# **Ian Sanchez**

# **Nov 26, 2019**

# **IT FDN 100 A**

# **Assignment07**

Exception Handling and Pickle

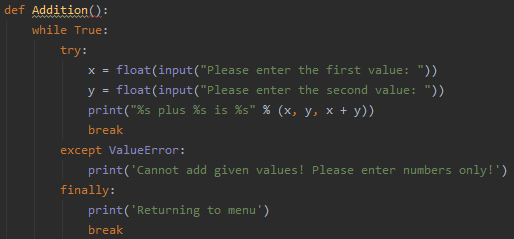
Introduction

For this week’s assignment, we were instructed to research exception handling and the pickle module in Python. We also needed to create a script to demonstrate both exception handling and pickle.

Exception Handling

For demonstrating exception handling, I created 2 simple math functions. One for addition and one for division. The script asks the user for input but will only work if the values entered are *floating point values* (Fig. 1) I wrapped the simple math script into a try-except block for exception handling should an error occur.

**Figure 1. Addition**

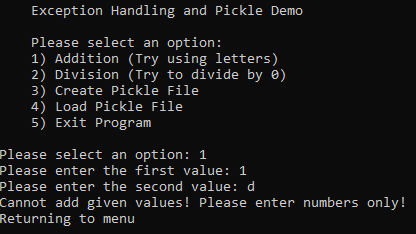


We user the try-except block so that the script tries to execute the script and if there is an error it will go to the exception clause. I added ‘ValueError’ to the except clause so that it only executes when that specific error happens. Otherwise you can add an ‘else’ to catch an error that was not specified.

Demonstration of Exception Handling

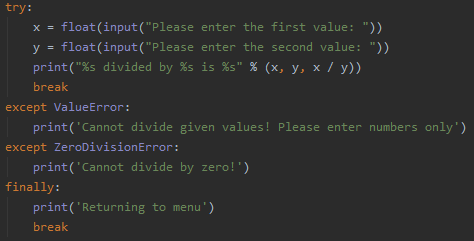
Let’s see what it looks like when we run the code. In for the Addition part we have an except clause to catch ‘ValueError’ which will occur if anything other than a floating point value is entered by the user. (Fig. 2)

**Figure 2. ValueError example**



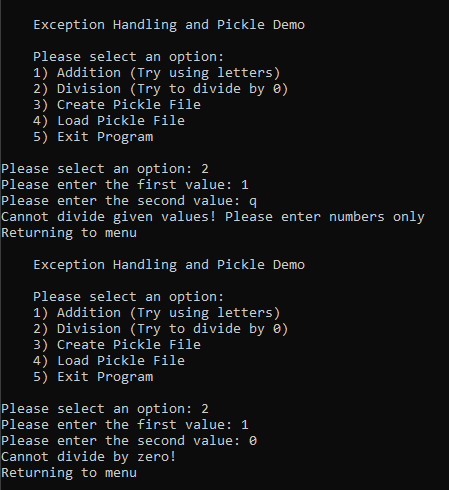
In the Division part of the script, I added an except clause for ‘ZeroDivisionError’ as well. (Fig.3)

**Figure 3. Division**



Now let’s see how it works in the script. It catches the ‘ValueError’ and ‘ZeroDivisionError’. (Fig. 4)

**Figure 4.**



I’ve found that these articles explain exception handling really well. It helping me understand and demonstrate how to user it properly for this assignment:

**Exception Handling Websites:**

<https://www.tutorialspoint.com/python3/python_exceptions.htm>

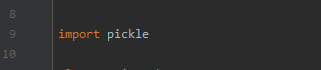
<https://docs.python.org/3/tutorial/errors.html>

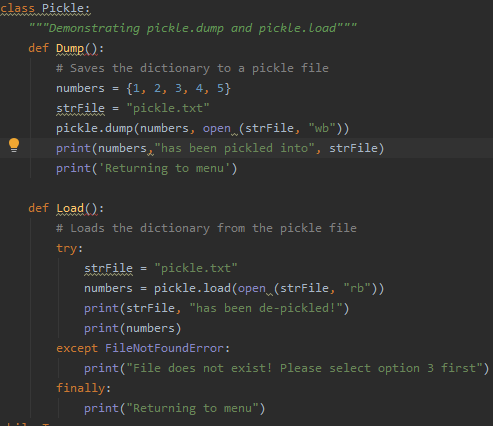
Pickle Module

Through research the pickle module is mainly used for serializing and de-serializing Python objects using binary. When you pickle an object using *pickle.dump* it becomes basically unreadable to humans. We can read a pickled file using *pickle.load* where it will de-serialize the file.

For demonstrating the pickle module, I wrote a basic script to pickle/de-pickle a dictionary to/from a text file. (Fig 5). Before we can use pickle, we need to import the pickle module which I have entered at the beginning of the script.

**Figure 5. Pickle script**

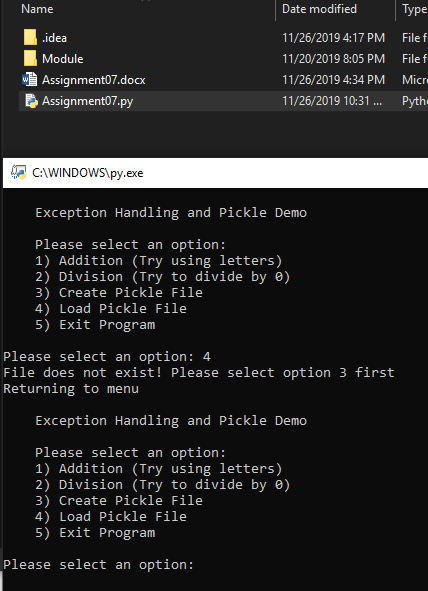




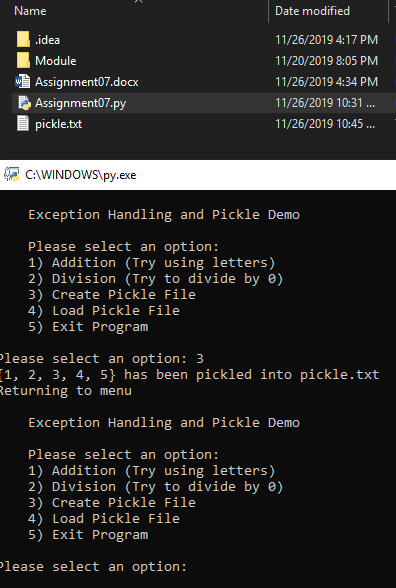
Now let’s see what happens when it’s executed in the program.

I try to run option 4 before option 3 to load or read a pickle file. Since there is no file, I have an except clause to catch the ‘FileNotFoundError’. (Fig. 6) When we select option 3, informs the user that the dictionary has been pickled. When we open the file, the characters look like it doesn’t make any sense. (Fig. 7, 7.1)

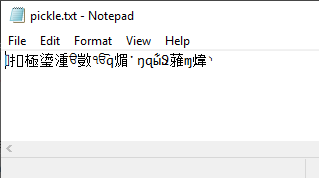
**Figure 6.**



**Figure 7. Dictionary pickled to file**

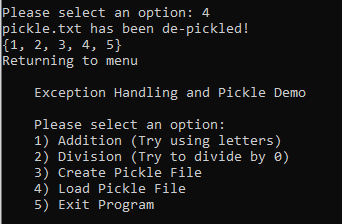


**Figure 7.1 Pickle file**



Now when we select option for it should display back to us what is in the text file. When we open the file from the folder its random characters, but in the program it shows what the actual data is in the file.

**Figure 8**



**Pickle Module Websites:**

<https://wiki.python.org/moin/UsingPickle>

<https://www.geeksforgeeks.org/understanding-python-pickling-example/>

<https://docs.python.org/3/library/pickle.html>

Summary

In this assignment I was able to research and demonstrate exception handling and the pickle module. I learned how to properly use the try-except block for exception handling and how to properly use the pickle module.