Brynn Lampman

11/30/2022

IT FDN 110A

Assignment 07

<https://github.com/blampmanuw/IntroToProg-Python-Mod07>

# Assignment 07 – Pickling and Exceptions Handling

# Introduction

This week’s assignment was to research and apply knowledge for pickling and exception handling.

# Pickling

In my research for pickling on the web, I particularly liked <https://docs.python.org/3/library/pickle.html>

The parts that I liked about this site was that it made comparisons to other python modules, like json and marshal. This site explains various parts of pickling, how to unpickle, as well as provides many examples of code.

In this assignment I chose to make a list [ID, Name]. (See Figure 1)

import pickle # This imports code from another code file  
  
intID = int(input("Enter ID: "))  
strName = str(input("Enter Name: "))  
lstCustomer = [intID, strName]  
print(lstCustomer)  
  
# Store the data with the pickle.dump method  
objFile = open("AppData.dat", "ab")  
pickle.dump(lstCustomer, objFile)  
objFile.close()  
  
# Read the data back with the pickle.load method  
objFile = open("AppData.dat", "rb")  
objFileData = pickle.load(objFile) #load() only loads one row of data.  
objFile.close()  
  
print(objFileData)

Figure 1

I tested it in Command Prompt and PyCharm. (See Figures 2 & 3)

Text

Description automatically generated

Figure 2 – Testing in Command Prompt

Text

Description automatically generated

Figure 3 – Testing in PyCharm

# Exception Handling

<https://docs.python.org/3/tutorial/errors.html#handling-exceptions>

I like to go to Stack Overflow to get people’s opinions on code that they like to try. The link above was embedded in the Stack Overflow thread. I liked an example in particular that threw an error if it was expecting an integer. (See Figure 4)

while True:  
 try:  
 x = int(input("Please enter a number: "))  
 break  
 except ValueError:  
 print("Oops! That was not a valid number. Try again...")

Figure 4 – Error Handling when expecting an integer

Then I tested the error handling code in Command Prompt and PyCharm (See Figures 5 & 6)

Text

Description automatically generated

Figure 5 – Testing Error Handling in Command Prompt

Text

Description automatically generated

Figure 6 – Testing Error Handling in PyCharm

# Using Pickling and Error Handling Together

I inserted the error handling code in the input area where the user is to input an ID. This way, if they don’t input an integer, it will tell the user that it was not a valid number, and try again. (See Figure 7)

import pickle # This imports code from another code file  
  
# intID = int(input("Enter ID: "))  
while True:  
 try:  
 intID = int(input("Please enter an ID: "))  
 break  
 except ValueError:  
 print("Oops! That was not a valid number. Try again...")  
strName = str(input("Enter Name: "))  
lstCustomer = [intID, strName]  
print(lstCustomer)  
  
# Store the data with the pickle.dump method  
objFile = open("AppData.dat", "ab")  
pickle.dump(lstCustomer, objFile)  
objFile.close()  
  
# Read the data back with the pickle.load method  
objFile = open("AppData.dat", "rb")  
objFileData = pickle.load(objFile) #load() only loads one row of data.  
objFile.close()  
  
print(objFileData)

Figure 7 – Pickling and Error Handling together

Here is a screenshot of the binary file (Figure 8)

Graphical user interface, text

Description automatically generated

Figure 8 – AppData.dat binary file

# Conclusion

The purpose of this assignment was to teach how to use pickling and error handling. Pickling can come in very handy because it is more obscure and is smaller in file size. Error handling will be able to tell the user or the developer where they made the error. Instead of getting a generic error message, it will be more specific of what the error was.