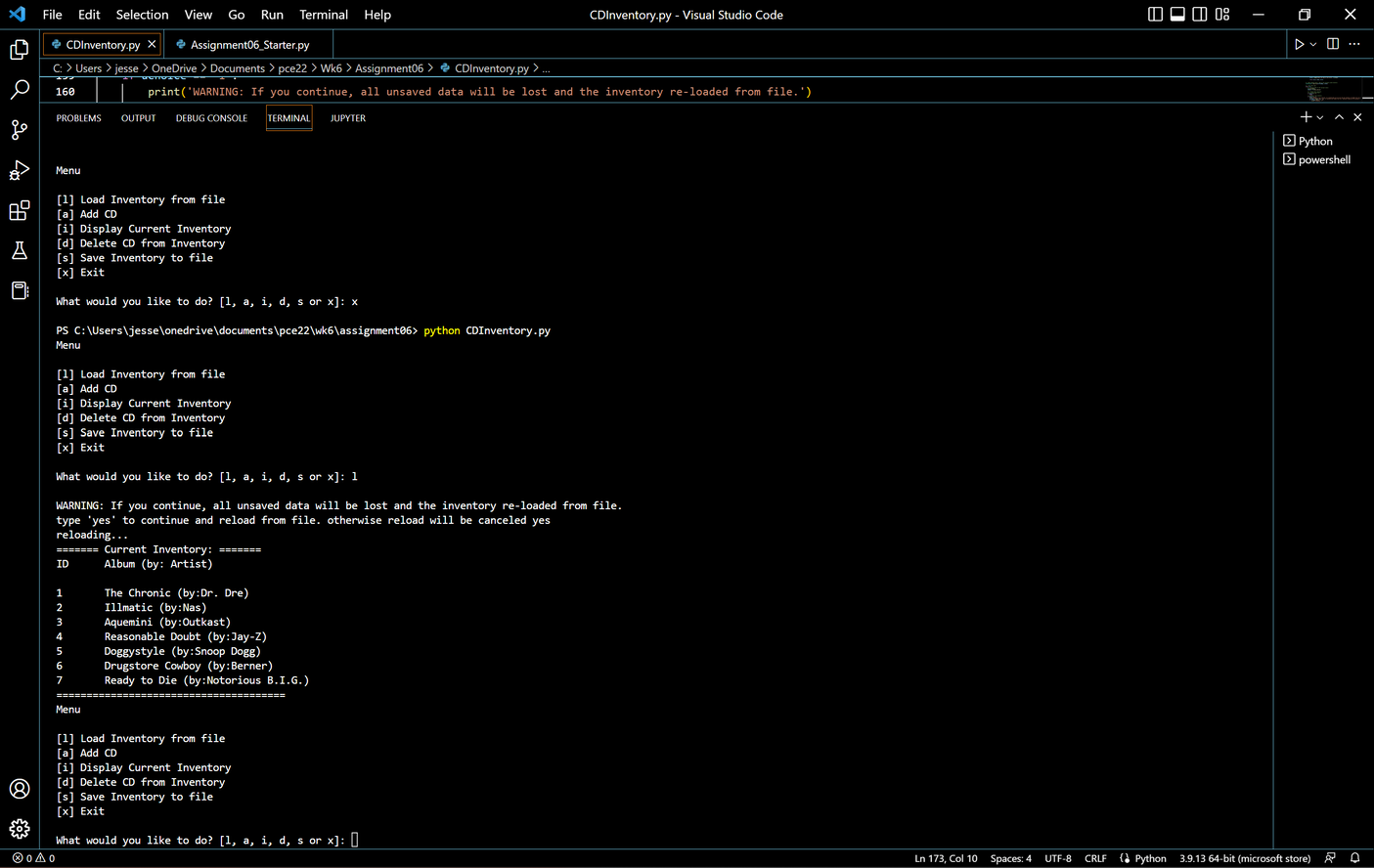
#### *Adding a couple of new albums*



*Whoops, wrong list...delete ID 8 from the list, then save the file*



#### *Exit the program then load again and see if the latest changes were saved to the file*



## Summary

I liked this week’s exercise a lot better than last week. It was nice to have a bit of time to fix my code from last week. I don’t exactly know why but taking the code elements out of the while loop and embedding them into classes and functions made more sense. Although, relating them back to the arguments took me a little while to understand and fix while debugging the errors in my code. This also made the actual running loop seem a lot less intimidating than last week.