Aime Ayers
November 16, 2022
IT FDN 110 B
Assignment 06

# Modifying the CD Inventory Program

## Introduction

In this document I will describe the concepts used to modify and test the CDInventory.py script as required for assignment 06.

## Approach

I took a different approach in this assignment due to the requirements of moving existing code into functions. Since we were primarily moving existing blocks of code I needed to focus on a few things. The first, was to make sure I was not moving too much or too little of the existing code into the functions. The second was to make sure I understood what the code was doing in order to write helpful doc strings for each new function I created.

## Classes

While we did not define any classes ourselves, we needed to modify existing classes that were already created in the script. Through my self-study and what we've learned in class, I understand that classes are a way that you can group objects. More importantly, we group the objects based on their basic behaviors. In this assignment, there were three pre-defined classes: File processing functionality, Data Processing functionality, and Input and Output functionality.

## Functions

The key to this assignment was making sure that we converted code into functions. This was a great addition to this program due to the semi-repetitive tasks that we needed to write in the program. By converting these to functions, we are able to write the code once and reference it throughout the program.

Swapping out more code to functions was beneficial in this assignment for a number of reasons. This change allows for the primary program to be more concise and readable. It has allowed us to remove repetitive code and only have it in one place, which also helps when modifying and testing.

Since we were modifying existing code, I needed to make sure I was not moving too much into the function. One example of this was in section 3.6. My initial inclination was to move the entire if/else block into the function, but after additional consideration, I realized the if/else code did not cleanly fit the definition of file processing, but rather I/O. For that reason, I decided to leave the if/else in the main program, and only move the write file code into the function.

## Doc Strings

After moving the existing functionality into a function we needed to write doc strings for each. These are meant to define the functionality of the function, as well as any arguments or returned items. There are helpful for future developers, as well as documentation.

## Testing and Validation

To test my program modifications, I ran the program after each block of code was moved into a function before moving on to the second modification. To simplify my troubleshooting process, I wanted to make sure I did not alter too many things at once before validating it works. After completing all of the modifications, I executed the program in both spyder and the terminal to validate everything was running as expected, as well as displaying as I thought. Additionally, I validated the data saved to the file as expected.

Testing evidence is included in the screenshots below.

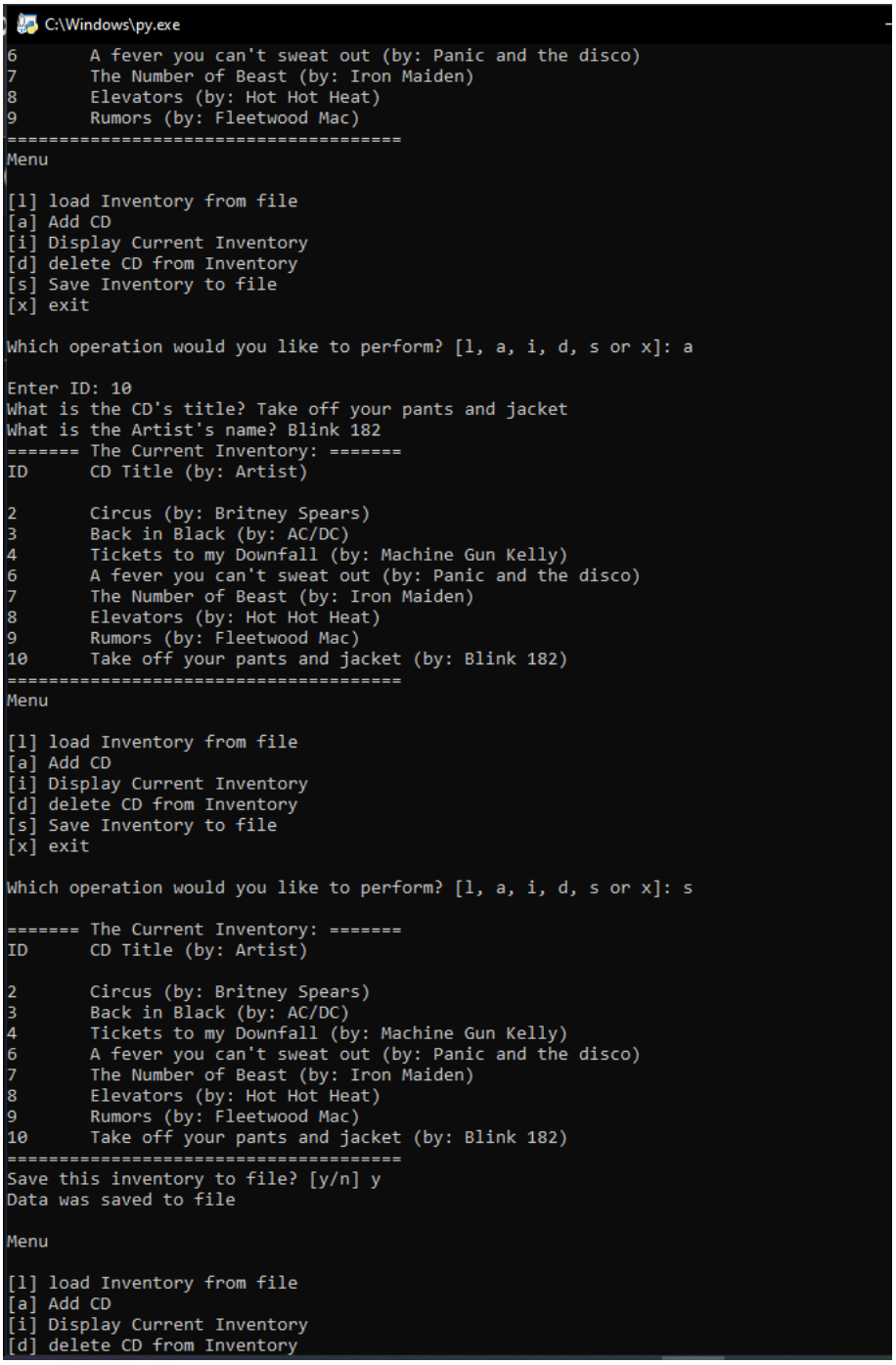


*Figure 1 Terminal Execution of Add and Save functionality*

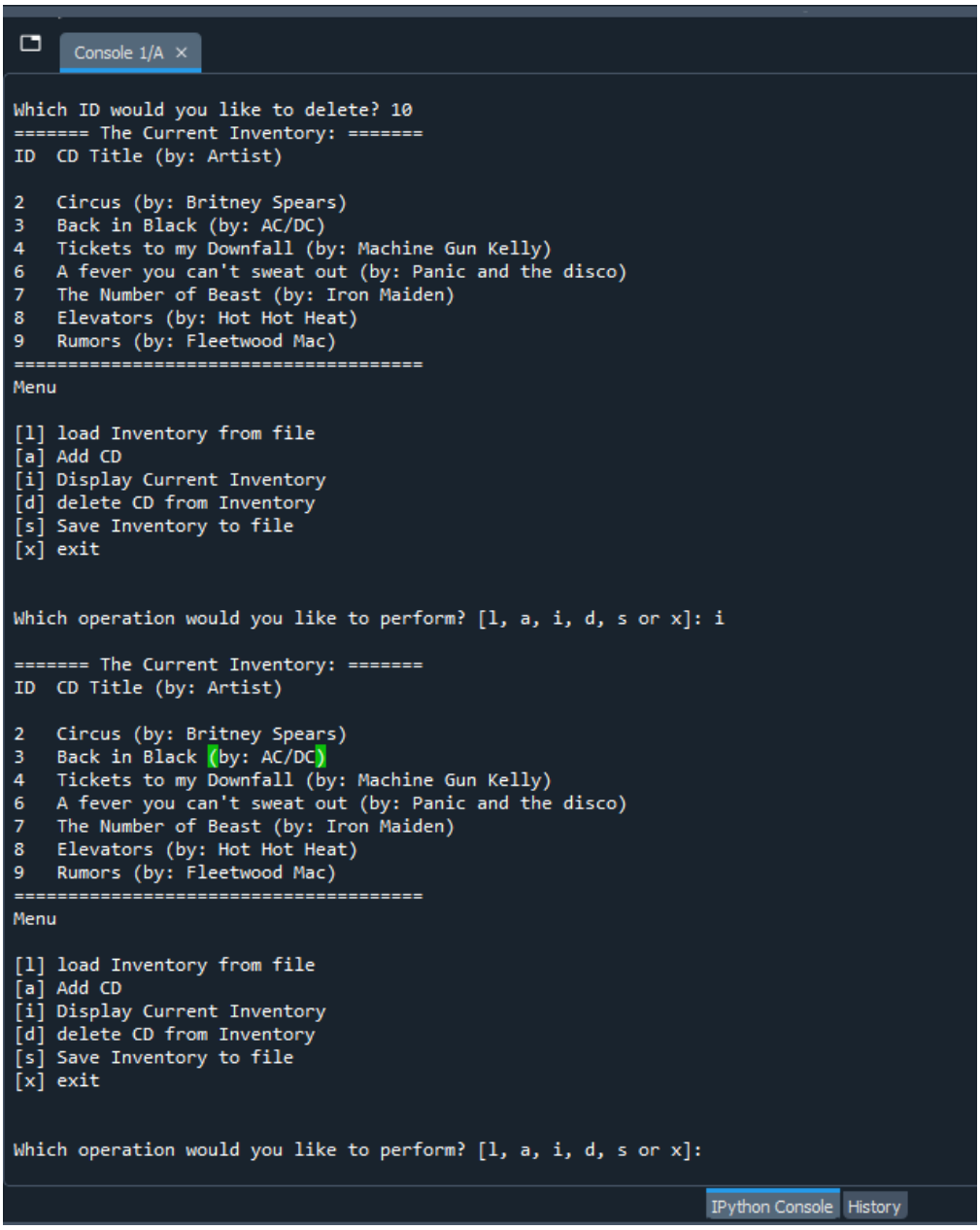


*Figure 2 Spyder execution of delete and display functionality.*

## Publishing

Code was published to the following git repository:  https://github.com/a-ayers/Assignment_06