Laura Truong

May 30, 2020

Introduction to Programming

Assignment07

GitHub Repository: https://github.com/laurat-tech/ITFnd100-Mod07

Error Handling and Picking

# Introduction

Software developers are creative people who automate repetitive tasks and strive for efficiency. While writing code, one may improve productivity by using error handling and picking. This week Students were given some flexibility to do research on exception handling and pickling and create a new script that demonstrates how these concepts work. I created two scripts. One script called Assignment07 that demonstrates both error handling and pickling while another script titled Assignment07 Error Handling that covers Error Handling with numbers and uses a few other exception handling methods. This paper will cover error handling and pickling. I will then run the script in a command shell from the Windows OS using Python version 3.

# Error Handling

Error handling is used to guard against potential failures in code that would usually cause the program to exit. During my research, I found this site to be particularly helpful: <https://www.tutorialspoint.com/python/python_exceptions.htm>. There are a variety of errors a programmer may encounter; this site lists all the types of errors and gives a brief explanation on what this error is. It then gives code examples that outline how to handle the error and defend the code. The tip that I found useful from this website is that a try statement can contain multiple *except* statements in the case that a try block contains statements that have multiple types of exceptions. In my example, I created a code that asks a user to enter a number to find its reciprocal in decimal form. I created 3 exceptions as seen in **Figure 1.** The value error exception does not allow the user to enter string characters or anything that is not a number. ZeroDivisionError does not allow a user to enter a number less than 1. Last of all, Exception prints out any other exception that the end user may have gotten that is not displayed or handled in the code. In **Figure 2,** I ran the code in pycharm and tested with a 0. As seen, my exception tells the end user that they cannot enter a number and divide by 0. To run the script in an OS command, I opened a command prompt by typing “cmd” into the search bar of my desktop. I typed “Python "C:\\_PythonClass\Assignment07\Assignment07 Error Handling.py" into the command prompt providing the file location to open up the file. **Figure 3** shows the result of my program being run in the cmd line.

Figure 1

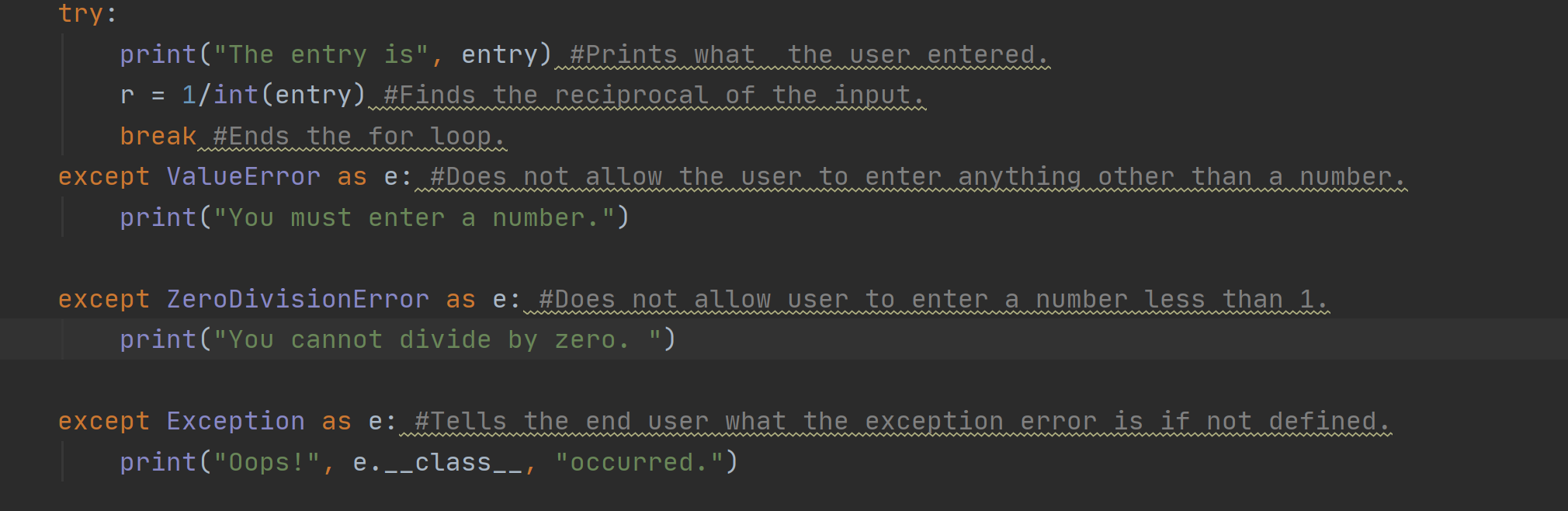


Figure 2

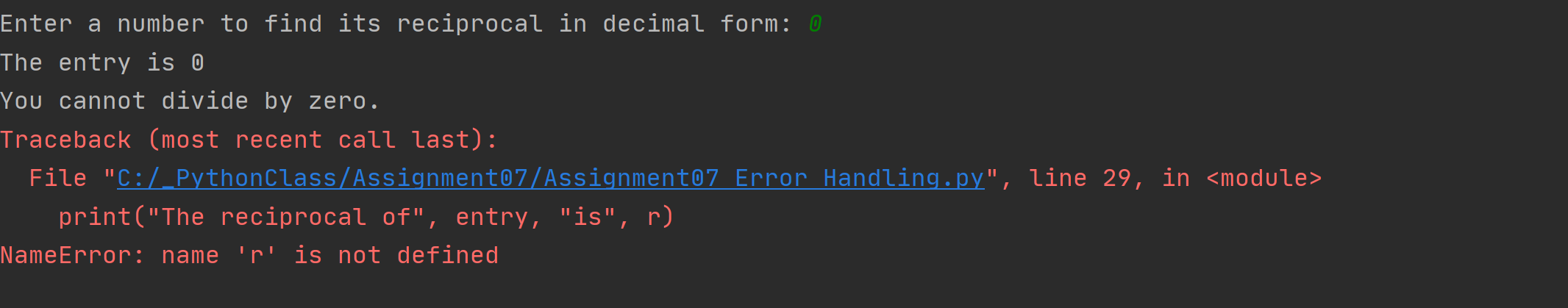
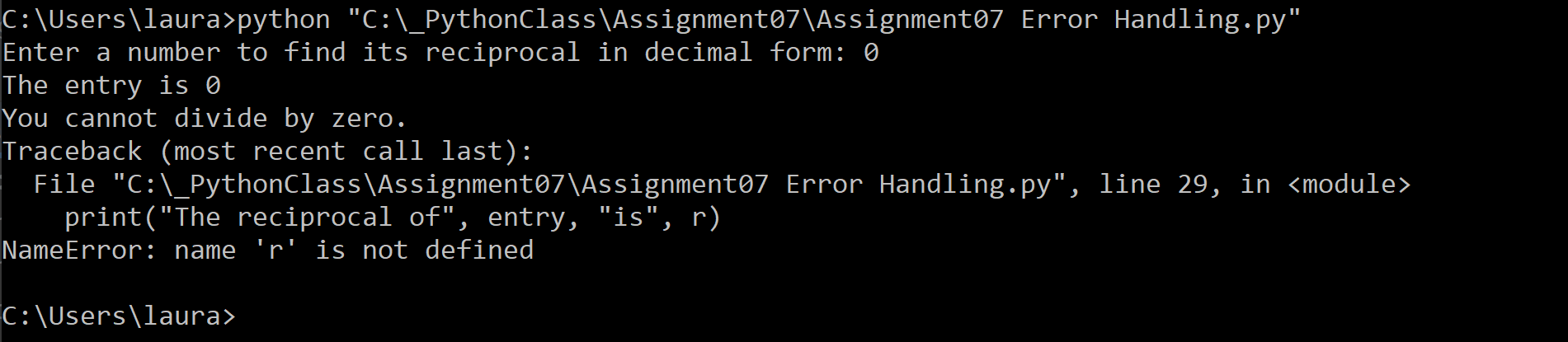


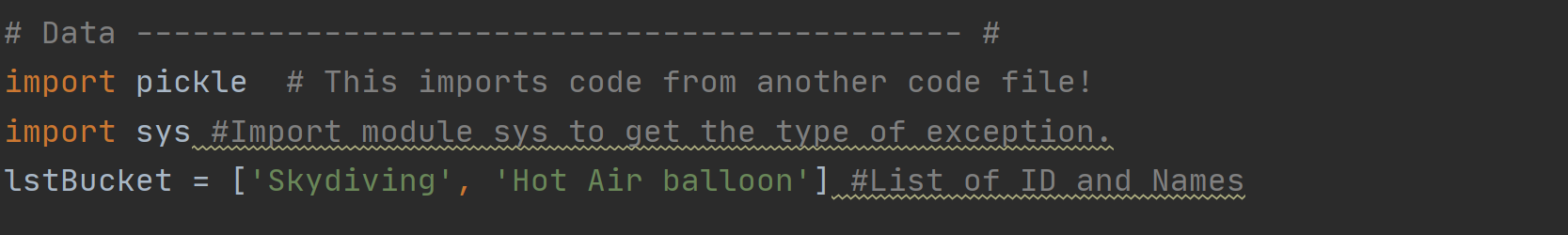
Figure 3



# Pickling

Pickling is a serialization and deserialization module that converts python objects to byte streams and back again. People will generally use pickling when they have a large data set that they want to read. The read-in process tends to take a while, pickling will process this data a lot faster. When a pickled file is read, it can often be misunderstood as being encrypted due to the binary conversion but it is by no means encrypted. A site that I found extremely helpful in explaining this concept is <https://python.readthedocs.io/en/latest/library/pickle.html>. The explanation on this site was straightforward. Not only does it show how to serialize data, it shows how to de-serialize To demonstrate pickling, I created a code based off of example 7-1 from the Assignment 7 Starter Code. In my code, I ask the end user to enter something they would like to add to their bucket list and pickle it. Using the example outline from class, I started my code shown in **Figure 4.**

Figure 4



**Figure 5** shows the result of the pickling. I also went ahead and added an exception to this script for when a pickle is unpickled. The code is shown in **Figure 6.** The unpicklingerror exception is used when there is a problem unpickling an object that may be because data was corrupted or changed whether intentionally or unintentionally.

Figure 5

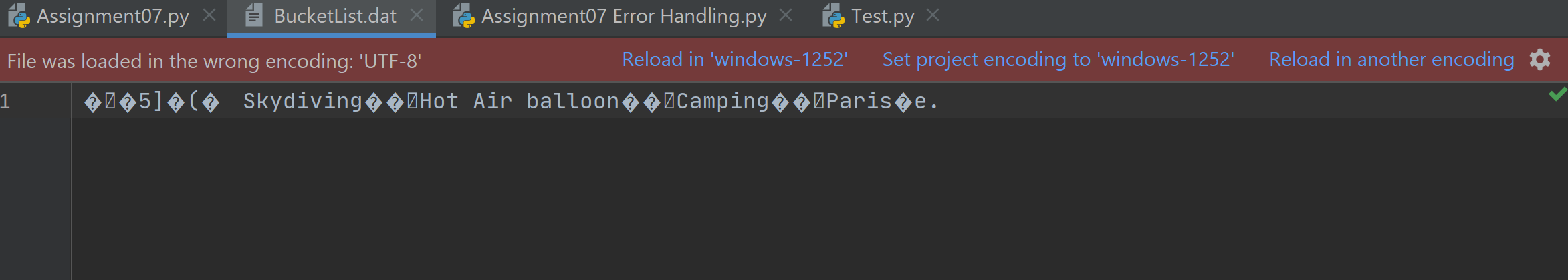
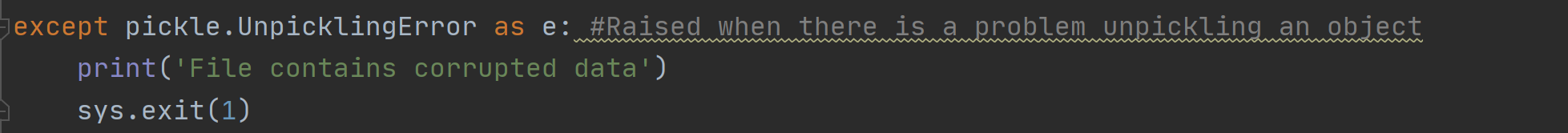
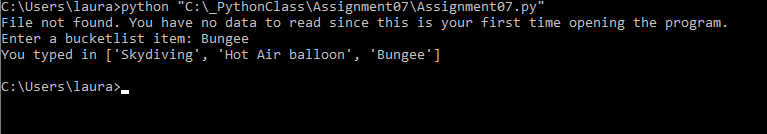


Figure 6



I can run my script in the command prompt the same way that I ran my Assignment07 Error Handling script by pasting the file path into the command prompt. **Figure 7** shows the result of running my script in the command prompt.

Figure 7



# Summary

This paper covered error handling and pickling and showed the code run from Pycharm as well as the Command Prompt. Through Error handling, we can prompt or prevent the problems that end users may see before they see it. Through pickling, we can flatten our code to load it more quickly when it needs to be accessed. As a software developer it is important to use error handling, pickling, and other methods where applicable to create the most efficient code for end users as well as those reusing your code.