**CD\_Inventory**

**(Error Handling)**

**(Pickle Module)**

*Introduction:*

In current Assignment we were introduced to 2 concepts: Error handling and serializing and deserializing data using “Pickle” module. Error handling and raising exception tool is powerful method to sharpen your program runtime we can not only use different Exception classes introduced in python but also create our own custom ones

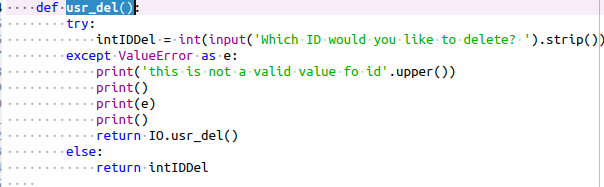
“Pickle” module is used to save different types of data structures starting from simple ending with complex ones (list of directories, tuple of tuples) in binary mode. It operates using several instruments like dump and load, It is mainly intended to use within python itself.

*When you learn from your ‘Errors’:)*

In this part I tried to represent my understanding of error handling concept and use it to improve my program.

I started working with IO class and my first function on the way was “*usr\_del()”* function. It uses takes user input string converts it to integer using int() function.

Possible error for this function is wrong input value from user, in such case to prevent from crushing program appropriate exception is raised (*figure01)* “Value Error”

In “Try” block, the user input is checked for mentioned error and in a positive case the execution is passed to “Except” block where it first prints custom error message and than the standard one from variable “e” previously saved by block. (*ref. FDN.mod07.18) figure01*

at the end it returns itself back to the caller, to initiate a loop. (This concept I used all through this program really not sure if I made any mistake by doing so, otherwise the return value from except block was issuing additional errors)

The next step in the same class was “usr\_input()” function is basically same error but in this case possible initiation would be the user input in “strID”variable, as in case above except block prints 2 messages: custom and standard one. (*figure02)*

(ref. FDN.mod07.18)

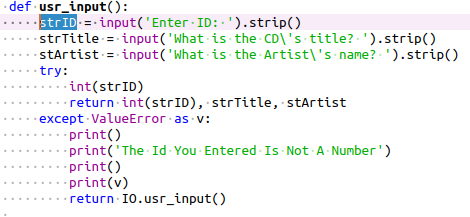
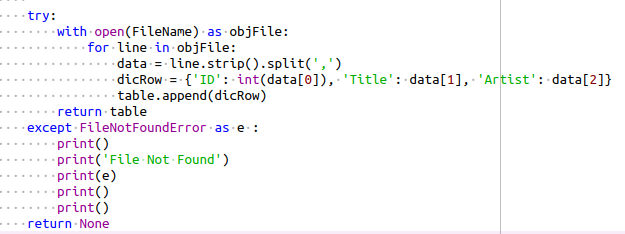
**

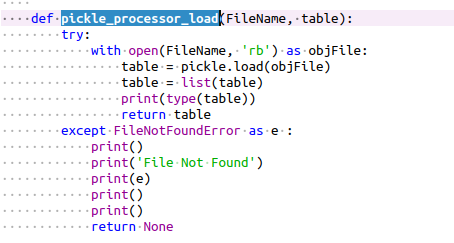
figure 02

On the next step we move to “File\_Processor Class” and with the function “read\_file” we get introduced to the new error class “FileNotFoundError”.

There is an attempt to open a file “CDInventory.txt” from the current directory by the “Try” block and in case of success the script executes as intended. Otherwise if the file was not found it is passed to “except block for predefined error messages and return value in this case is none.

(*figure03**)*

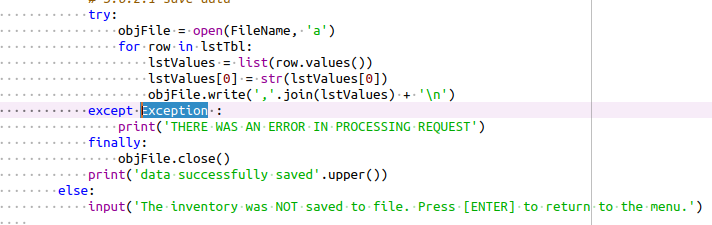
(Same structure of error handling you can find in “pickle\_processor\_load” but this block is still under construction :)



*Figure 04*

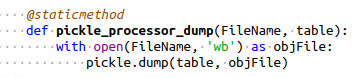
The final step in this assignment would be the last possible error in same “File\_Processor Class” in

“write\_file”(). Here I am catching possible permission error but I choose general class “Exception”. The main difference here is that together with try/except blocks I used “finally” block to make sure the file is closing. (ref.docs.python.org/3/tutorial/errors.html)



*Figure 05*

The last part of our assignment was writing data in binary files particularly using the pickle module. Here I used dump function to save the data to 'CDInventory.dat' file. *(ref. FDN 07 14)*



Conclusion

In this assignment I used materials provided from module as well as tried to make research on my own. I still working on appropriate and correct format for my scripts. I did my best effort take into consideration remarks from the previous assignment.

