Kelsey Kawaguchi

Nov. 24, 2019

IT FDN 100

Assignment 07

<https://github.com/kelseyk9/IntroToProg-Python-Mod06>

Exception Handling and Pickling

Introduction

This paper explains the logic behind how I developed the exception handling and pickling code. The goal and purpose of the assignment was to research error handling and pickling, then writing code and an explanation to display my understanding of the lesson. I created two separate bits of code: one that applies pickling to record a student’s name and GPA and another that exercises error handling.

Methodology - Pickling

Figure 1 contains the code that demonstrates my understanding of pickling. For this exercise, I collect a student’s name and GPA.

The very first part of the code “import pickle” is to identify that the code will be called out in the code to serialize and de-serialize an object in the python structure.

The next section identifies what each variable’s meaning (ex. text file, string, etc.).

Next, I have defined two functions, one of which will save the serialized data input by the user into a text file and the other to read the de-serialized data from the text file back to the user.

The save\_data\_to\_file function calls on the text file that is identified in the variables and the list of data that will be input by the user. The first action on line 23 is for the objFile to open in “wb” mode (write binary). On line 24, “pickle.dump” allows for the list of data to be serialized into the objFile. To follow, line 25 closes the text file.

The read\_data\_from\_file function reads first calls to open the file on line 34 as identified by “rb” which means read binary. Line 35 then identifies the variable of list\_of\_data which calls for the objFile (serialized) to become de-serialized per the callout of “pickle.load”. The function then returns the de-serialized data (which is readable by the user).

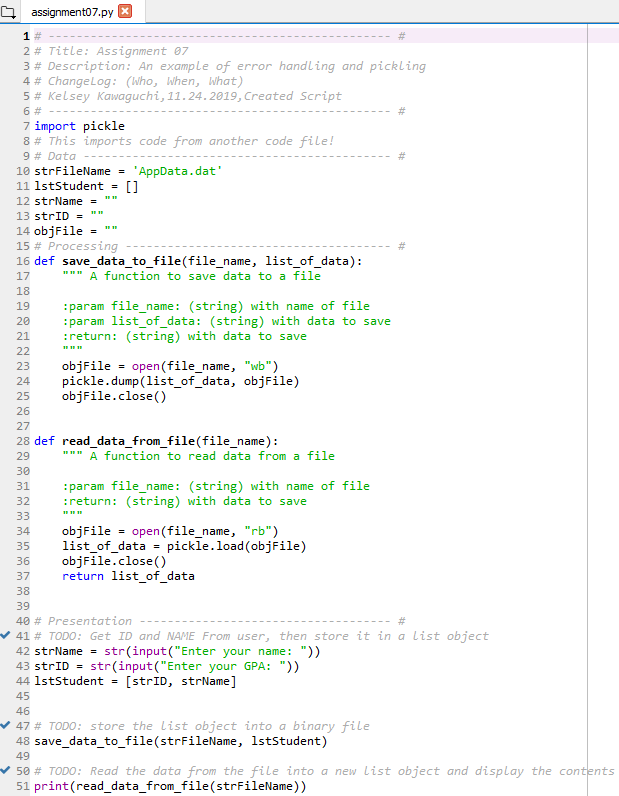


Figure 1 – Pickling Code

The next part of the code is the presentation section, where the user is prompted to enter a student’s name and GPA, which both become part of a list.

Next, the save\_data\_to\_file function is called to serialize and save the data that was input by the user to the text file.

Lastly, the code prints the de-serialized data from the text file based on the read\_data\_from\_file function.

Command Results – Pickling

Figure 2 shows the working code in the command window view. As seen, the code successfully records the data and reads the de-serialized data back to the user as expected.

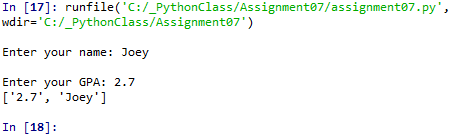


Figure 2 – Pickling Command Window Result

Figure 3 is a screen shot of the assignment 7 folder structure to prove that the AppData.dat text file was created in the same folder as the code.

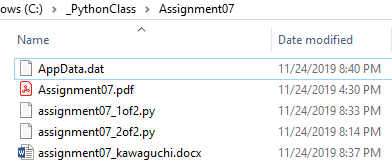


Figure 3: Proof of the AppData.dat location

Figure 4 shows the data stored in the AppData.dat file in the pickled, serialized mode, which is made of random characters.



Figure 4: AppData.dat serialized data

Methodology – Error Handling

Figure 5 displays code prompts the user to input the amount of remaining days in the year until Christmas, and calculates how many weeks remain until Christmas. If done successfully, the result that is printed to the user will be the amount of weeks left until Christmas with 2 decimal places.

The error handling message created comes up when the user enters a string of data, rather than an integer. In order to figure out what to call the error, I first had to test out the code by entering a string of data into the prompt, which then prompted me with an error titled “ValueError”. Based on the callout, I used the ValueError as a callout, and when the ValueError in the code occurs, the response in the command window will alert the user that the program will only work if an integer is entered.

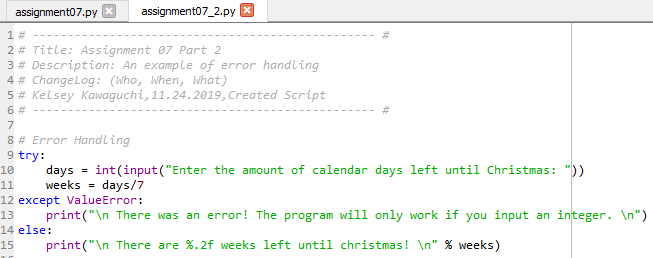


Figure 5 – Processing Functions Pt 1

Command Results – Error Handling

Figure 6 is the command window that prompts the user to insert how many days are left until Christmas. The program then prints the amount of weeks left until Christmas with 2 decimals places.

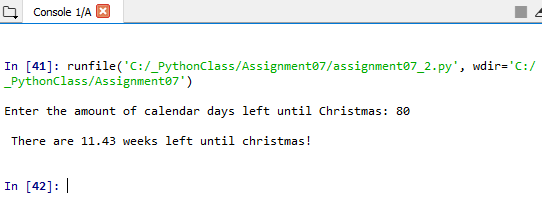


Figure 6 – Error Handling Command Window

Summary

In Conclusions, I learned how to research different python functions, then was able to create practical code that displayed my understanding of both pickling and error handling.