*David Jamieson*

*8/22/20*

*IT FDN 110 A Su 20: Foundations Of Programming: Python*

*Assignment 07*

*GitHub:* <https://github.com/djamies1/IntroToProg-Python-Mod07>

**Assignment 07 – Documentation**

**Summary:**

The purpose of this assignment was to incorporate python pickling and structured error handling into a basic application, which can store user input into a .dat file using binary and then read the data back to the user if prompted. Pickling is a built-in function of python which allows the user to take a python object and de-serialize it into smaller bytes, which can then be easily consumed by a different python application. My understanding is this is largely used for improved processing speeds when transferring large amounts of data or objects between python applications. Structured error handling is a process of catching some sort of scripting malfunction that would cause your program to crash, and instead routing it to an error exception route. There are lots of different ways to do this, but for this assignment I used a very simple exception catch that simply reads back to the user that there was an error, and what the basic python error message was.

**Scripting:**

For the scripting piece of this assignment, I used much of the same logic used from assignment 6 (see assignment 06 documentation). Throughout my scripting I defined a number of functions that would prompt the user for input, save the cumulative user’s input into a binary file using pickling, or read the saved data back to the user from their file.

Fig1 below shows the pickling scripting piece of this assignment, defined in functions.

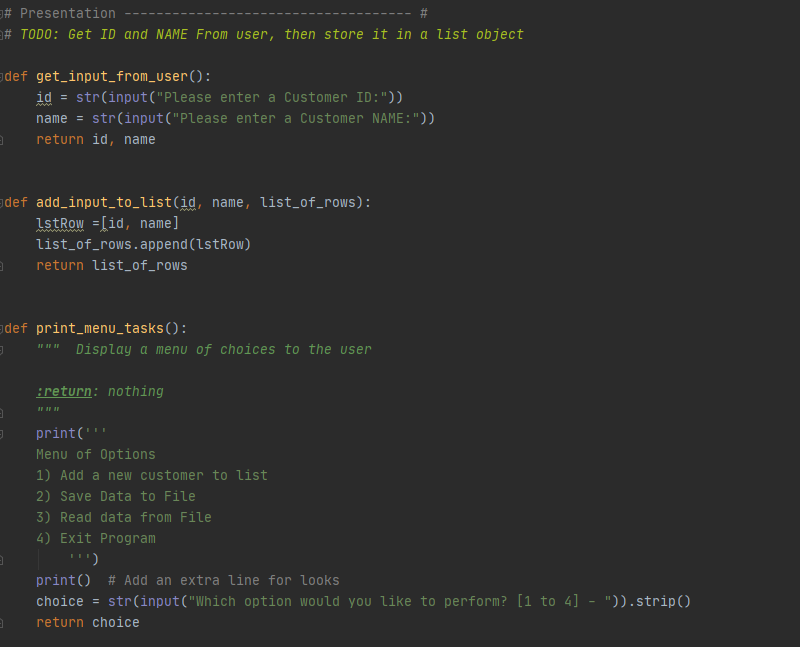
Fig1:

Save\_data\_to\_file – this takes the user’s current cumulative list of customer Ids and names which they have defined in the application and writes over the current file with their data. It does not add to the existing data in the file, simply overwrites the file with new data. This uses pickling to “wb” which means write binary and stores the data in a .dat file.

Read\_data\_from\_file – this reads whatever is currently stored in the .dat file back to the user. This could be from a previous application run or from their current run, depending if they have already saved the file. It uses the “rb” open option, which means read binary.

The second piece of this scripting assignment was to prompt the user for input on a customer ID and name with which they could slowly add to their list and then save it down to the binary file. Fig2 below shows the presentation functions used in this assignment.

Fig2:

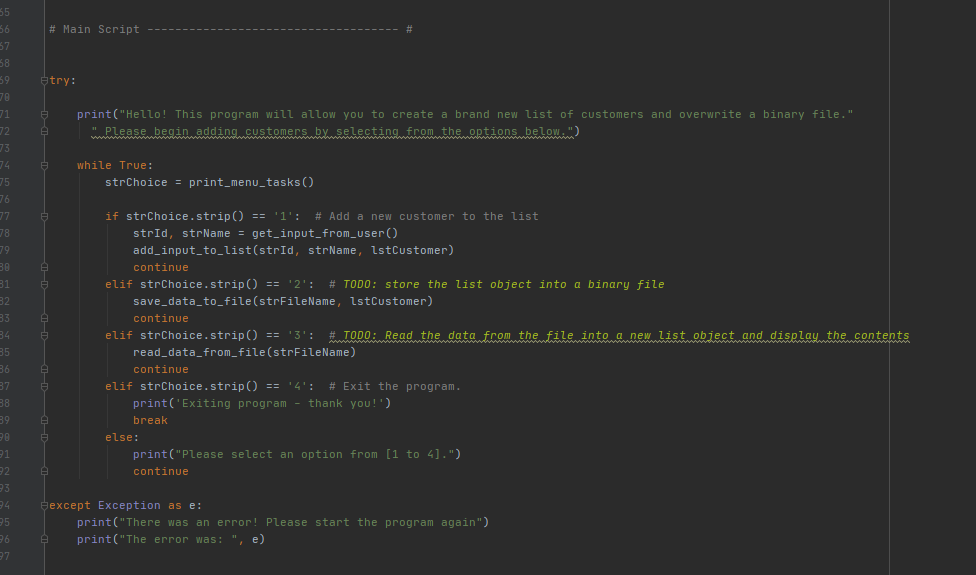


Get\_input\_from\_user – this is a very simple function which prompts the user for a customer Id and name.

Add\_input\_to\_list – this is a very simple function which takes the input provided in get\_input\_from\_user function and then appends it to the existing customer list.

Print\_menu\_task – this function prints a list of options for the user to select from (1 to 4), which is used to determine which of the previous functions to process, based on the users input.

The last piece of this scripting was to combine all these functions together to create an actual application with which the user can interact with. This is accomplished in Fig3 below. The script prompts the user for their input, then runs one of the functions depending on their selection. There is also some basic error handling, so if the user somehow manages to break the script during the execution, it will move to the “except” portion of the code and prompt them with the error message.



The above shows the full script in action, accessing the various functions that were defined above. It also includes error handling for any potential application errors.

**Conclusion:**

Pickling is a great way to de-serialize your data and allow them to be processed more efficiently across python applications. Structured error handling allows you to build catches into your code to prevent users from breaking an application flow unintentionally (or perhaps intentionally). This assignment allowed us to build on many of the previous learning about python, combining functions with the new concept of pickling and then building in structured error handling on top of everything.