Daniel B

August 24, 2020

Foundations of Programming: Python

Assignment 07

# **Assignment 07**

# Introduction.

This document describes the steps taken to perform Assignment 07, which requires a program that utilizes Python’s “pickle” and custom error handling.

# Steps Taken for Assignment 07.

1. Created a new project for Assignment 07 inside of PyCharm. Downloaded Assignment 07 starter.
2. Watched the getting started video for Assignment 07.
3. Created an idea for the project – a simple log of drinks consumed.
4. Added the assignment header.
5. Started by importing pickle. See Figure 1.



***Figure 1.***

1. Added a message to the user that describes the purpose of the program. See Figure 2.

******

***Figure 2.***

1. Added inputs to collect a drink and the amount of a drink from the user. See Figure 3.

******

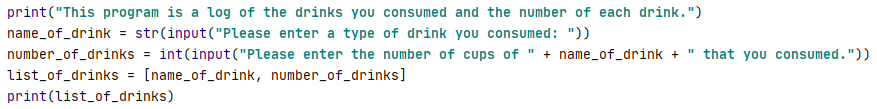
***Figure 3.***

1. Placed the two variables into a list object. See Figure 4.

******

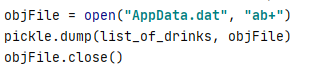
***Figure 4.***

1. Added a print statement so the user can see the full list of drinks after entering data. See Figure 5.

******

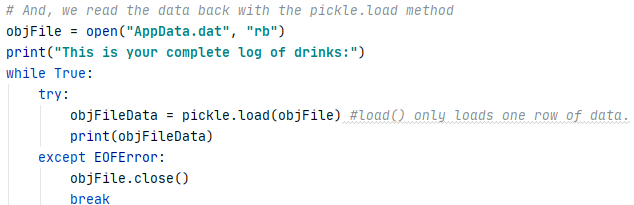
***Figure 5.***

1. Added a pickle.dump to store the data in list\_of\_drinks to a .dat file. See Figure 6.

******

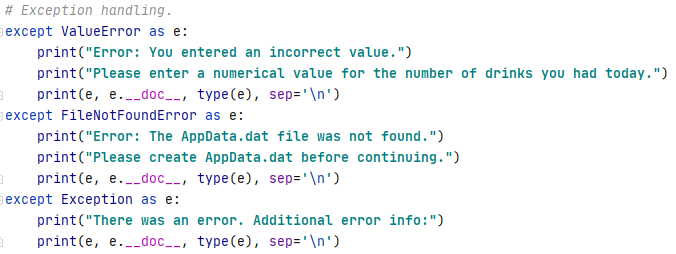
***Figure 6.***

1. Added a while loop with a pickle.load that prints each object in the AppData.dat file. I utilized an exception for when it reaches the end, which initiates a break instead of giving an error to the user. See Figure 7.



***Figure 7.***

1. Added exception handling. ValueError initiates when a user fails to enter an integer for the number of drinks consumed. I also added FileNotFoundError, even though the code is written to write a file if it does not exist. I lastly added a general exception that is a catch all. See Figure 8.



***Figure 8.***

1. I felt like I could use more error handling. One idea I came up with is to reject a drink that is named a numerical value, because that’s probably an incorrect entry. To do this, I used an .isnumeric() if statement that raises an exception with a custom message. See Figure 9.



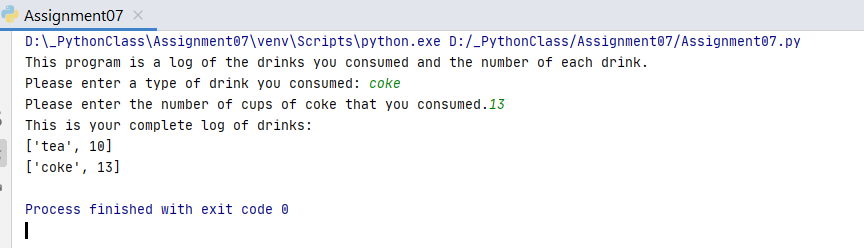
***Figure 9.***

1. I also added a custom error for when a user enters a number for drinks in excess of fifty, because it is unbelievable that anyone had more than 50 drinks. See Figure 10.

******

***Figure 10.***

1. I tested my code, and it works as expected. See Figure 11.



***Figure 11.***

1. Here is the complete code:
2. *# ------------------------------------------------- #  
   # Title: Assignment 07  
   # Description: A program that pickles data into a binary file, with error handling.  
   # ChangeLog: <2020-08-04>, added error handling  
   # <DanielB>,<2020-08-04>,Created Script  
   # ------------------------------------------------- #*import pickle *# This imports code from another code file!*try:  
    print(**"This program is a log of the drinks you consumed and the number of each drink."**)  
    name\_of\_drink = str(input(**"Please enter a type of drink you consumed: "**))  
    if name\_of\_drink.isnumeric():  
    raise Exception(**"Please do not use numbers as a drink name."**)  
    number\_of\_drinks = int(input(**"Please enter the number of cups of "** + name\_of\_drink + **" that you consumed."**))  
    if number\_of\_drinks > 50:  
    raise Exception(**"Are you made of liquid? Enter a reasonable number (under 50)."**)  
    list\_of\_drinks = [name\_of\_drink, number\_of\_drinks]  
     
    *# Now we store the data with the pickle.dump method* objFile = open(**"AppData.dat"**, **"ab+"**)  
    pickle.dump(list\_of\_drinks, objFile)  
    objFile.close()  
     
    *# And, we read the data back with the pickle.load method* objFile = open(**"AppData.dat"**, **"rb"**)  
    print(**"This is your complete log of drinks:"**)  
    while True:  
    try:  
    objFileData = pickle.load(objFile) *#load() only loads one row of data.* print(objFileData)  
    except EOFError:  
    objFile.close()  
    break  
     
   *# Exception handling.*except ValueError as e:  
    print(**"Error: You entered an incorrect value."**)  
    print(**"Please enter a numerical value for the number of drinks you had today."**)  
    print(e, e.\_\_doc\_\_, type(e), sep=**'**\n**'**)  
   except FileNotFoundError as e:  
    print(**"Error: The AppData.dat file was not found."**)  
    print(**"Please create AppData.dat before continuing."**)  
    print(e, e.\_\_doc\_\_, type(e), sep=**'**\n**'**)  
   except Exception as e:  
    print(**"There was an error. Additional error info:"**)  
    print(e, e.\_\_doc\_\_, type(e), sep=**'**\n**'**)

# Summary.

Assignment 07 called for modification of a program that uses pickle and custom error handling. Pickle is useful for reconstructing objects. Custom error handling is useful for making a program more usable by handling unexpected user input and making the user experience smooth.