

Name: Brittany Vesel

Date: March 2, 2023

Course: Foundations of Programming: Python

GitHub URL: https://github.com/bdvesel21/IntrotoPython_Mod07

GitHub Website: https://bdvesel21.github.io/IntrotoPython_Mod07/

Assignment 07 – Files and Exceptions

Introduction

This week in Foundations of Programming: Python we focused on using the incorporated pickling function in Python, writing and reading to binary files, and creating custom exception messages. The homework was also more involved this week as we had to research our own explanations online for how certain aspects of the program work.

Pickling

Pickling is a processing module included in the Python language that packs data into a simple binary file of 0s and 1s. This helps store large data sets or databases in a smaller format and can be packed and unpacked on many different machine types and systems. The first step of our assignment was to find a webpage we liked that we thought explained the process well. I chose <https://www.tutorialspoint.com/python-pickling> because not only did it use simple examples to help understand the concepts, it included potential errors for the process in case I ran into them creating my own program. In this way, it helped me prepare better for the assignment and what errors I could see when I moved on the exception handling part of the assignment.

Figure 1: Pickle Display Code

```
Pickling Data Example
import pickle

# Data ----- #
FileName = 'Books.dat'
lstBooks = []

# Pickling ----- #
book_id = int(input("Enter an ID: "))
book_title = str(input("Enter the title of a book: "))
book_author = str(input("Enter the name of the author: "))
lstbooks = [book_id, book_title, book_author]
print(lstbooks)
print()

# Store the Pickled list
objFile = open("books.dat", "wb")
```

```

pickle.dump(lstbooks, objFile)
objFile.close()
print("List Pickled")
print()

#Retreive the Pickled List
print("Unpickling List")
objFile = open("books.dat", "rb")
book1 = pickle.load(objFile)
objFile.close()
print(book1)
print()

```

After watching the lectures and reading the course book, I decided to try to build my own pickling code using a list based on books, shown in Figure 1 above. First, I imported the pickle module at the top of the program. I then asked the user for an ID number, a book title and an author name, then stored that data as a list in memory. This was code we have been using for many weeks now, but it was fun to configure it for my own interest. Next in the code came the file writing via the pickling method. I opened the file using “wb” for binary write, dumped the list inside the file, then closed the list before displaying a message to the user. Once that was completed, I wrote the program to continue by displaying a message to the user that it would unpickle the list. I opened the file using the binary read feature, assigned the created line to the book1 variable, and then closed the file. I then displayed the book1 list values to the user to finish off the pickle display method. The output of the pickle display is shown in Figure 2.

Figure 2: Pickle Display Output

```

Run: Assignment 07 x
C:\Users\bdves\AppData\Local\Microsoft\Wind
Enter an ID: 1
Enter the title of a book: Tribute
Enter the name of the author: Nora Roberts
[1, 'Tribute', 'Nora Roberts']

List Pickled

Unpickling List
[1, 'Tribute', 'Nora Roberts']

Process finished with exit code 0

```

Exception Handling

Handling exceptions is when the programmer tries to predict what potential problems could result from the code or user input and try to either negate them or explain as the program runs for the user. The second part of this week's assignment started with researching exception handling in Python and creating custom error messages. For exception handling, I actually liked the Python documentation (<https://docs.python.org/3/tutorial/errors.html>), as it not only gave explanations and examples of potential errors, it also linked to a complete list of errors you could see and read about.

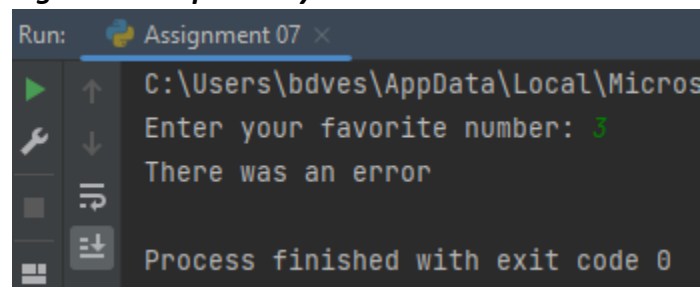
Figure 3: Exception Handling Code

```
Error Handling ----- #
try:
    str(input("Enter your favorite number: "))
    print(favorite_number)

except NameError as e:
    print("Variable 'favorite_number' not found")
    print("Python error explanation: ", e)
```

After watching the lectures and reading the book, I picked the NameError exception from the list in the textbook to display in the exception handling part of my code. I initially wrote code asking for the user to input their favorite number but didn't assign it to a variable. When I asked the program to print that number to the user again, it returned an exception. Figure 3 shows the code I initially wrote, and Figure 4 is the output in Pycharm.

Figure 4: Output in Pycharm



```
Run: Assignment 07 x
C:\Users\bdves\AppData\Local\Micros
Enter your favorite number: 3
There was an error
Process finished with exit code 0
```

Combining Pickling and Error Handling

The final assignment was to combine the two into one program. I cut my code into pieces and tried to implement a potential exception on each part. To display a different exception that was also in the text book, I chose to create an exception handling block for if the user input a non-integer character for the ID. I chose to display my own customer error, and then add a second line to include the Python Exception Explanation, shown in Figure 5.

Figure 5: Initial User Input Exception Handling

```
try:
    book_id = int(input("Enter an ID: "))
```

```

book_title = str(input("Enter the title of a book: "))
book_author = str(input("Enter the name of the author: "))
lstbooks = [book_id, book_title, book_author]
print(lstbooks)
print()
except Exception as e:
    print("There was an error! Please check that you are using integers for IDs.")
    print("Python Exception Explanation: ", e)

```

Next, I copied the code to save the list into a pickled binary file. I added the try and except block around that code to display if there was an error saving the file. I don't know if there are potential errors with this but thought it would be best to include it anyway. This code is shown in Figure 6.

Figure 6: Pickling the List with Exception Handling

```

try:
    objFile = open("books.dat", "ab")
    pickle.dump(lstbooks, objFile)
    objFile.close()
    print("List Pickled")
    print()
except Exception as e:
    print("Python Exception Explanation: ", e)

```

The last part of the program was to create exception handling around the reading of the pickled binary file. This was displayed in Randall Root's notes, so I did use his code as a reference to create my own exception handling here. I added a FileNotFoundError exception to display text to the user, as well as a general exception error in case something else went wrong. Figure 7 shows the code for error handling while retrieving a pickled list, while Figures 8 and 9 show the code running without exceptions in Pycharm and the Command Prompt.

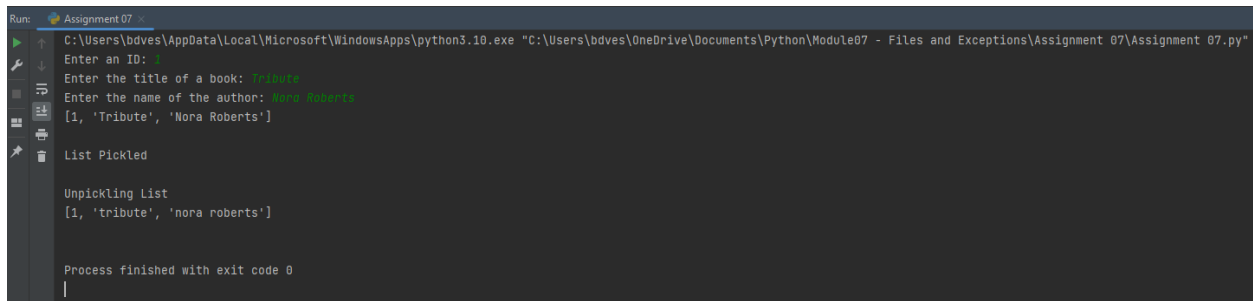
Figure 7: Retrieving a Pickled List with Exception Handling

```

try:
    print("Unpickling List")
    objFile = open("books.dat", "rb")
    book1 = pickle.load(objFile)
    objFile.close()
    print(book1)
    print()
except FileNotFoundError as e:
    print("The file must exist before reading it!")
    print("Python Exception Explanation: ", e)
except Exception as e:
    print("There was an error!")
    print("Python Exception Explanation: ", e)

```

Figure 8: Output in Pycharm



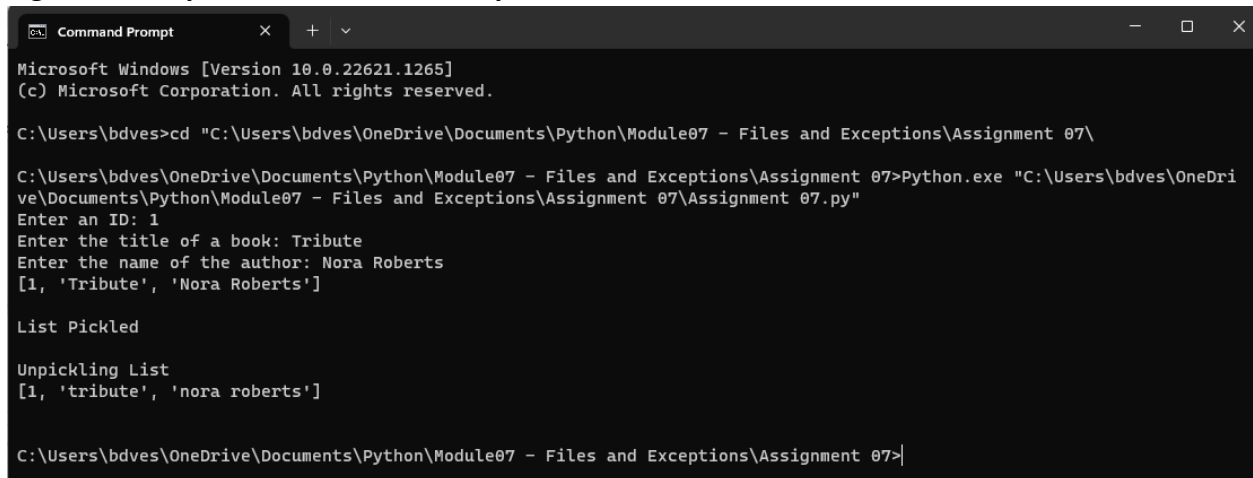
```
Run: Assignment 07
C:\Users\bdves\AppData\Local\Microsoft\WindowsApps\python3.10.exe "C:\Users\bdves\OneDrive\Documents\Python\Module07 - Files and Exceptions\Assignment 07\Assignment 07.py"
Enter an ID: 1
Enter the title of a book: Tribute
Enter the name of the author: Nora Roberts
[1, 'Tribute', 'Nora Roberts']

List Pickled

Unpickling List
[1, 'tribute', 'nora roberts']

Process finished with exit code 0
```

Figure 9: Output in Command Prompt



```
Microsoft Windows [Version 10.0.22621.1265]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bdves>cd "C:\Users\bdves\OneDrive\Documents\Python\Module07 - Files and Exceptions\Assignment 07\"

C:\Users\bdves\OneDrive\Documents\Python\Module07 - Files and Exceptions\Assignment 07>Python.exe "C:\Users\bdves\OneDrive\Documents\Python\Module07 - Files and Exceptions\Assignment 07\Assignment 07.py"
Enter an ID: 1
Enter the title of a book: Tribute
Enter the name of the author: Nora Roberts
[1, 'Tribute', 'Nora Roberts']

List Pickled

Unpickling List
[1, 'tribute', 'nora roberts']

C:\Users\bdves\OneDrive\Documents\Python\Module07 - Files and Exceptions\Assignment 07>
```

Conclusion

This week's assignment was slightly more difficult because we had to start the code on our own and assimilate knowledge from not only the textbook and lectures, but what we could research online about the topics. I didn't have many wrong turns in the code this week not working, because I was modifying code examples from Randall Root's notes and the internet. If anything, my most common mistake is forgetting to put the print strings in quotations and forgetting colons after certain commands like try or else. Luckily, Python is very good at explaining what I missed on those occasions. As I obtain more experience writing Python, I hope those little errors go away and I move on to trying to debug bigger issues with my syntax.