Jesse Kim

12/4/2022

UW PCE – IT FDN 110

Assignment 08

Assignment 8 - Creating Public and Private Methods to Handle CD Inventory

## Introduction

This week’s lesson and modules covered the use of creating methods, public and private, passing attritubtes, and creating objects within Classes. We also learned how to use and create decorators, constructors, properties, and destructors when working with methods as well as the keyword ‘self’.

## Starting with the Starter Code

I opened the starter code then brought in elements of my code from previous assignments. Then I tweaked those elements to ensure it was following the correct structure from the lessons from this week. Initializing the objects, and ensuring each method and property had the correct reference to ‘self’.

## Data Processor Class

I decided to keep the Data Processor class from previous weeks and update my code to reflect the method formatting learned this week. It made more sense to me that those were actions that made more sense in their own class rather than moving them to the main while loop below. As noted in the previous assignment, I felt this made it much cleaner and easier to code that loop and keep these as methods that could be called on.

## FileIO Class

Similar to previous week’s code, I updated the arguments, or attributes, passed through the method in addition to ‘self’. I did find it interesting that the write function would only work if the iterable rows were written as strings.

For the load inventory method, I also wrote in some exception code to specify whether there is nothing saved to the file.

IO Class

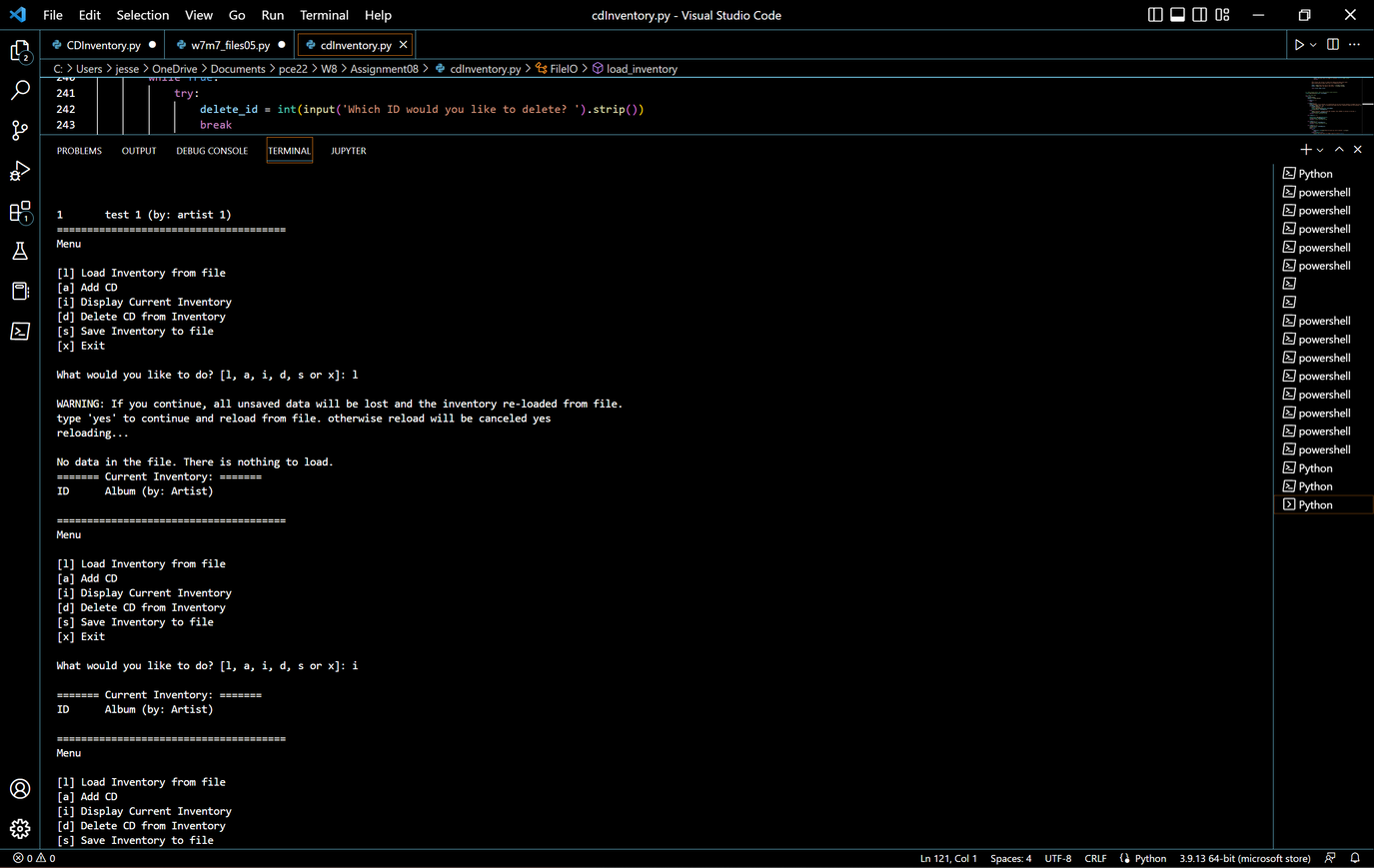
The IO Class was pretty straightforward without many changes from previous week’s code. Mostly I added the self reference to eatch method without changing too much of how the code worked.

Main Body

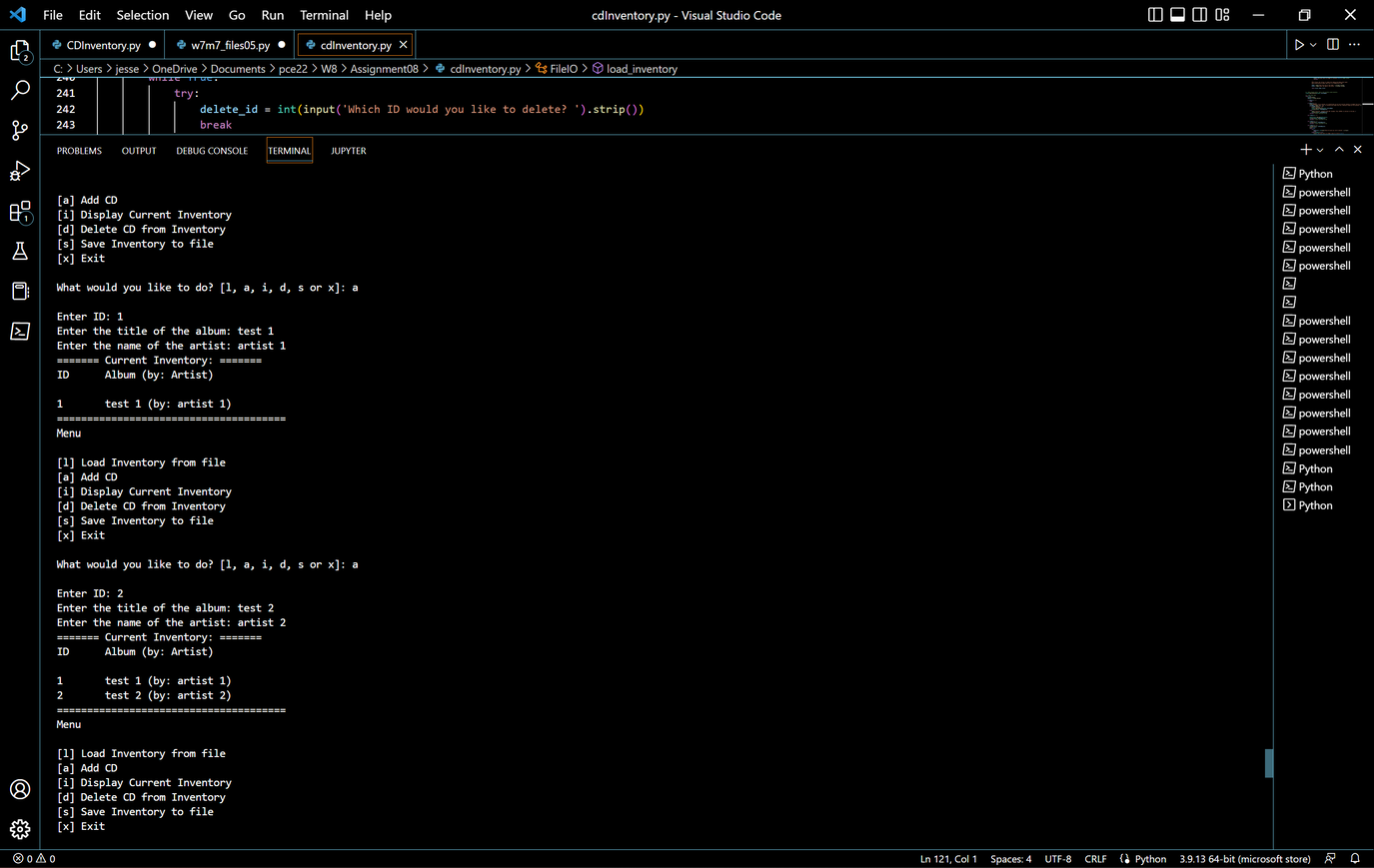
Here I updated each call to the class and method to ensure the arguments passed were correct and matched the named methods above. For the most part I was able to use a lot of the previously submitted code, but this was a lot cleaner as much of the functionality is contained within the class methods. I added some lines to account for exceptions then proceeded to testing my code.

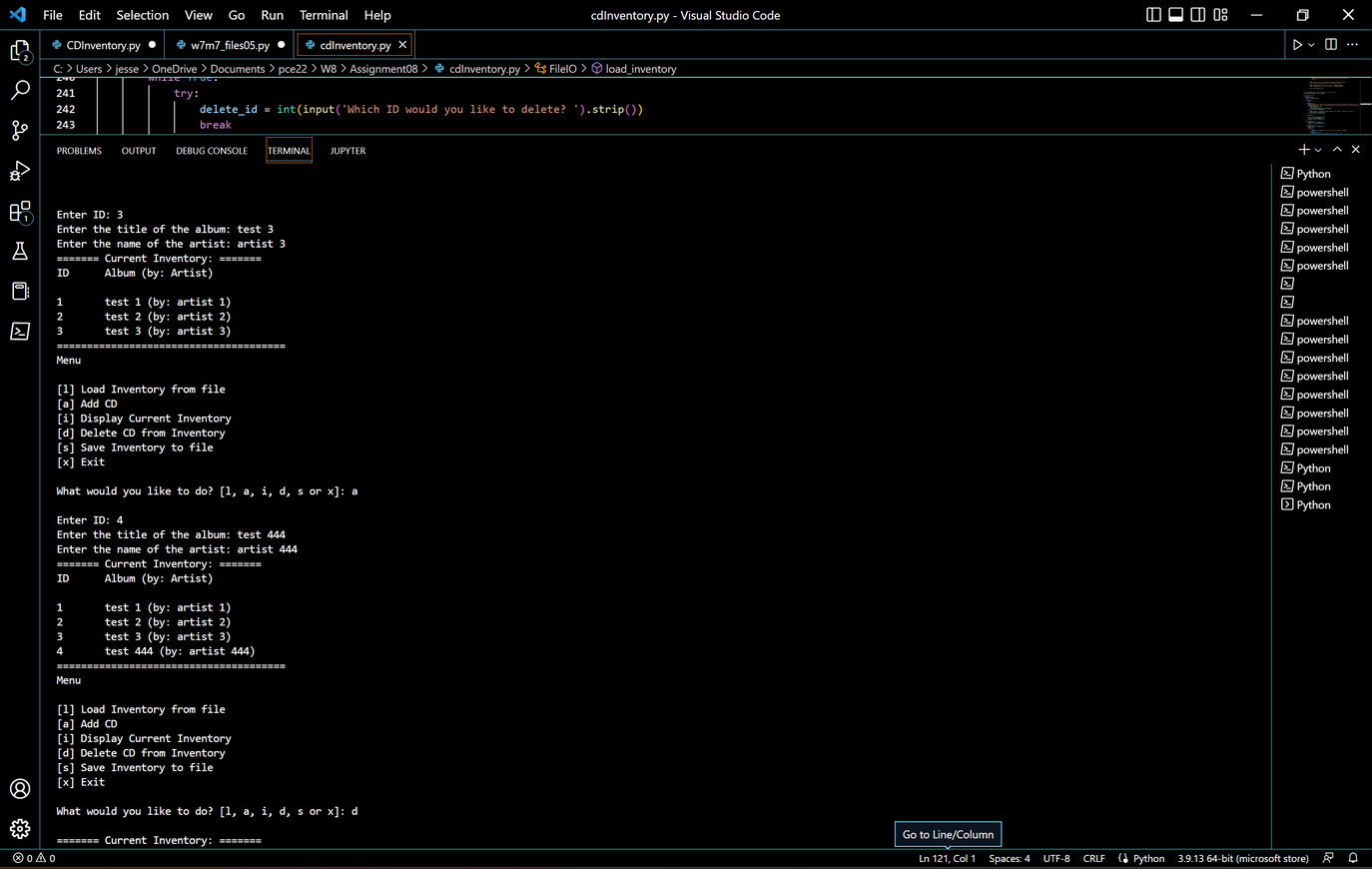
### Testing in the VS Code Terminal

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

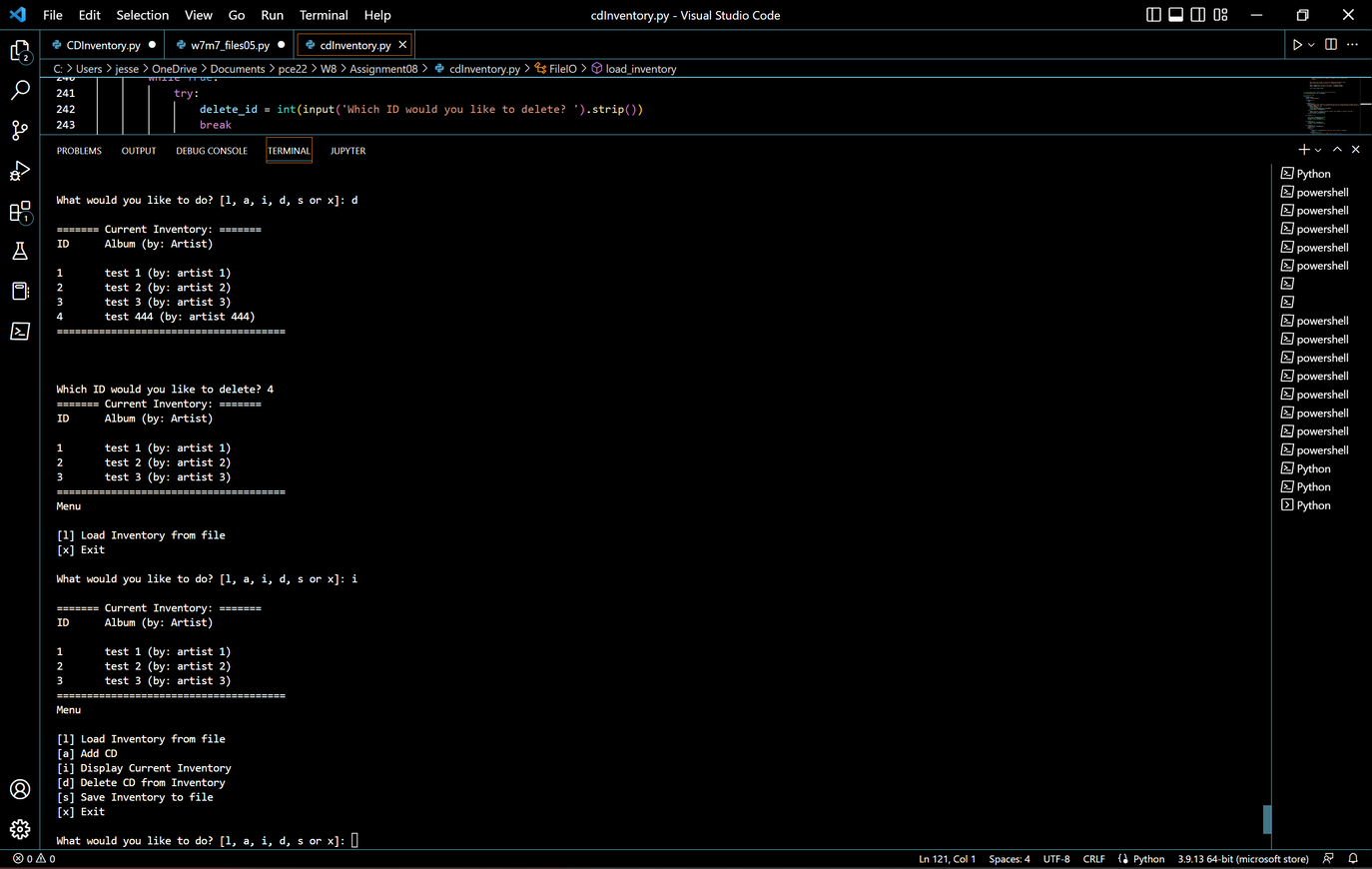


#### *Adding a couple of new albums*

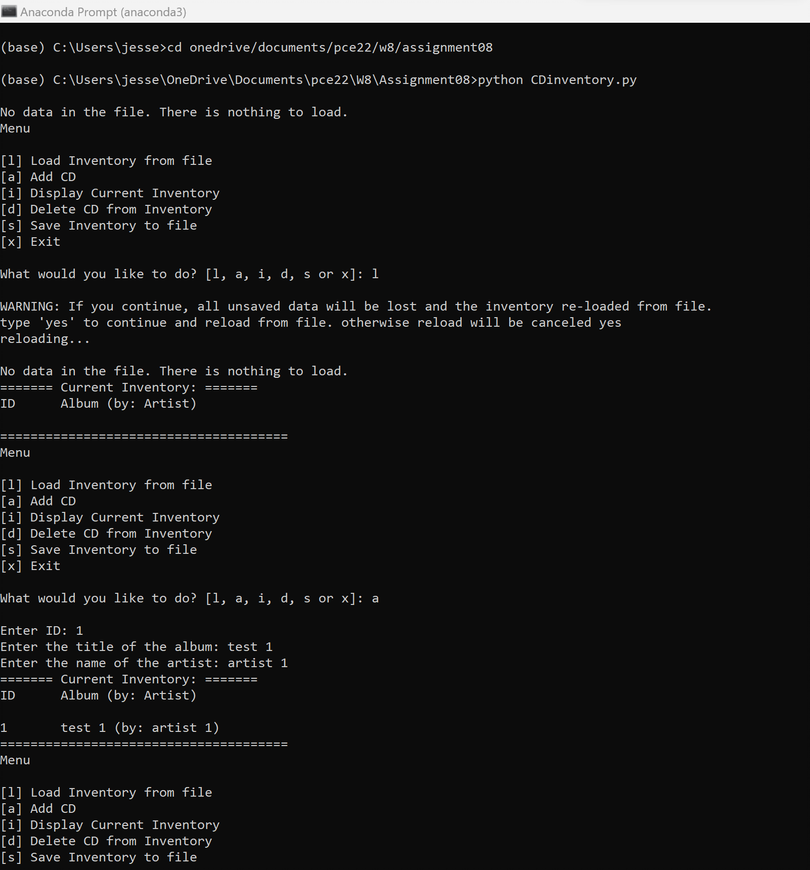


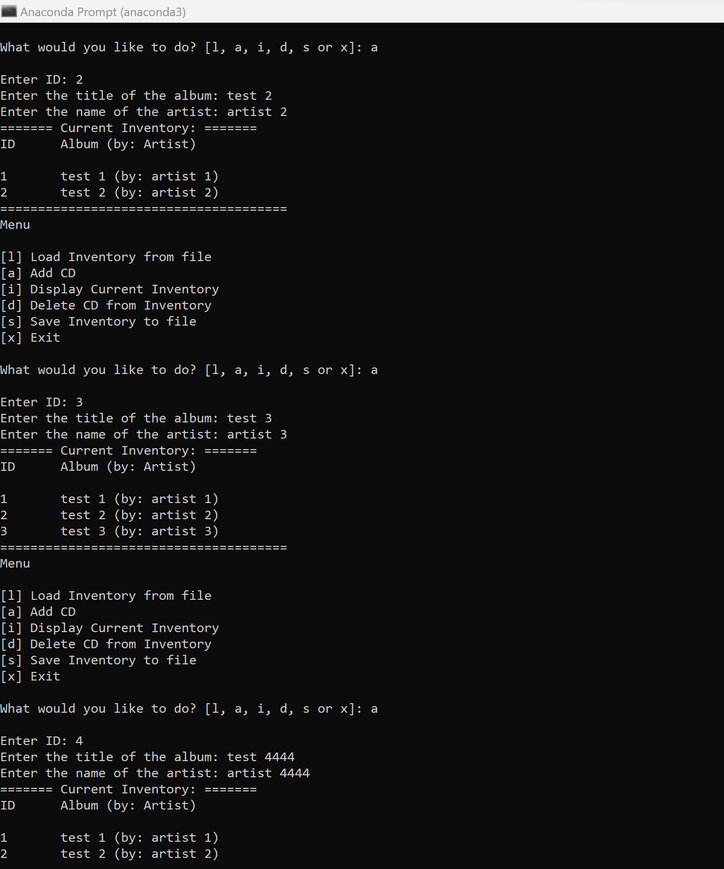


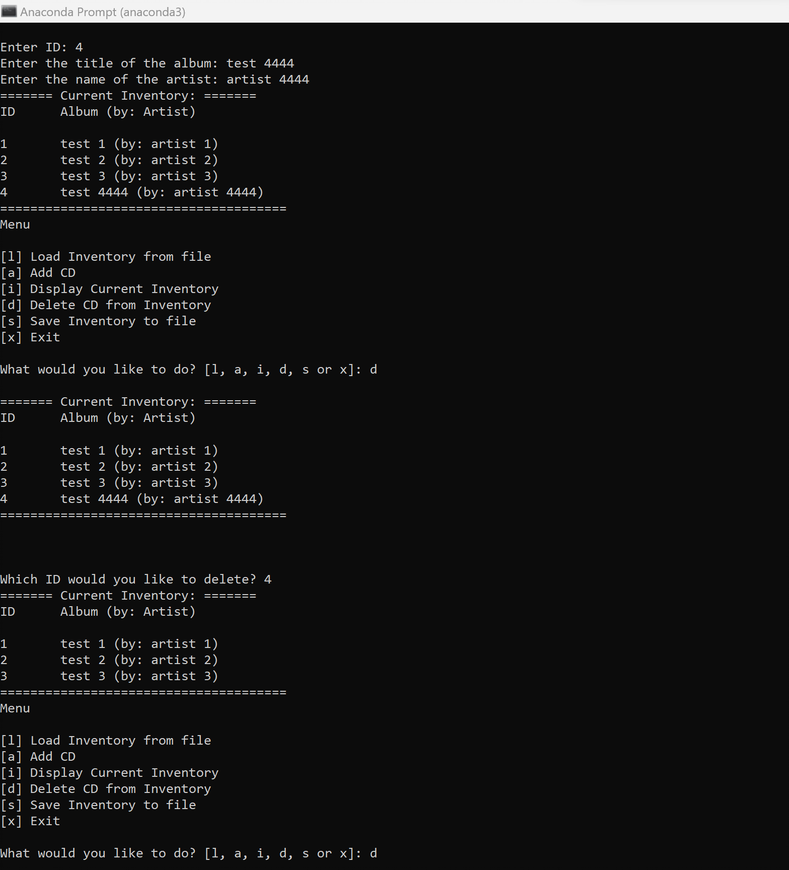
*Whoops, wrong list...delete ID 4 from the list, then save the file*

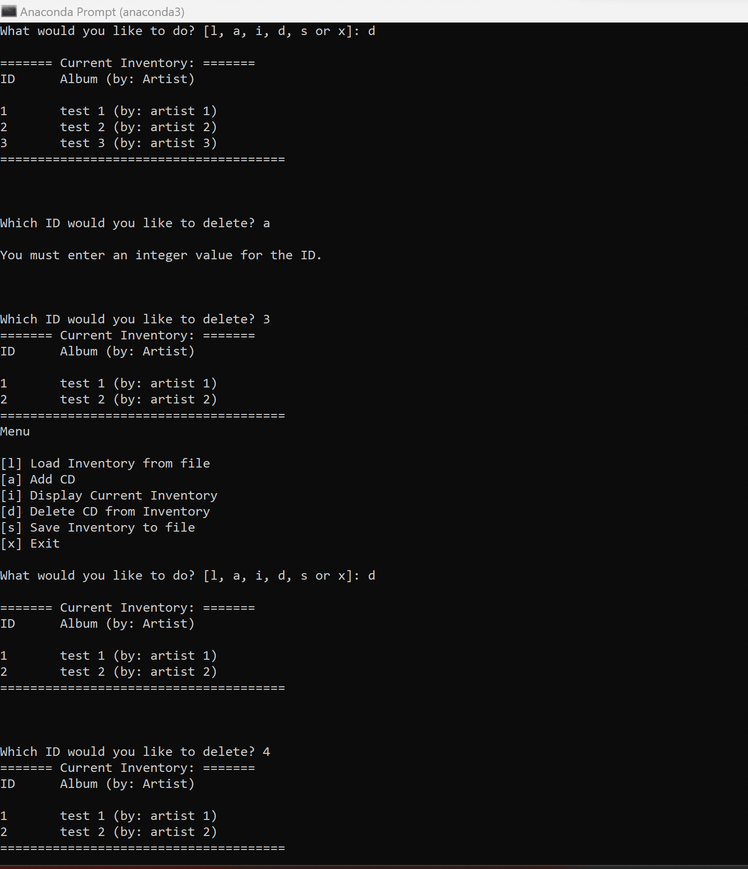


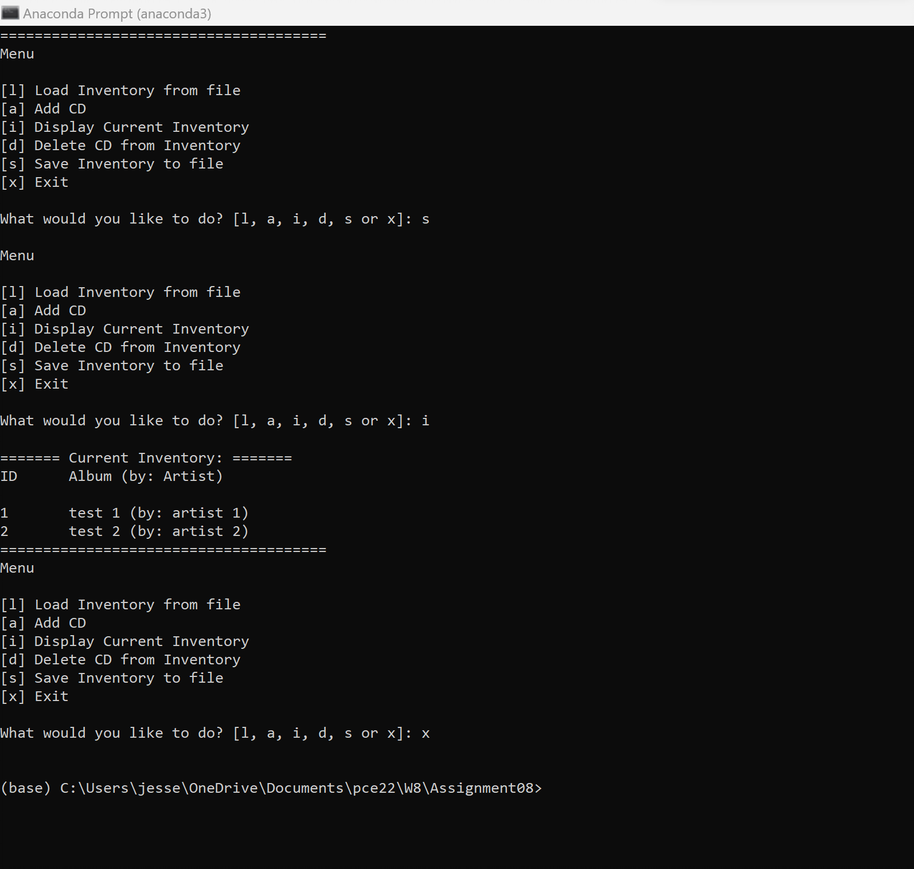
## Running in a Terminal Window











## Summary

This week’s exercise seemed more like a code review than actual coding. The one area where I stumbled the most through this exercise was ensuring each call to a class and method was passing the correct arguments as well as the correct number of arguments. Removing some of the pseudocode from previous week’s also helped make the code a little cleaner and more readable. While I still had a few issues around loading from a saved file, I had significantly less errors and issues this week.