George Biehl

August 23, 2020

IT FDN 110 A Su 20: Foundations of Programming: Python

Assignment07

# Exception Handling and Pickling

## Introduction

**The purpose of this assignment is to locate and discuss good explanations of Python exception handling and pickling found on the Internet. A script is then used to demonstrate how both can be used.**

## Exception Handling Research

A programmer envisions how the program will interact with external sources such as a user, another program or module, or a file. As the program actually runs, however, the interactions may not occur as planned. A user might mistype an entry or enter a name when asked for a number. A module or file may not be available. When such errors occur and the program stops, it’s very useful to see specific details.

I looked at some online explanations of exception handling. One site (<https://www.pythonforbeginners>) (2020) (External Site) began in a way I found a bit humorous – “When you think you have code which might produce an error...” I would think only the most over-confident program would believe in perfect code! Later, the same site was too simplistic. Asking the user to enter a number between 1 and 10, the short example code responded well to a non-numeric entry but would not have detected a number outside the range of 1 through 10. I didn’t like that this site needlessly presented problematic code. The example could have simply asked for a number and still made is point about handling a non-numeric entry.

The next site (<https://www.geeksforgeeks>) (2020) (External Site) was better for my purposes. In addition to keep the examples simple, this site discussed a couple of exception handling statements which might be useful to me. The “raise” command forces a specific exception to occur which might be useful, for example, in troubleshooting a program. Another new keyword I learned about was “finally” which always executes as part of the “try” statement.

## Pickling Research

Serialization, I learned, is the process of converting a data structure into a stream or bytes which can then be put on disk or transmitted in some other way (<https://realpython.com>) (2020) (External Site). Python supports three modules for serializing and deserializing – marshal, json, and pickle. Marshal is the oldest and its use is no longer recommended. Json is useful for compatibility with other languages. It also can be read by humans. The pickle module is recommended for use in all cases where json is not specifically needed.

It seemed clear to me that data serialized with one of the above three modules would need to be deserialized with that same module. The next site I visited also stated that even different versions of pickle might not be compatible (<https://datacamp.com>) (2020) (External Site). For example, data pickled with version 2 might not be able to be unpickled if using version 3. One would need to install version 2 in order to unpickle the file,

## Demonstration Script

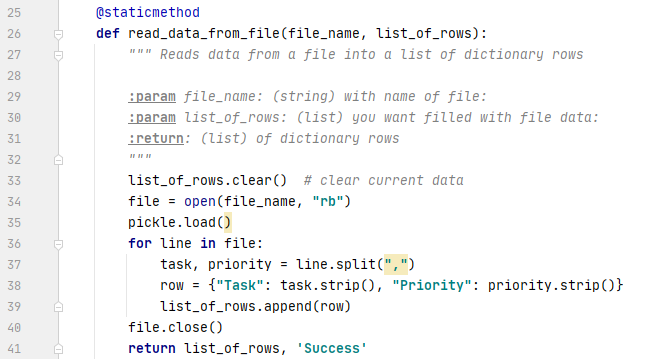
To demonstrate exception handling and pickling, I attempted to convert last week’s script to include exception handling and pickle. Last week’s assignment used functions extensively. One of the main points made was that the function was an abstraction, i.e., the main program did not depend on how that function worked. I reasoned that to save and load files I would only need to adjust the two function that did those tasks. Figure 1 shows the adjusted code for reading the binary file.

Figure 1: Adjusted code for reading a binary file

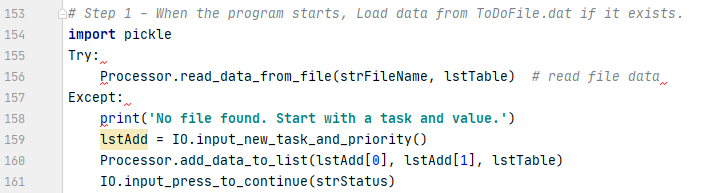
At the start of the main script last week’s program began by reading the file. This week there was no assumption that the file existed so it seemed a good place to demonstrate the use of try/except. In particularly, I hoped for the FileNotFound exception. Shown below is the adjusted code for the start of the main script. Of course it includes “import pickle.”

Figure 2: Main script start with importing pickle and testing if the file exists

## Summary

I couldn’t get these adjustments to work. They seemed so simple. Now I’m out of time and need to hand this in. On to Assignment 08!