

Tim Randall

May 30th, 2023

Intro to Programming (Python) – IT FDN 110A Spring 2023

Assignment 07

<https://github.com/trandallUW/IntroToProg-Python-Mod07>

<https://trandalluw.github.io/IntroToProg-Python-Mod07/>

Python Pickling and Exception Handling – an Introduction

Pickling Introduction

As a developer or data analyst using Python, you might have a situation where you want to “serialize” your data. “Serialize” is a technical term for converting the data into a linear form, or a series of bytes, so that it be easily stored, sent over a network, or used by other data and software functions. To do this, you can use the pickling function.

How to Pickle

The pickle module consists of four methods:

1. `pickle.dump (obj, file)` – use this to pickle your data into a file.
2. `pickle.dumps(obj, file)` – use this to pickle your data and output a string.
3. `pickle.load()` – use this unpickle your data and read a file.
4. `pickle.loads()` – use this to unpickle your data and output a string.

Below is an example set of code that uses a set of classes and functions. The purpose of this code is to collect a user ID and name from the user, store the data into a binary file, then read the data, and output the data back to the user in a a list format. Notice how I used the `import pickle`, `pickle.dump`, and `pickle.load` code to convert the data.

```
#----- #
import pickle # This imports code from another code file

# Data----- #
strFileName = 'UserInfo.dat'
lstCustomer = []
strUserID = None
strUserName = None
lstNewCustomerInfo = None

# Input/Output----- #
def get_user_ID ():
    user_ID = input("What is your user ID?: ")
    return (user_ID)
```

```

def get_user_name ():
    user_name = input("What is your user name?: ")
    return (user_name)

# Processing----- #
def save_data_to_file(file_name, list_of_data):
    objFile = open(file_name, "wb") # open file, write to binary
    pickle.dump(list_of_data, objFile) # pickle the data into file
    objFile.close() # close the file
    pass

def read_data_from_file(file_name):
    file = open(file_name, "rb")
    userInfo = pickle.load(file)
    return(userInfo)

# Presentation----- #
# Get ID and NAME From user, then store it in a list object
strUserID = get_user_ID() # get user ID
strUserName = get_user_name() # get user Name
lstCustomer = [strUserID, strUserName]

# Main Body of Script----- #
# Store the list object into a binary file
save_data_to_file(strFileName, lstCustomer)
print("User ID and Name pickled into a file! ")

# Read the data from the file into a new list object and display the contents
lstNewCustomerInfo = read_data_from_file(strFileName)
print("Reading from file...This is the data that was unpickled: ")
print(lstNewCustomerInfo)

```

What does the binary pickled data look like?

Here is the output of the script using Pycharm (see Figure 1). The data is stored in a list. Figure 2 shows my command shell output.

What if the `save_data_to_file()` function didn't execute properly? Well, use the "try" and "except" function to print out a potential error message for this situation.

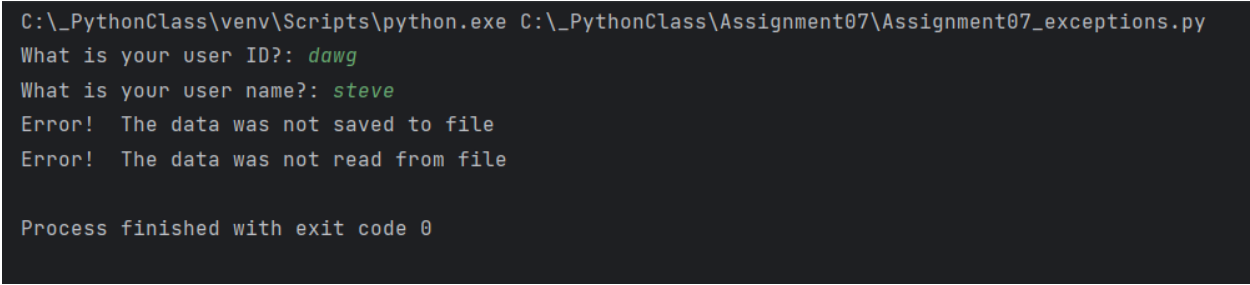
Or what if the `read_data_from_file()` function got all messed up by your coworker? Well, that's a great situation to plan for and put in an error message.

Here's the same "Main Body Script" code used in my pickling example with some exception handling – notice the `try` and `except` code.

```
# Main Body of Script----- #
# Store the list object into a binary file
try:
    save_data_to_file(strFileName, lstCustomer)
    print("User ID and Name pickled into a file! ")
except:
    print("Error! The data was not saved to file ")

# Read the data from the file into a new list object and display the contents
try:
    lstNewCustomerInfo = read_data_from_file(strFileName)
    print("Reading from file...This is the data that was unpickled: ")
    print(lstNewCustomerInfo)
except:
    print("Error! The data was not read from file ")
```

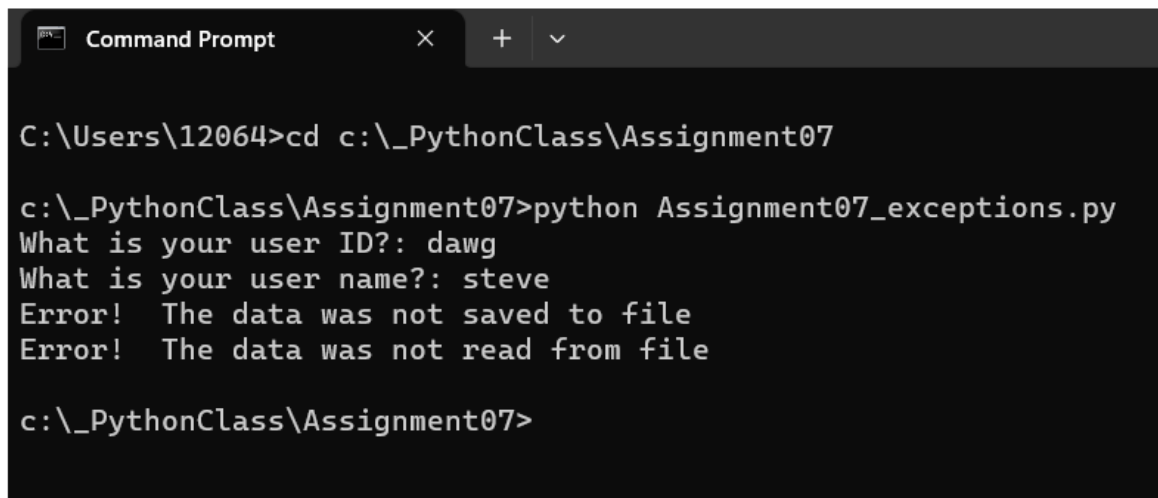
Figure 3 shows my Pycharm output for these error messages. Figure 4 shows my output in command shell.



```
C:\_PythonClass\venv\Scripts\python.exe C:\_PythonClass\Assignment07\Assignment07_exceptions.py
What is your user ID?: dawg
What is your user name?: steve
Error! The data was not saved to file
Error! The data was not read from file

Process finished with exit code 0
```

Figure 3 - Error Pycharm



```
C:\Users\12064>cd c:\_PythonClass\Assignment07

c:\_PythonClass\Assignment07>python Assignment07_exceptions.py
What is your user ID?: dawg
What is your user name?: steve
Error! The data was not saved to file
Error! The data was not read from file

c:\_PythonClass\Assignment07>
```

Figure 4 - Errors Command Shell

Summary

Pickling is a great way to convert your data in order to “serialize” it and make it easier to store or send over a network. Exception Handling is a foundational way to develop custom error messages that are easy to understand by humans. Both Python functions are foundational for any programmer or data analyst to know. There are great resources on the internet to find out more info on these functions.