Andrew Hernandez

February 28, 2021

Foundations of Programming (Python)

Assignment 07

Professor Dirk Biesinger

**Error Handling and Pickling**

*Keywords: Error Handling, Try-Except, Exception Class, Pickle, Custom Functions,*

*Separation of Concern, Classes, Docstrings, Dictionaries, 2-dimension lists,*

*User input, While-loop, if-elif-else statement*

**Introduction**

The purpose of this assignment was to take our previous assignment and wrap all user interactions/ user inputs in try-except blocks. Furthermore, we were asked to change the method of how we save the file. Previously, we saved the file as a .txt; however, for this assignment we needed to save the file in binary format or a .dat file. We also needed to change the function to load the data from a .dat file. Whilst doing this, we were to refactor our existing code taking in consideration the comments that were given. For this assignment, I think that I did things a little backwards because I was not sure where to begin. I first started with wrapping the user inputs in Try-Except blocks, but I soon realized that I would need to rethink this once I started working on pickling the results to a .dat file.

**Pickling**

Pickling is “a module that implements binary protocols for serializing and deserializing python objects”[[1]](#footnote-1). What this means is that the objects are converted into a byte stream and saved in a .dat file. To use pickling, we had to import the pickle module. After importing the pickle module, I began by setting the objFile with the correct extension which is the .dat extension. Then I changed the open method variable to a pickling ‘rb’. I chose ‘rb’ because in the original assignment we used the ‘read’ variable which overrides all existing data in the destination. I then used the pickle.load method to load the data from the .dat file. Then I used the same logic that was done previously to assign the list the appropriate labels. For the writing data to a .dat file, I changed the logic a little to use the with open(file\_name, ‘wb’). Then, after assigning the data to the variable, I used the pickle.dump function to dump the data.

Figure :Using Pick and Unpickling

**Try-Except Error Handling**

The most challenging part of this assignment was the error handling for all user inputs. Completing this task was very time consuming because I was having issues grabbing the actual error that was being thrown. Furthermore, I had a little difficulty deciding which parts other than the obvious one needed to be handled. Also, I had to change some of the code so that the error could be thrown since previously I used if-elif clauses to handle user input. I spent most of my time trying to throw errors from the initial user selection. Unfortunately, I was not able to accomplish this in a try-except, so I left the existing code with the while loop.

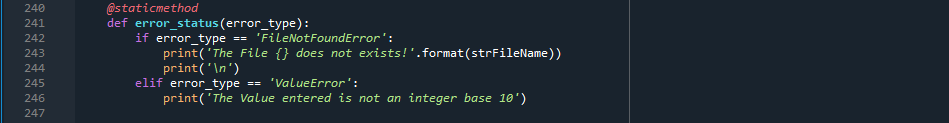
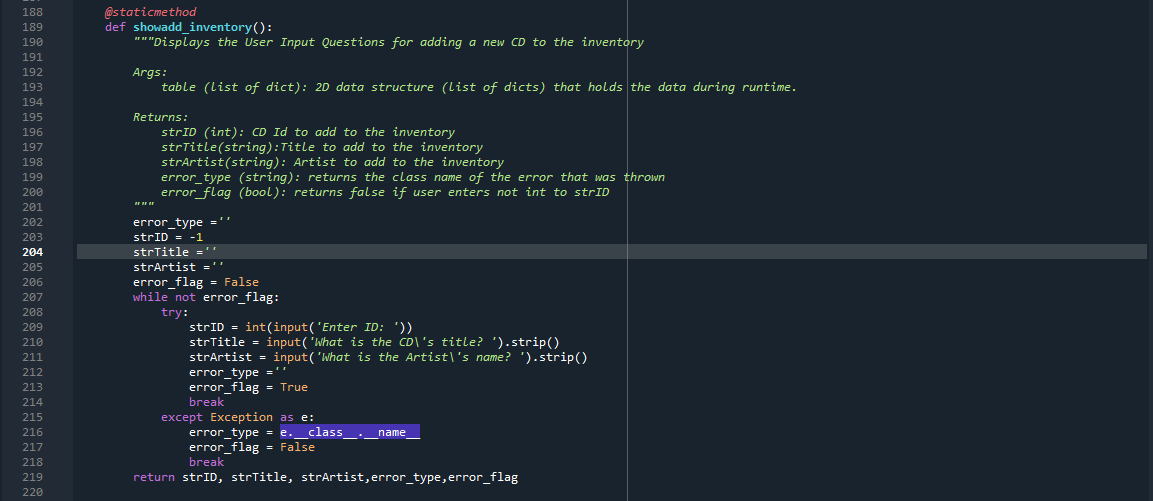
In order to properly get the error, I used the except Exceptions as e. Then I would print(type(e)) to get the class of the error. Once I got the class I would change this code to use the e.\_\_class\_\_.\_\_name\_\_[[2]](#footnote-2) to extract the class as a string. I then returned the error\_type to the main method and called my newly created error\_status function which contained the error list. 

Figure : Custom Error Handling Function

 When adding to the inventory, I had to get a little creative because I wasn’t sure how to throw the error back to the main without sending all of the return variables back; however, since the program errored out and no other variables were assigned after the Id it got a little tricky. I was able to accomplish this by creating an error\_flag and setting the flag within he try-except block. Doing this allowed me to capture the error.

One area that I was a little confused and decided not to tackle since I was not sure if it was the appropriate way to complete the task was calling another IO function within the IO class. To me, this would seem OK since they are in the same class and are both part of the presentation Separation of Concern and it seemed a little awkward to send the error\_type back to the main then call the error\_status function. Instead of doing this I created the code above that sends all the data back then relies on the strId to decide if the error function needs to be called.

**Summary**

Overall, I feel much more comfortable working with basic classes and functions and passing variables to functions. After reviewing last week’s assignment, I realized the issues that I was making and I was able to fix those in this assignment. I feel like I have an OK understanding of error handling and using the try-except blocks. However, one thing that I’m still confused about is why we would use Pickling since according to the python manual it is not a secure way of storing data[[3]](#footnote-3). I do think it is an interesting concept, I just don’t know when I would be using such a function in programming.

To wrap up this assignment, I created a repository in GitHub for my source code and this document. The link to my repo is as follows: <https://docs.python.org/3/library/pickle.html>

1. <https://docs.python.org/3/library/pickle.html>, accessed 28-Feb-2021 [↑](#footnote-ref-1)
2. <https://teamtreehouse.com/community/classname-2>, accessed 28-Feb-2021 [↑](#footnote-ref-2)
3. <https://docs.python.org/3/library/pickle.html> accessed 28-Feb-2021 [↑](#footnote-ref-3)