Jeffrey Hsu

2020-Feb-27

IT FDN 100

Assignment06

Modifying CD Inventory to use classes and functions

Introduction

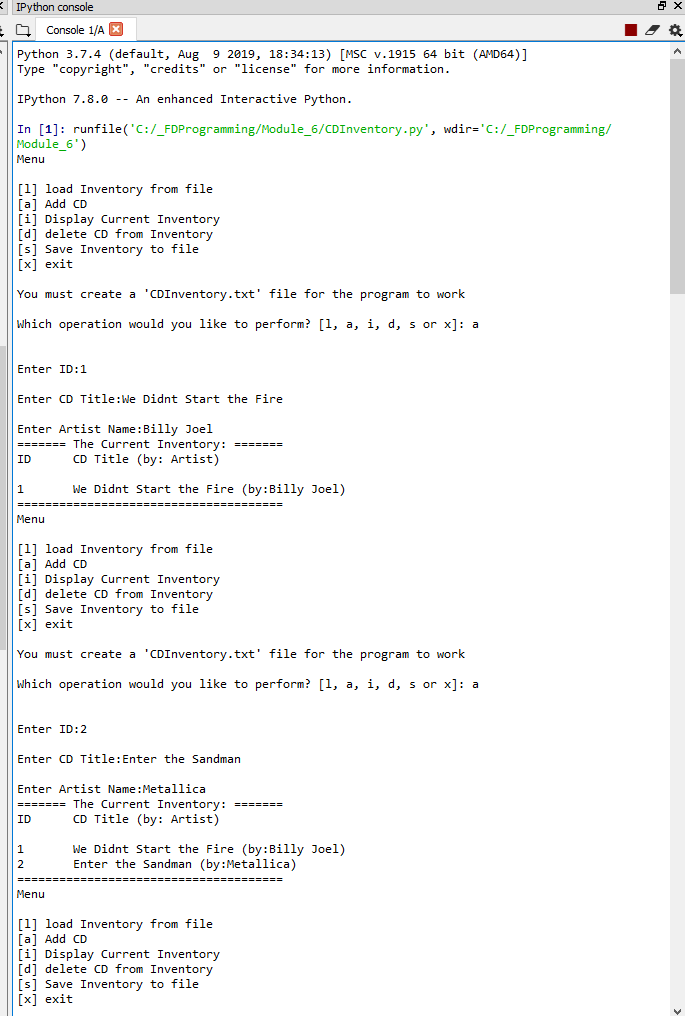
This assignment is an introduction to classes and functions and how to use them within a script. We also learn about docstrings which explains each line of code in depth and help others understand what you’re writing.

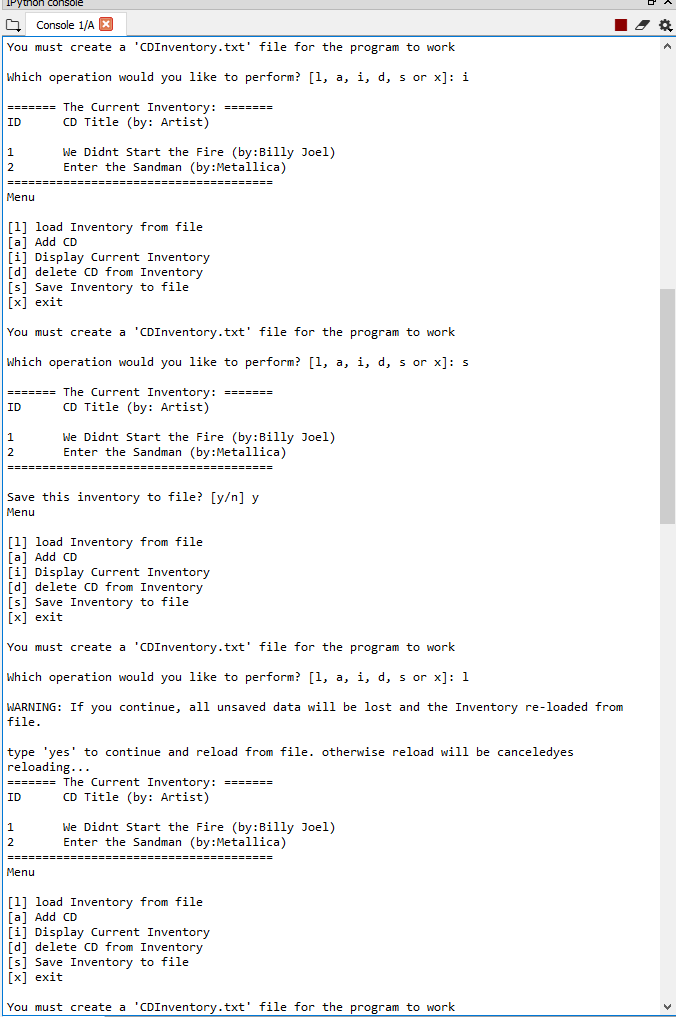
Getting Started

I first review the document provided, Foundations of Python Module 6[[1]](#footnote-1). I looked through the documents to see how to use functions and what their purpose was. Then I looked over docstrings and how classes fit with functions. After looking over the examples and how it works for functions and classes, I opened the CDInventory starter.py file and started to look over the task needed to complete the assignment.

Working on the CD Inventory Script

Like last assignment, we were given a base code and a todo list. This time however, we have a sample solution from last weeks assignment to work with. The main objective of this assignment is to group up the functions of the script like adding, deleting and saving into classes and then functions to help organize the script. This works by essentially moving the syntaxes into a class and function and then implementing the appropriate line of script within the main body. One advantage to this is it makes the script cleaner and more organized, sometimes making it easier to understand what its trying to do. At first glance, I was a bit intimated by the task at hand but after reading through the script several times along with the main goals, I started to get the jest of what Im suppose to do. I played around with script and referred to the document in order to group all the functions of the script into actual classes and functions. The DataProcessor and FileProcessor classes were relatively straightforward to implement in the script however the biggest issue I ran into was the I/O function. This was a bit tricky due to the input of 3 separate strings or arguments but after playing around with the coding and experimenting it multiple times, I was able to figure it out. Additionally, adding the code that uses the defined functions was also giving me some trouble. Some didn’t require arguments while others like the save function did. Another change I opted for was warning the user that they must create a txt file for this script to work. I decided to go this route just so I wouldn’t mess up the save function. As you can see in *Figure 1: Script running in Spyder*, the program works fine in Spyder. It also works fine in Anaconda as shown in *Figure 2: Script working in Anaconda.*





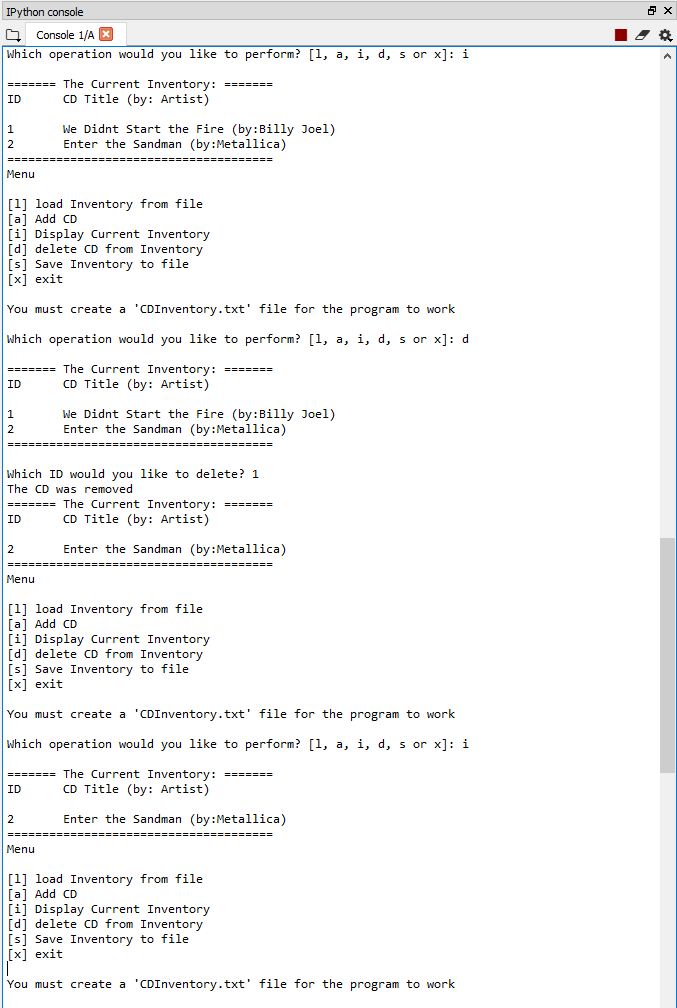
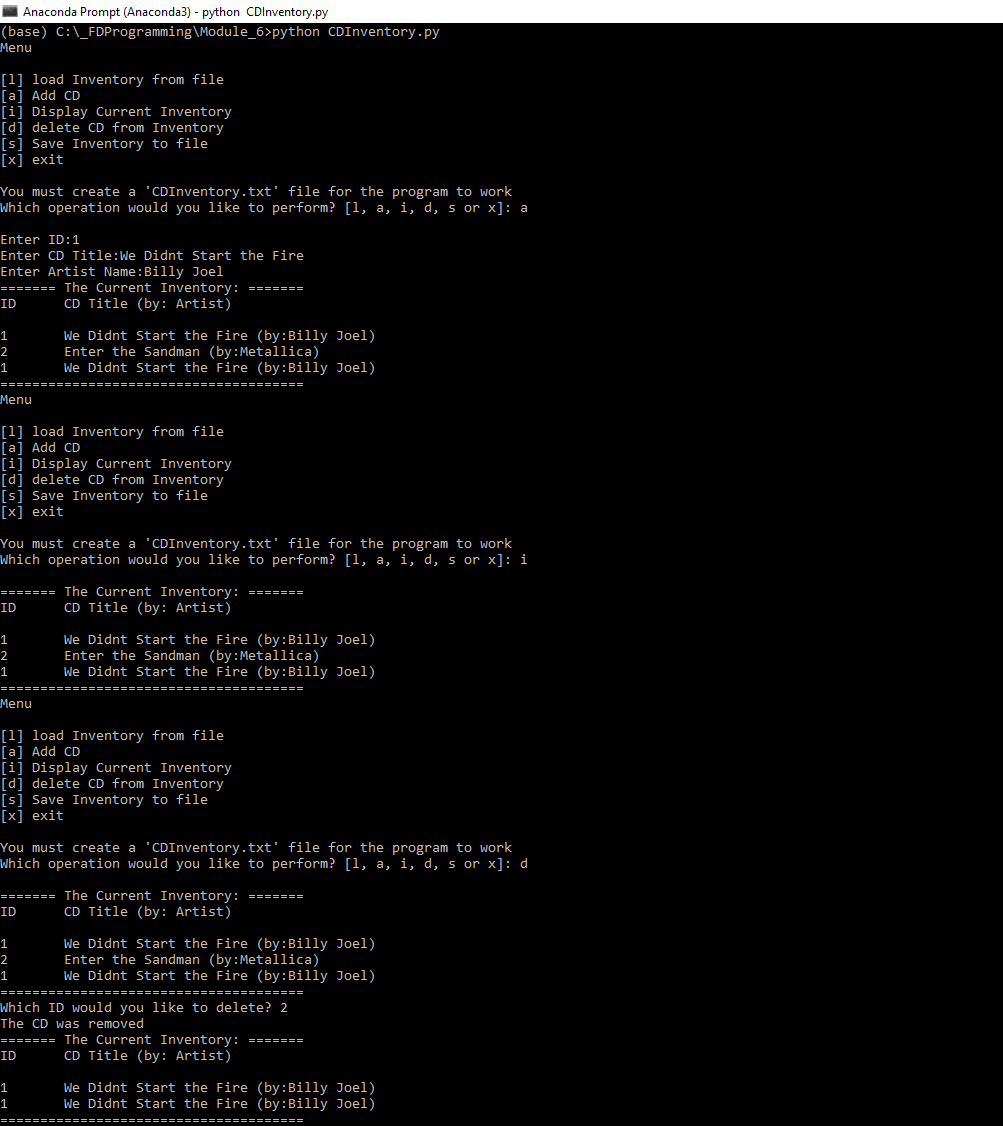


Figure 1: Script running in Spyder



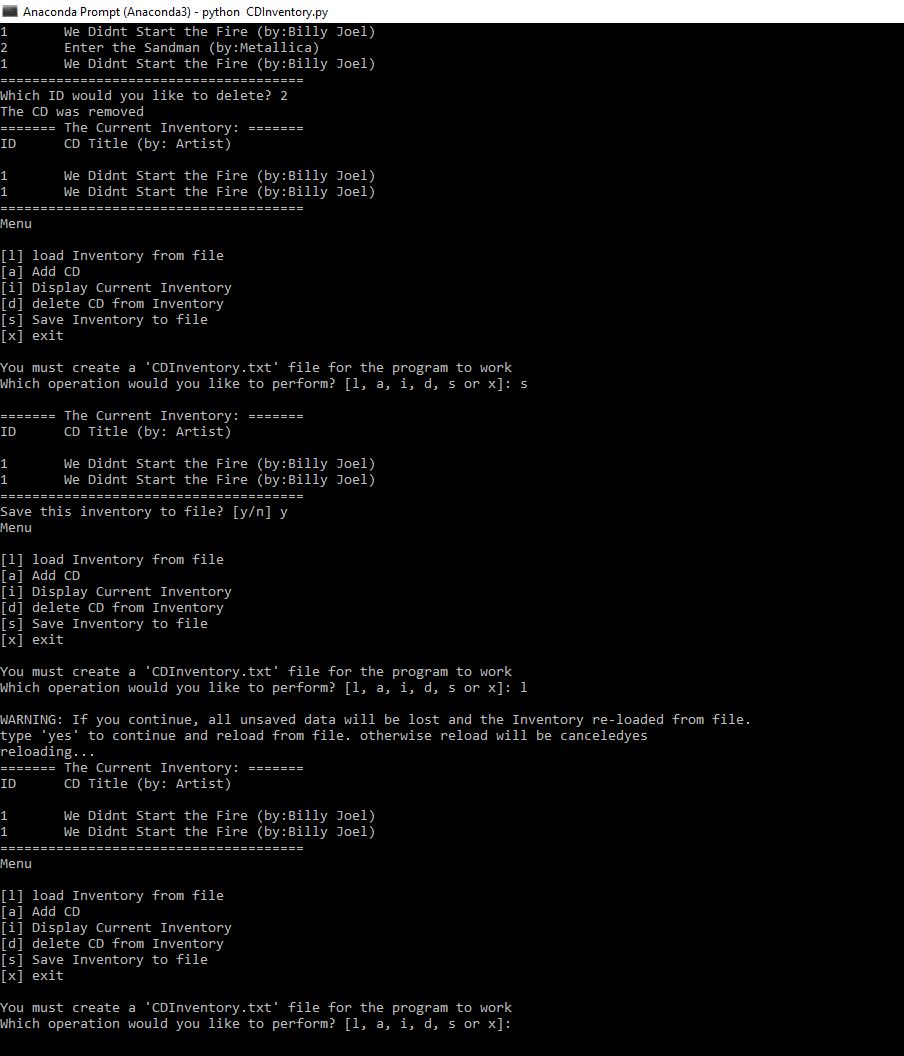


Figure 2: Script working in Anaconda

Summary

Overall this script seemed very intimating at first due to how long it was however it requires the user to carefully look over the script along with the tasks in order to understand it. By reviewing through the document and script, I was able to slowly implement functions and classes by moving all the syntax commands and entering the proper code to make them work. One of the more difficult parts was figuring out how to use classes and functions in the actual script and make them work properly.

1. Foundation of Python Module 6 <file:///C:/_FDProgramming/Module_6/FDN_Py_Module_06.pdf> [↑](#footnote-ref-1)