Kenji Petrucci

8/22/2022

IT FDN 110 B Su 22: Foundations of Programming: Python

Assignment 7

https://github.com/Mekong84/IntroToProg-Python

**Assignment 7 Write Up**

**Introduction**

Assignment 7 is to demonstrate knowledge learned by searching the internet for pickling in Python and structured error handling.

**Pickling**

The term “pickling” in Python refers to “serializing” objects into byte streams. This is especially useful when saving data to disk or transmitting data over a network connection. It allows for the retrieval of python objects from saved files on a disk or network. There are many uses for this but one such use related to this class is to be able to read in and manipulate lists/tables that have already been created. Many of our examples so far had us write data to files saved on our individual machines. This method of pickling allows us to read in and populate dictionary objects, for example, from a previously saved file.

**Structured Error Handling**

Structured error handling is a useful method to mitigate potential errors caused by user input (it can also be a useful tool to help determine what types of errors are being generated while building the code). As demonstrated with the class example, the automatic Python error handling can be cumbersome and may not provide much useful information to the user. Structured error handling provides a more effective way to capture and display error data. This can be used to inform the user of a way to avoid the error in the future.

**Examples**

For this exercise I used a previous assignment as a template. The pickling process is handled in the processing section of code where we define functions governing the behavior of reading and saving the data to file. It’s fairly straightforward with the exception that the standard “r” for read changes to “rb” to signify reading a binary file (this is the same for all modes; “wb” for write, “ab” for append, etc.). To pickle the data we use the dump() function and to unpickle the data we use the load() function. Loading pickled data this way results in loading the previously created object (in this example it’s a list of data).

The error handling is done in the IO section of code which we’ve modified from a previous example. In this example we use a menu of options to show saved data, add new data, and save new data. It’s here we call the previously defined functions and establish alternate behavior if any number of errors are encountered. For this example, I want the ID category to be whole integer numbers and the NAME category to be letters only and contain no numbers. Also, in the event the user tries to display data before any data has been saved there is a catch to inform the user the file does not yet exist.

Detailed code is included that shows the method of handling these errors but at a high level I used the try-except method to govern normal behavior and handle errors that may arise. The try-except block allows for the execution of the code and provides a mechanism to define specific behavior in the event of an error. It also allows targeting of specific error types; a divide-by-zero error may drive a different program behavior and message to the user than a file-not-found error, for example.

This assignment calls for documentation of the script running in **both** PyCharm and the OS command window. Figure 1 below shows screen shots of the program running in PycCharm. Program executes from left to right --------------------------------->

Graphical user interface, text

Description automatically generated

*Figure 1, Assignment07\_Kenji.py running in PyCharm*

Figure 2, below, shows a screen shot of the data captured in the .txt file.

Text

Description automatically generated with medium confidence

*Figure 2, AppData.txt*

Figure 3, below, shows the script running in the Windows command window. Program executes from left to right. ------------------------------------------------------------------------------------->

Graphical user interface, text

Description automatically generated

*Figure 3, Assignment07\_Kenji.py running in the Windows command window*

It can be seen here that the file, previously saved when running the script in PyCharm, is read in and creates the object when run from the CMD window. Different instances of running the script using different methods shows successful pickling/unpickling of the data.

**Summary**

In this example we defined functions to pickle and unpickle data, implemented a try-except block to allow for structured error handling, and successfully demonstrated the ability to save and load previously defined Python objects spanning multiple program sessions.