Nitasha Woodward

December 11, 2021

IT FDN 110 B Au 21: Foundations of Programming: Python

Assignment 09

Working with Modules

# Introduction

In this assignment I will explain the steps I used to update the Lab09\_B starter code provided for creating a CD Inventory program using CD and Track objects and various modules.

# Creating the Code

### Data Classes

I started here by defining the \_\_init\_\_ constructor and track properties outlined in the *class Track* docstring: position, title, length. I started with this module because no other modules were being imported into this code; as such I figured it would be easier to complete and test this module because it was not relying on code from other modules.

I added a *setter property* to the cd\_tracks property [not sure if this was excluded intentionally? i.e. should we not set *setter properties* for lists?]

I realized while testing that the attribute name that I created for cd\_t*racks* was different than the one mentioned elsewhere in the starter code. The starter code used *\_\_tracks*, I used *\_\_cd\_tracks* as noted in the *class CD* docstring. I had to update all of the places in the started code where \_\_*tracks* was referenced.

### Processing Classes

I updated this module next, since it only relied on importing the DataClasses module. To select a CD, I used a for loop to loop through the list of cd objects and checked to see if any of the CD numbers matched the CD index argument. The ability to use dot notation to reference properties of another module is still a bit confusing, because it seems like I would have to have both modules open to know that I needed to select .cd\_id, but I think there’s likely an easier way to make this association/connection.

I am unsure of how, if, and when I should be using try / except versus if /else and if they should be used together? However, I did find out when I tried to use if/else in the *select\_cd* method, that I had to get rid of the *else* or it would return the Exception message if I picked a cd that was not the first CD in the list.

### IO Classes

I was initially confused how the *file.write()* code on line 45 knew to use the *.get\_record* method from the *CD* class versus the *Track* class, since each class has a *.get\_record* method and the class is not referenced on this line, but I think the correct method is being invoked because the *cd* in *lst\_Inventory* is already a CD object [not a Track object]. Therefore, there’s not need to call out that *cd = DC.CD()* in the code.

This module was by far the MOST difficult to update because of needing to understand the relationships between the DC, PC and this module. I added inline comments to help me understand what each line of code was doing as I wrote and rewrote the steps that needed to occur.

I also added some error handling around the ID input entry, so that users cannot get past this input section if they do not enter an integer [keeping the error handling closest to the input]. It took me a long time to figure out the right structure to put in place to repeatedly ask the user for an integer input, instead of just ending the program.

### CD Inventory

This module was the easiest to update, since all that needed to be added was a sub menu structure, to handle Track information, that was similar to the CD menu structure.

# Saving the Script

After writing and testing my code in Spyder using the starter code, I navigated to the folder in my Home Folder called ‘Python’, in which I created a folder for ‘Assignment09’ and saved my 4 updated files.

# Running the Script

Once I saved my final script, I ran the program successful in Spyder as well as my Terminal. *No figures included for this optional Knowledge document.*

# Summary

After completing the assigned reading, labs, and videos in Module 09 I was able to modify the Lab09\_B starter files to create a working program that uses software objects for CD to create a CD Inventory program, as well as handle CD track information. I have posted the assignment .py file and this document to GitHub for peer review:

## 