Assignment 08

# Introduction

The goal of this assignment was to apply the lessons from Module 8 to create a functional CD Inventory program using object-oriented programming.

This document will describe the steps taken to create a functioning set of code from the pseudocode Assignment08\_starter.py file. This document will be organized into sections based on the individual TODO tags:

# TODO Add Code to the CD class

# TODO Add code to process data from a file

# TODO Add code to process data to a file

# TODO add docstring

# TODO add code to show menu to user

# TODO add code to captures user's choice

# TODO add code to display the current data on screen

# TODO add code to get CD data from user

# TODO Add Code to the main body

# Create CD Class

The CD class was built with Fields, a Constructor function, “getter” and “setter” properties, a static method to count the number of CDs, methods to format the object into a dictionary or list, and a dunder function to format the CD Info as a list of strings for easier processing and troubleshooting.

The counting CDs method and the method to populate the object into a list did not end up being used in this CD Inventory program.

# Add File Processing Code

The file processing class was built as three static methods: load, save, and export.

The load function used the pickle module to directly read the list of objects from a binary file and assign them to a list variable.. Error handling was implemented to deal with a missing file and general exceptions. In the ‘except’ cases, the list variable was initialized to an empty list to avoid returning a None list.

The save function used the pickle module to directly write the existing list of objects to a binary file. Error handling was implemented to report general errors without crashing the program (file access being the most likely to occur).

The export function converts the CD attributes to a formatted string and then writes them to a text file. This allows the user to have a human-readable inventory list they can save on their computer and import into other programs or view outside of the CD\_Inventory script. Error handling was implemented to report general errors without crashing the program (file access being the most likely to occur).

# Add IO Code

The IO code was mostly repurposed from the previous assignment.

The display inventory function was different in that it directly referenced the attributes of the object in each row, which makes the code more readable. Also, if the order of ID, artist, and title had been mixed up in a different function, it would have remained formatted correctly in this function.

The collect data function was different in that instead of creating a dictionary, it created a list of strings to be later passed into a function that creates the CD object. Creating a dictionary would be unnecessary as having the attributes of the object labeled correctly eliminates the need for keys.

# Add Code to the Main Body

# Conclusion

Opportunities for improvement: automatic IDs, adding metadata and formatting to export function,

# Appendix

## Source Code

## Code Screenshots (Spyder)

## Code Screenshots (Terminal)

## GitHub Repository: