Assignment 07

# Introduction

The goal of this assignment was to research Exception Handling and Pickling and then to apply these concepts to the existing code from Assignment 06.

This document will describe the research sources used and how the CD Inventory program was updated to included error handling and binary data.

# Exception Handling Research Sources

<https://www.programiz.com/python-programming/exception-handling> – Describes the concept of Exceptions in Python. Gives multiple examples of how to implement exception handling in a program. Concludes with a description of the finally function.

<https://www.geeksforgeeks.org/python-exception-handling/> - Similar content to the previous source. Includes example scripts to become familiar with application. Introduces the concept of raising exceptions.

<https://www.w3schools.com/python/python_try_except.asp> – Nicely formatted source that steps through exception concepts. Introduces the concept, describes how to handle multiple exceptions, describes implementation with an else clause, and introduces the finally and raise concepts.

# Pickling Research Sources

<https://www.geeksforgeeks.org/understanding-python-pickling-example/> - Short description of using pickle to serialize data structures. Goes into some advantages of using pickle, especially compared to marshal.

<https://www.synopsys.com/blogs/software-security/python-pickling/> - Article introducing the concept of pickling and detailing the security concerns with its use. Main takeaway: don’t pickle things willy-nilly if you don’t know where they came from.

<https://realpython.com/python-pickle-module/> - Nicely formatted, in-depth description of serialization using pickle. In addition to describing what pickle is, the article also tells you alternative methods (marshal and JSON) and their proper application. The article proceeds to give application examples, describe what things cannot be pickled, introduces dill as a way to serialize more data types, and describes the security concerns that come with pickling.

# Delete CD ValueError Handling

The existing CD Inventory program from Assignment 06 was altered to add handling for the ValueError exception in the del\_cd function. The intent was to allow the code to keep running if the user input for ID

was not an integer. The ‘except’ statement was placed in a manner to put the program back into the main loop if the error occurred.

# Read File FileNoteFoundError Handling

Error handling was added to the read\_file function to handle a FileNoteFoundError. The intent was to allow the program to keep running if the binary .dat file had not yet been created (which would be the case upon the first run of the program). The ‘except’ statement was placed in a manner that the program would go back to the main loop if the error occurred and the ‘table’ argument is re-initialized to an empty list in order to prevent a None type from being created.

# Collect Data ValueError Handling

The existing CD Inventory program from Assignment 06 was altered to add handling for the ValueError exception in the collect\_data function. The intent was to allow the code to keep running if the user input for ID was not an integer. The ‘except’ statement was placed in a manner to put the program back into the main loop if the error occurred.

# Pickling

The read\_file and write\_file functions were updated to use pickle.load and pickle.dump (respectively) to handle data in and out of the program. This turned out to be much more straightforward that using the read/write file functions; instead of converting lists to formatted strings, the lists were just pickled. Pickling accepts the list data type while reading/writing to text files only accepts string data.

# Conclusion

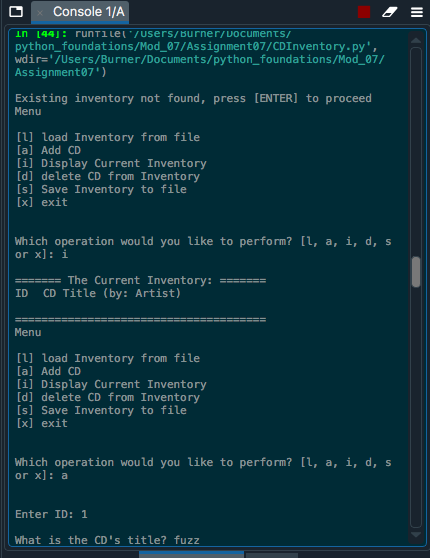
After researching the subjects, data serialization (‘pickling’) and error handling were added to the CD Inventory program. The resulting program ran as specified; it does not crash with incorrect user input and as long as the data has been previously saved, the program can recall the inventory even if the program was previously exited.

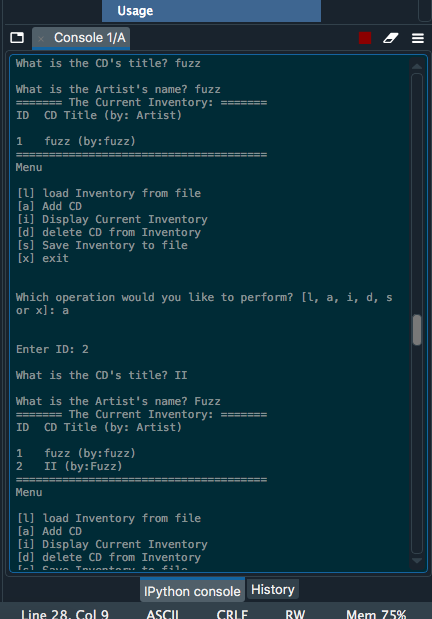
This required application of the following lessons from Module 07:

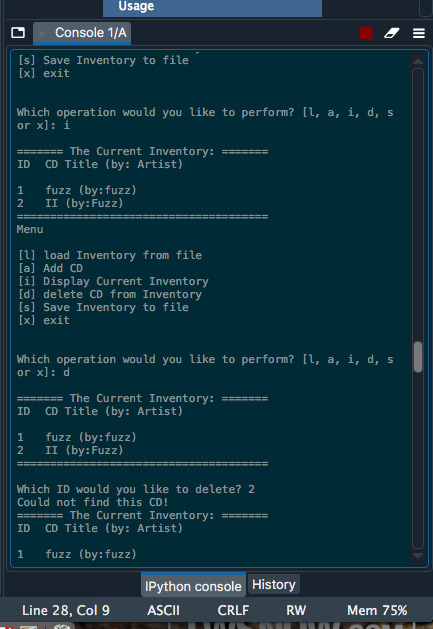
* Using the with… as option
* Working with binary files
* Structured Error Handling

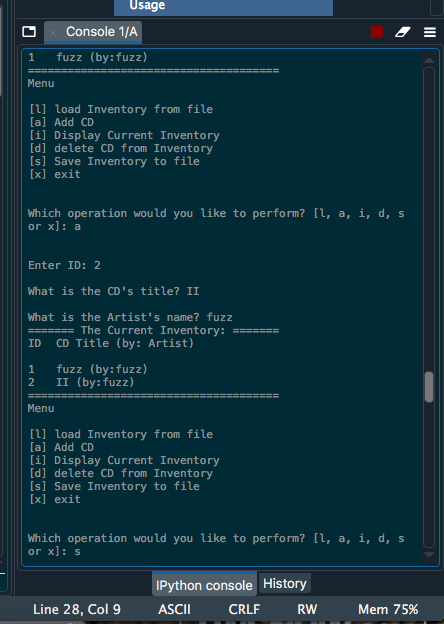
# Appendix

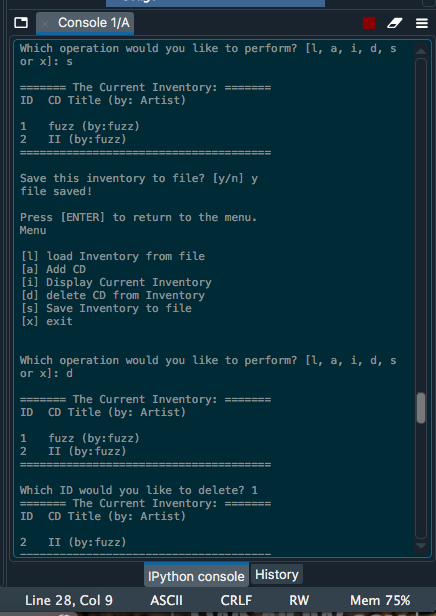
## Spyder Pictures

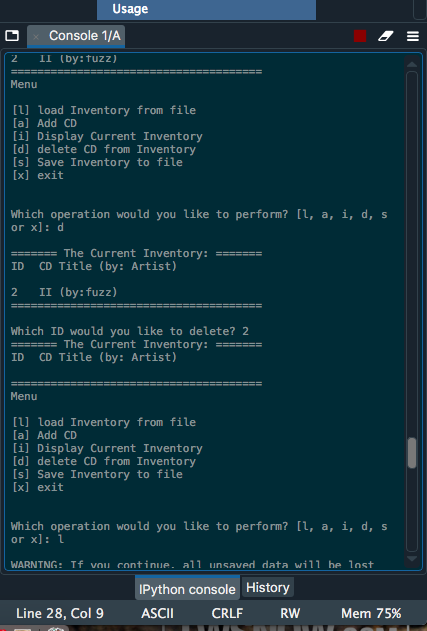


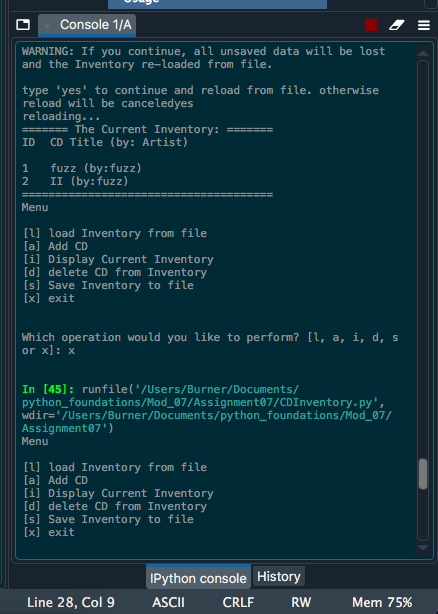


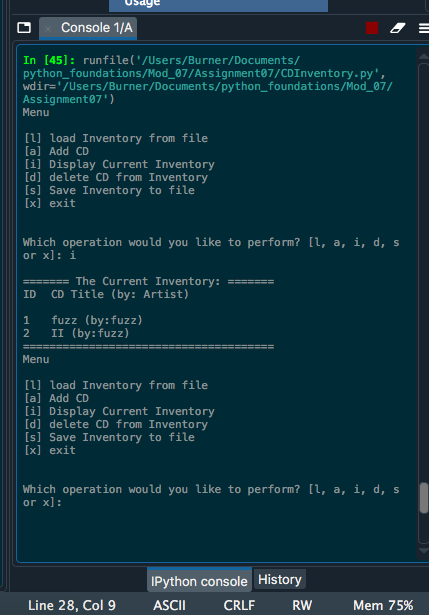






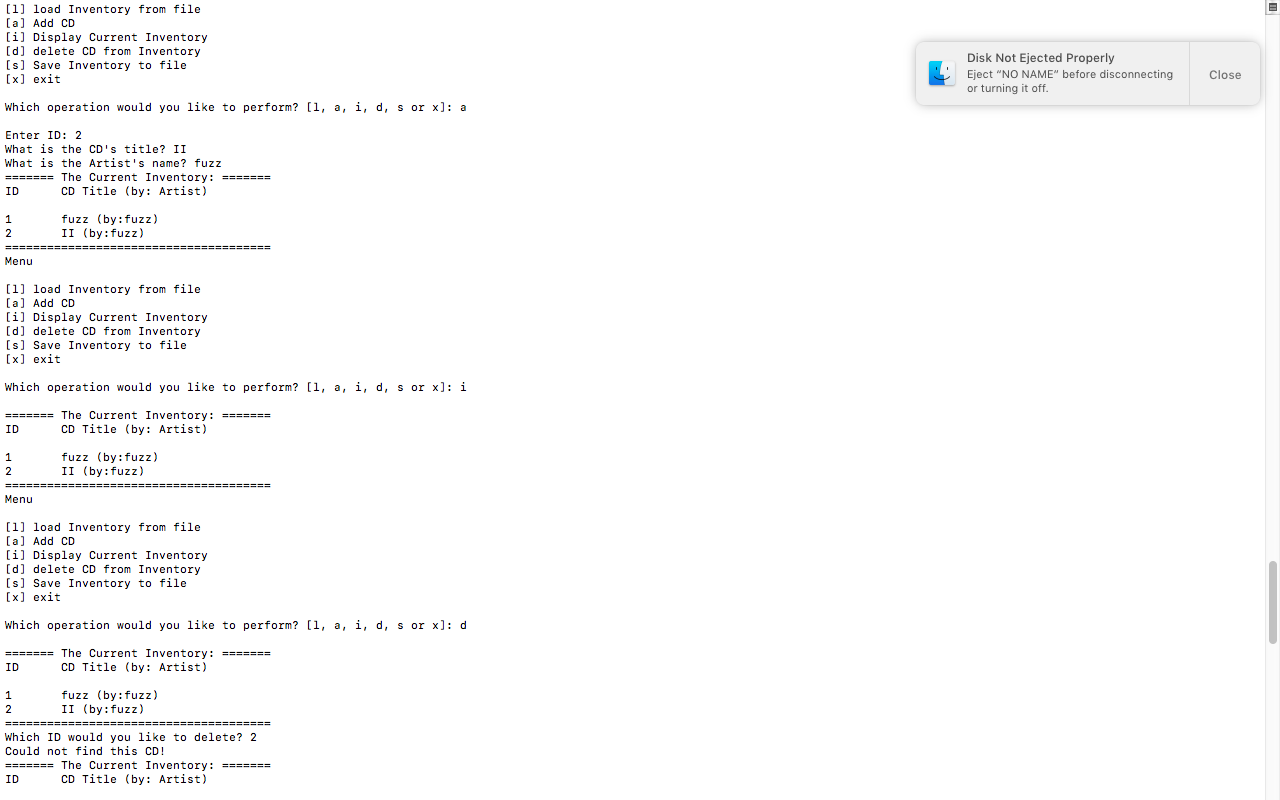


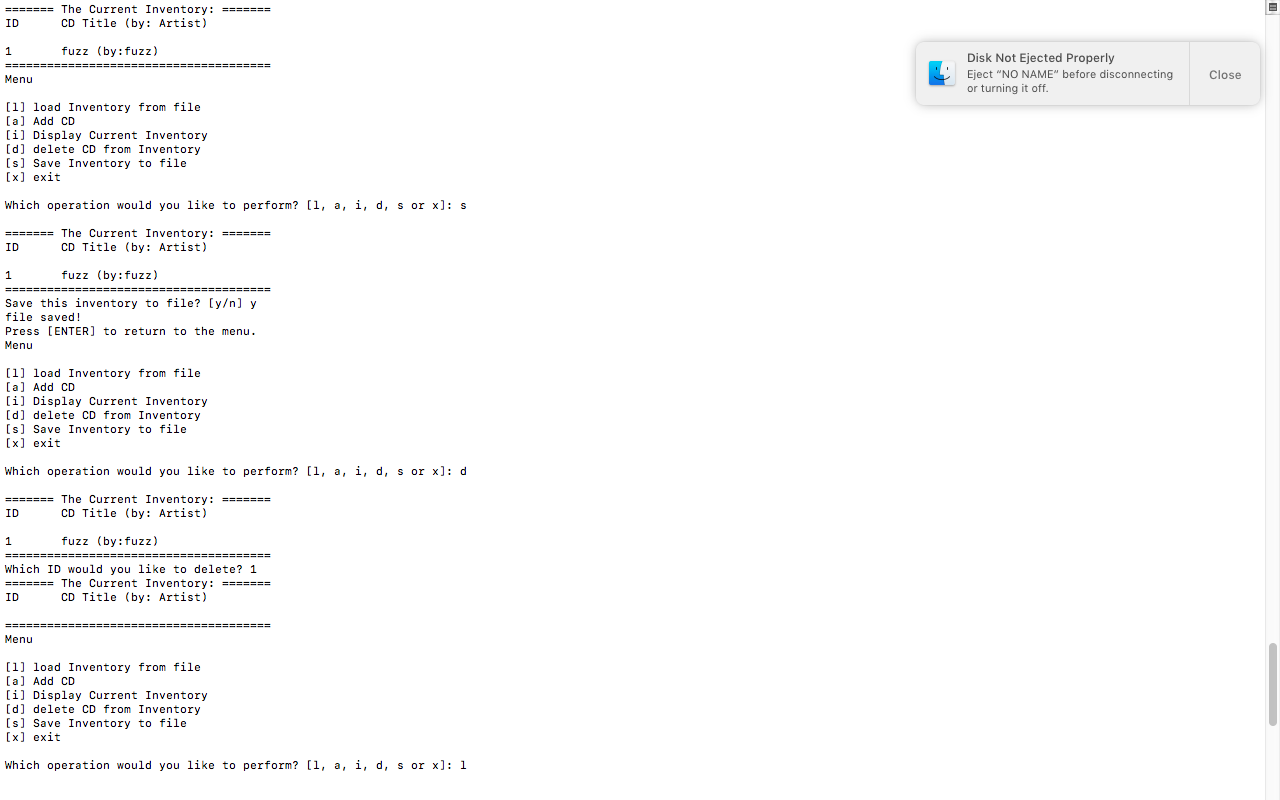




## Terminal Screenshots









## GitHub Repository

<https://github.com/canned-slammin/Assignment_07>