# Morgan Farmer

# August 24th, 2021

# Python Course

# Assignment07

<https://morganfarmer.github.io/ITFdn-110A-mod7/>

Binary Files and Pickling

Introduction

In this assignment I will be explaining how I was able to create a file, allow the user to name a file with custom errors created by me.

Step one

At the beginning of the assignment, I started out with the project header to explain what tasks were going to be completed.

# ---------------------------------------------------------------------------- #  
# Title: Assignment 07  
# Description: Working with pickling,  
# saving to structured error handling.  
# ChangeLog (Who,When,What):  
# Morgan Farmer, 8/23/21 , Pickling & structured error handling  
# <Your Name>,<Date>,Modified code to complete assignment 7  
# Dev: Morgan Farmer, 08/18/21  
# ---------------------------------------------------------------------------- #

At start of the actual code, I imported the pickle and import data that the list could be saved to, next open the file, pickle dump, then close the file. This now means that file exists to the user.

import pickle  
#import a list object  
lstData = ['aaa','bbb']  
  
objFile = open('seven','ab')  
pickle.dump(lstData,objFile)  
objFile.close()

Error handling

To do error handling you start off with the try statement. Next line, input new file name statement and display to the user what you want them to do. In this case, we want them to name the file. With “if” and “except” statements we create a custom error. To create a custom error, we raise exception that they cannot use numbers. If they use numbers the custom error that was created by you, will print to the user. The code should look like this.

try:  
 new\_file\_name = input('What do you want to name the file:')  
 if new\_file\_name.isnumeric():  
 raise Exception('Do not use numbers!')  
 objectFile = open(new\_file\_name,'rb')  
except Exception as e:  
 print('There was an error!')  
 print(e, e.\_\_doc\_\_, type(e), sep='\n')

Pickling

We started pickling in step one all we must do now is close it out. Start off by loading file data with object file and pickle load. Next, Close the file and create a print statement that reads the object file data. The code should look like this.

objFileData = pickle.load(objectFile)  
objectFile.close()  
print(objFileData)

Conclusion

In this assignment I learned that saving to binary files saves a lot more space in a computer than a regular text file. Data that needs to be as a binary file will not be able to be read by the user, but it will be there with ineligible text.