Name: Malea Sullivan

Date: 22 August 2023

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

GitHub URL: https://github.com/msully12/IntroToProg-Python-Mod07

Assignment 07: Files and Exceptions

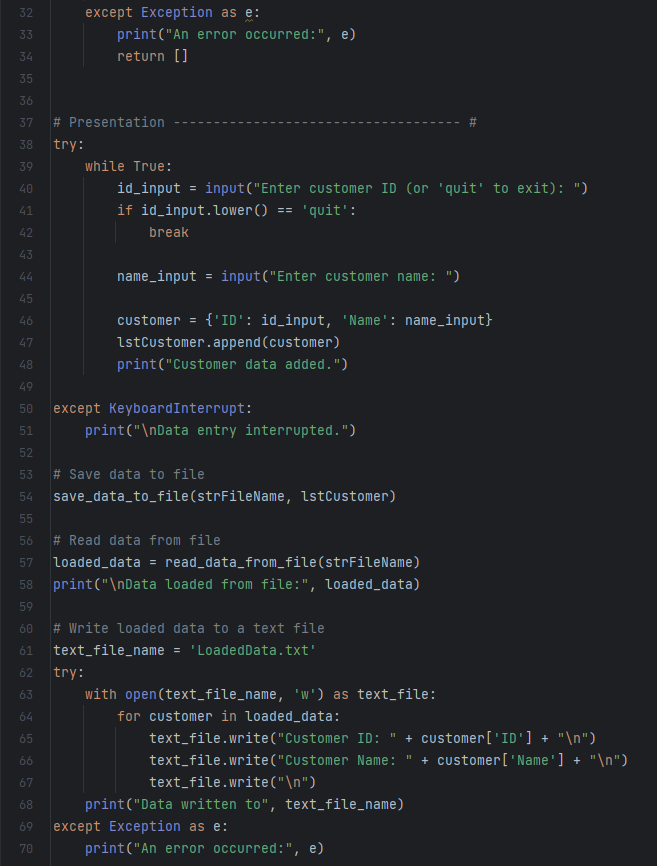
Introduction:

For the seventh assignment in this class, we learned pickling, and exception handling. Pickling in python is primarily used in serializing and deserializing a python object structure. In other words, it is the process of converting a python object into a byte stream to store it in a file/database. In the script that I created demonstrating pickling in python, I utilized the starter script that was given for this assignment and asked the user for a customer ID and a name and saved that information in a binary file and a text file. To demonstrate exception handling, I created another python script that asked for the user to input two numbers, and the script would divide those two numbers, however, if the user asked to divide by zero or input a non-numerical digit, an exception would be created alerting the user that an error has occurred.

The script that I utilized for pickling:

A screenshot of a computer program

Description automatically generated

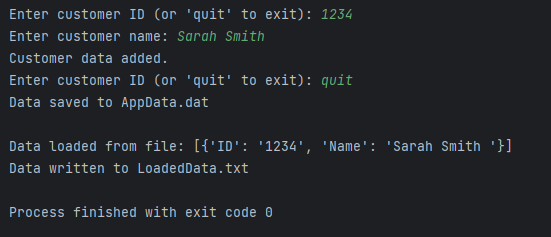


I will input an image of the script I utilized to display exception handling after I explain how the pickling script works. I started with my script utilizing the import pickle. This line imports the python pickle module, which provides functionality for serializing and deserializing python objects. The next line named the file where the data will be saved and loaded. In this case, it’s ‘AppData.dat’.

Saving data to a file is something we have done a lot in this class. The difference here is that I utilized pickle.dump to write the list of customer data to a file. If successful, a success message will be displayed, and if an exception occurs during the process, it prints an error message.

Reading data from a file is also something we have done in this class, but again, the difference here is that I utilized the pickle.load to read data from the file. From here, my script is like the scripts I have written in the past. I utilized the while True loop that prompts the user to input a customer ID and a name until they enter ‘quit’. Once the user enters ‘quit’, the data is stored using ‘pickle’ for serialization and deserialization. It saves the data to a binary file, loads the data back, and writes the loaded data to a human-readable text file.

Running the code I created in pycharm was successful:



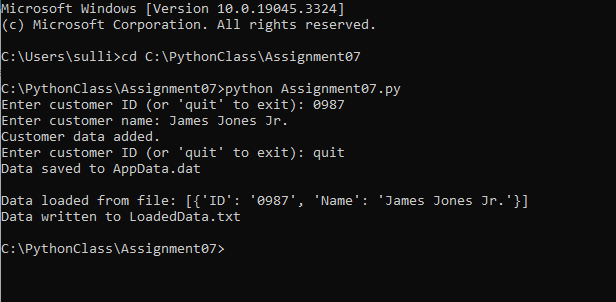
The binary file that was created:



The human-readable text file that was created:



Running this in the command shell was also successful:



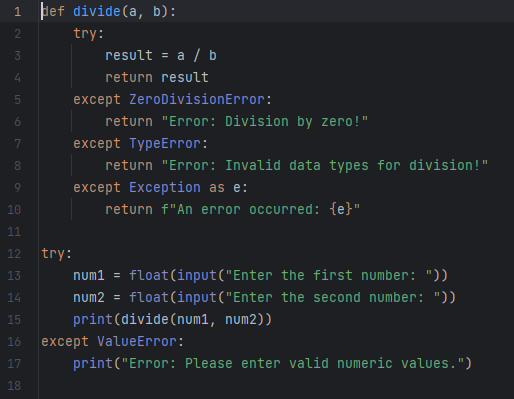
The binary file that was created:



The human-readable text file that was created:

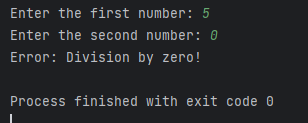


Now, I will show the code I utilized to display exception handling:



For this script, I wanted to demonstrate very simply how exception handling works, and there is really no better way than when someone tries to divide by zero. I listed three types of except blocks to catch specific types of exceptions that might occur during division. The first was the ‘ZeroDivisionError’. This exception is raised when attempting to divide by zero. The second exception was ‘TypeError’. This exception is raised when the data for num1 and num2 are not compatible with division. The third exception was the ‘Exception’. This is a catch-all exception that captures any other exceptions that might occur during the division process.

The script ran successfully:



Summary:

Overall, this assignment was similar to what we have done in the past. Exception handling was something I was already inputting in my scripts. It is important to note that the catch-all exception should be listed last if you are using multiple exceptions. Pickling was relatively easy to understand. I didn’t have a lot of trouble with it. The most confusing part was understanding if I was actually doing it correctly. When I first viewed the binary file, I wasn’t really sure if it was binary. I kept thinking that I was doing something wrong. It is important to note that when utilizing pickling it is important to consider security implications when loading data from untrusted sources, as the ‘pickle’ module can execute arbitrary code during deserialization.