Inderjot Brar

March 21, 2021

IT FDN 110 B

Assignment 07

<GitHub\_url>

ASSIGNMENT 7

**Introduction**

This assignment involves understanding how error handling and pickling works in Python. To demonstrate error handling, I created a script that requests users to input their name and age and uses try/except block, built in and custom error exceptions, and exception classes to handle any script errors. To demonstrate pickling, I added code to the script to save the user input data in binary format.

## **Exception Handling**

A python script will come to a sudden halt whenever it encounters an error. An error might be related to incorrect syntax which can be resolved by correct the script statements; or an error might result from attempting to execute a syntactically correct statement, resulting in an “exception”.

Python has a number of built-in exceptions, such as “ZeroDivisionError”, “FileNotFoundError”, “ValueError” etc. In python, exceptions can be handling using “try/except” block. The section of the script which can raise an exception is within the “try” clause and the code to handle exception is written in the “except” clause. A programmer can also incorporate their own custom exceptions using “raise” block and can use “finally” clause to execute a section of the script regardless of whether the “try/except/raise” block raises an error or not.

## **Pickling**

Pickling is a way to convert a python object (list, dictionary, etc.) into a byte data to store it in a file or database. The idea is that this data contains all the information necessary to reconstruct the object in another python script.

In order to work with a binary file in Python, “Pickle” module needs to be imported. The statement “pickle.load()” can be used to read a binary file and “pickle.dump()” can be used to write and append to the file. “rb”, “ab”, and “wb” are the 3 methods to read, write and append to a binary file respectively. However, it is important to remember pickled data is not encrypted.

## **Exception Handling & Pickling in the script:**

1. Raising exception for an undesirable input: In the script, users are requested to input their name and age. The script raises an exception if the name is numeric or if the age is not numeric. (See Figure 1 for the code and Figure 2 for the exception raised.)

Graphical user interface, text

Description automatically generated

***Figure 1: Script using custom and built-in exceptions***

Graphical user interface, text, application, email

Description automatically generated

***Figure 2: Custom exception raised.***

1. Using Pickling to read from a binary file and building in exception handling to write if no such file exits: In the script, a function is used to open the binary file in read “rb” mode and load the existing data using “pickle.load()”. If such file does not exist and a built-in exception “FileNotFoundError” occurs, the except block creates a new binary file by opening in write “wb” mode and create an empty list to store user input. (Figure 3)

Graphical user interface, text, application

Description automatically generated

***Figure 3: Reading from Binary file with custom exception handling for a built- in exception***

1. Using Pickling to store data to a binary file: Once user input is received, a list is created and added to the binary file using “pickle.dump()”. (Figure 4)

Graphical user interface, text, application, email

Description automatically generated

***Figure 4: Writing to Binary file***

## **Summary**

In summary, exception handling is a great way to prevent a program from crashing mid-run. It also allows programmer to display the error messages in a user-friendly phrasing rather than relying on system error messages. Lastly, we also learnt that pickling can be used to store data in a binary format for easy Pickle is a way to store large objects of data into binary files.