Chiyo Miyake

2023.03.01

IT FDN 110 A

Assignment 07

https://github.com/bread-buddies/IntroToProg-Python-Mod07

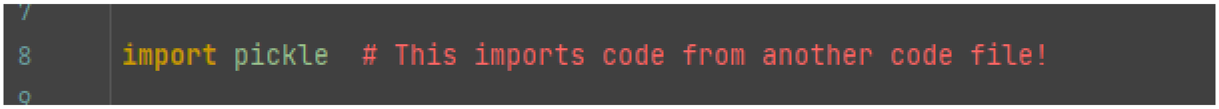
Assignment 07: Examples of Pickling and Exception Handling

Introduction

In this assignment, we learn about pickling and exception handling, also known as error handling.

Pickling

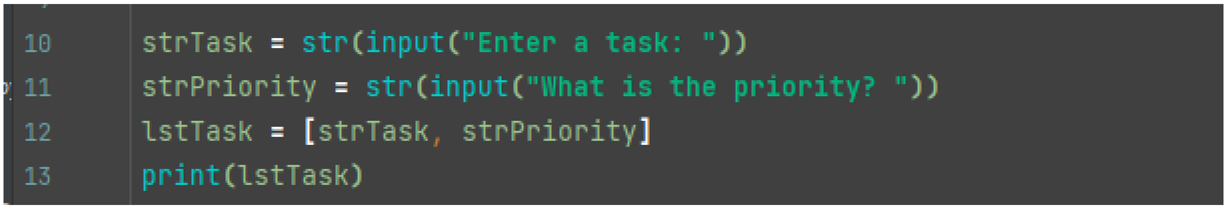
Pickling is a way of converting data into binary format. The binary format is a way to obscure the file content and reduce the size of the file. To use this method of formatting data, you first import the function (Figure 1).



```python
import pickle  # This imports code from another code file!
```

Figure 1: Importing the pickling function into our code.

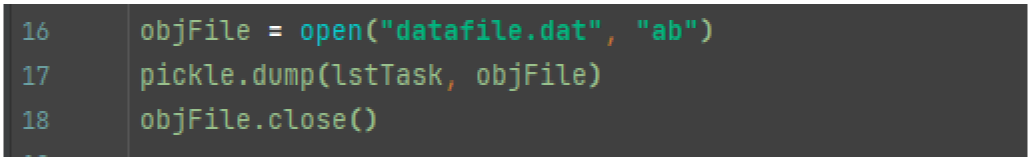
We can then create some data by obtaining some user inputs and store that into a list (Figure 2).



```python
strTask = str(input("Enter a task: "))
strPriority = str(input("What is the priority? "))
lstTask = [strTask, strPriority]
print(lstTask)
```

Figure 2: Creating a list with user input data.

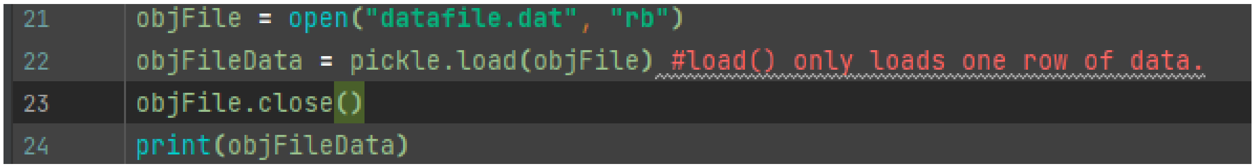
We then use the pickle.dump() function to 'dump', or store the data into a file (Figure 3).



```python
objFile = open("datafile.dat", "ab")
pickle.dump(lstTask, objFile)
objFile.close()
```

Figure 3: Using the built in function pickle.dump() to record data.

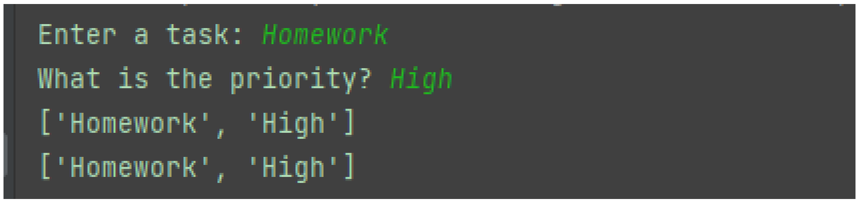
We can read the data by using the pickle.load() function to read one row of data (Figure 4).



```
objFile = open("datafile.dat", "rb")
objFileData = pickle.load(objFile) #load() only loads one row of data.
objFile.close()
print(objFileData)
```

Figure 4: Reading one line of data in the file.

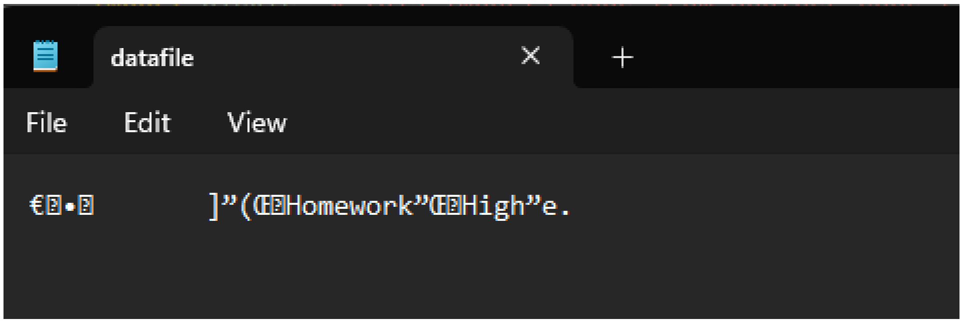
When we run the code, the output shows us the list before it gets pickled, and outputs the single row of data in the data file (Figure 5).



```
Enter a task: Homework
What is the priority? High
['Homework', 'High']
['Homework', 'High']
```

Figure 5: Output of the Pickling Example.

In the text file, it will show the user input as obscure data (Figure 6).

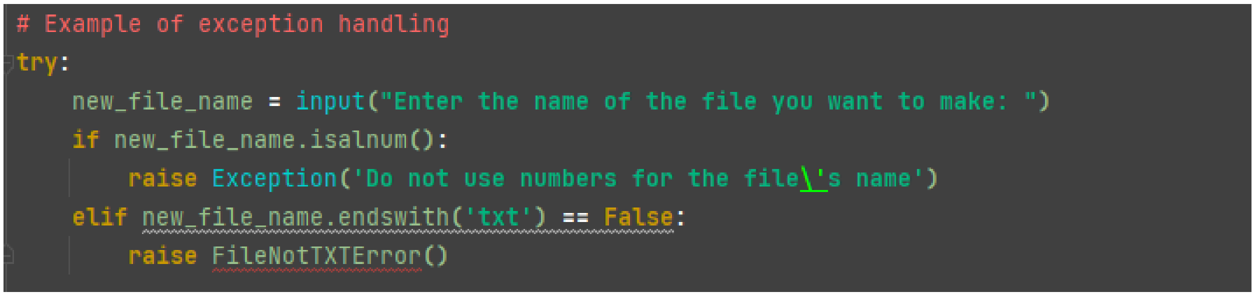


```
€□•□        ]”(Œ□Homework”Œ□High”e.
```

Figure 6: Text file with the user data.

Exception Handling

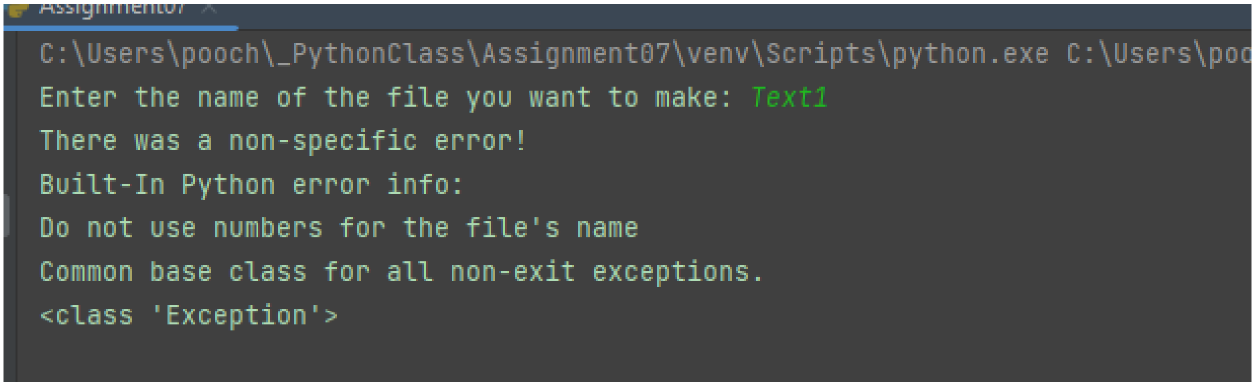
I struggled with the exception handling part of this assignment, so I did my best to use the example from the module to help guide me through understanding it. In exception handling, it helps to organize our errors in the code by streamlining what information we want to know. The first part of the exception handling is the ‘try’ block where we try the code by asking for a user input (Figure 7). If the user inputs an alpha-numeric title, it will ‘raise’ the error tell the user to not add numbers into the title. The next ‘raise’ exception is if the user fails to add the file format as ‘.txt.



```python
# Example of exception handling
try:
    new_file_name = input("Enter the name of the file you want to make: ")
    if new_file_name.isalnum():
        raise Exception('Do not use numbers for the file\'s name')
    elif new_file_name.endswith('txt') == False:
        raise FileNotTXTError()
```

Figure 7: Try block with raising specific errors.

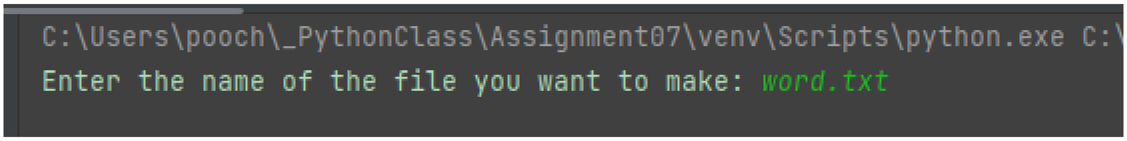
When we make the title alpha-numeric, the program will print the error information (Figure 8).



```
C:\Users\pooch\_PythonClass\Assignment07\venv\Scripts\python.exe C:\Users\poo
Enter the name of the file you want to make: Text1
There was a non-specific error!
Built-In Python error info:
Do not use numbers for the file's name
Common base class for all non-exit exceptions.
<class 'Exception'>
```

Figure 8: User error, added alpha-numeric title as file.

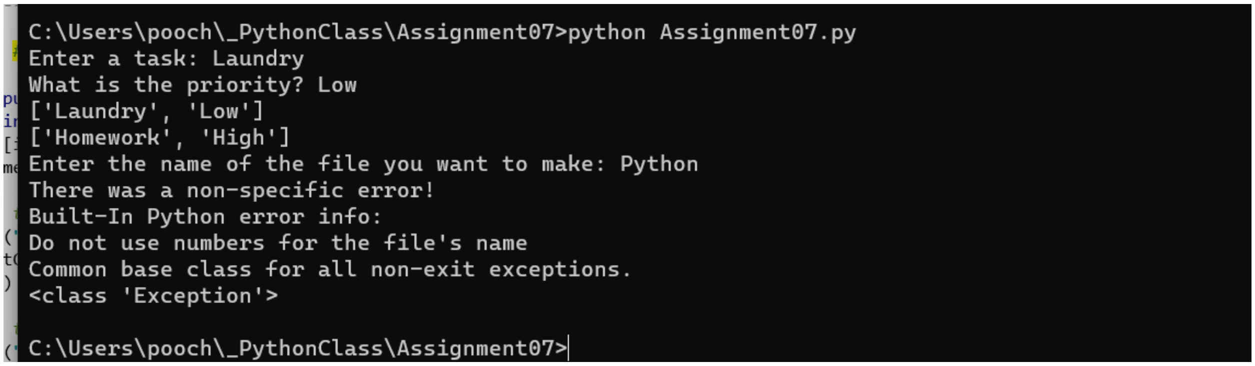
When we successfully create a text file with alphabetic characters, the program creates a new text file (Figure 9).



```
C:\Users\pooch\_PythonClass\Assignment07\venv\Scripts\python.exe C:\
Enter the name of the file you want to make: word.txt
```

Figure 9: Successful program.

We can run both examples on the command prompt (Figure 10). Since the last task to be added to the data file was the Homework task, it will read the first line of the data. And for our exception handling example, this time the error that is raised when creating a new file is not specifying the file type as a text file.



```
C:\Users\pooch\_PythonClass\Assignment07>python Assignment07.py
Enter a task: Laundry
What is the priority? Low
['Laundry', 'Low']
['Homework', 'High']
Enter the name of the file you want to make: Python
There was a non-specific error!
Built-In Python error info:
Do not use numbers for the file's name
Common base class for all non-exit exceptions.
<class 'Exception'>

C:\Users\pooch\_PythonClass\Assignment07>
```

Figure 10: Running the examples in the command prompt.

Summary

In summary, pickling is a good way of condensing file size by obscuring the file data into a binary format. Exception handling is also a good way of managing the errors in a program. Instead of making a user read the complex debugging messages, we can organize and format the specific errors we want to keep track of.