

Germán Gornalusse

Sunday November 28, 2021

IT FDN 110 A Au21: Foundations of Programming: Python

Assignment Module 07: Files and Exceptions

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

GitHub Web Page: <https://germangornalusse.github.io/IntroToProg-Python-Mod07/>

## How to Do Pickling and Exception Handling in Python

### Introduction

In this paper, I will show you two useful concepts in Python programming language. In the first one, I will show you how to write and read files in binary format, a concept called “pickling”. In the second concept, I will demonstrate with an example how to do structured error handling; you will be using try-except block of code. For each section, I included useful websites and appropriate demos. Finally, you will upload both files (PDF, Word and Python scripts) into a GitHub repository and post a GitHub Web page.

For simplicity, I will assume you will be using Windows operating system.

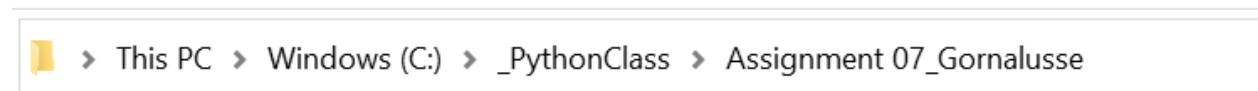
### Step 1. Create a subfolder in your C: Drive\\_PythonClass

The following instructions will allow you to create this subfolder in your hard drive:

**C:/\_PythonClass/Assignment07**

- a) Left double click on “\_PythonClass” folder (to open it)
- b) Right click> New > Folder
- c) Name the folder as Assignment 07\_Yourlastname

I am showing you how the final path to this folder will look like (**Figure 1**):



**Figure 1.** Path to the folder where you will save your Assignment 07. I used my first and last name (“German Gornalusse”) as an example to personalize my subfolder.

## Step 2. Create a new Project in PyCharm

You will create a new project in PyCharm that uses the `_PythonClass\Assignment07_` last name folder as its location. I assume you will have installed PyCharm on your `C:\` drive or on your desktop.

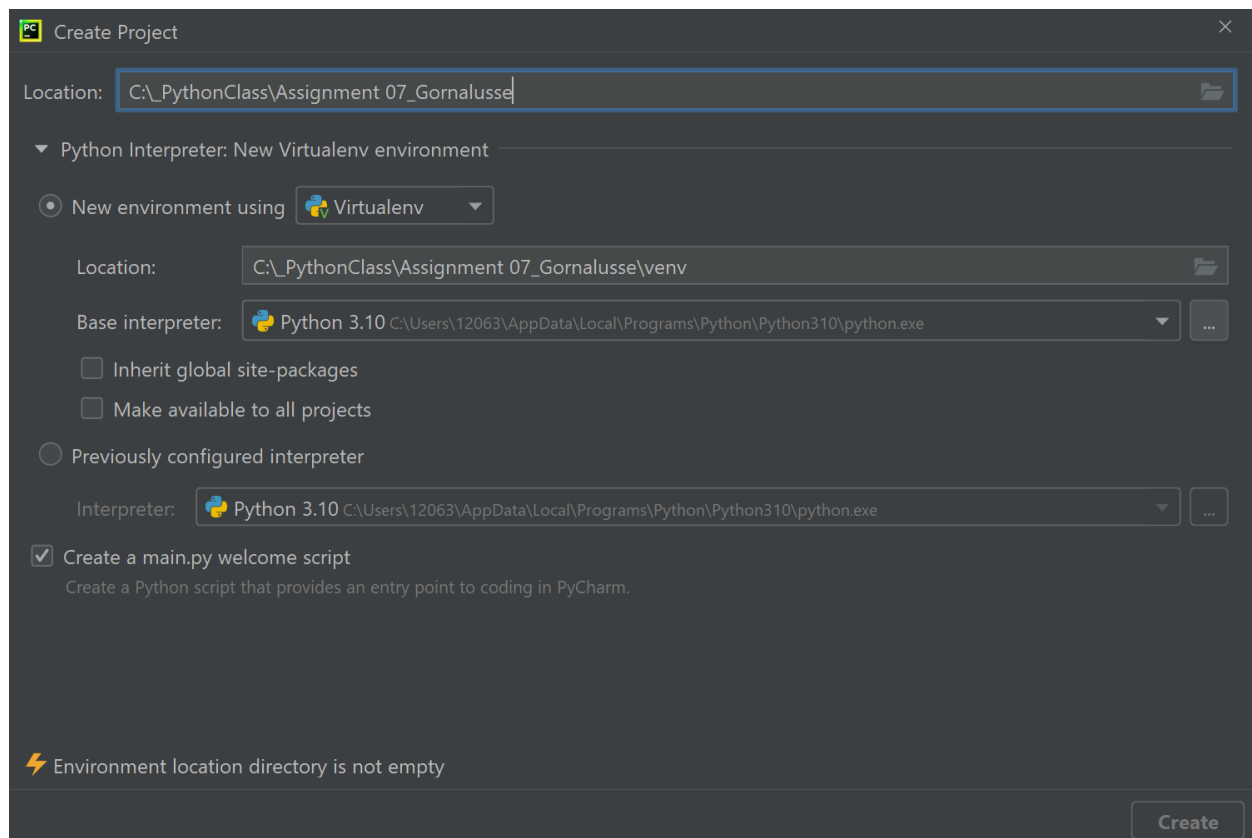
a) Double click the icon “PyCharm Community Edition 2021.2.3”. Mine shows up on my desktop.

b) Select: File> New Project

c) In location type `C:\_PythonClass\Assignment07` to select the file subfolder wherein you will save your project. Alternatively, you can browse the destination folder by selecting the “open folder” symbol at the end of “Location” and manually by browsing and selecting the final folder. [See yellow arrow, on **Figure 2**]

d) Select “New environment using Virtualenv” option. And “Create a main.py” welcome script option. [See orange arrow, Figure 2]. Make sure the Base interpreter is set “Python 3.10” (or the latest version you installed in your computer).

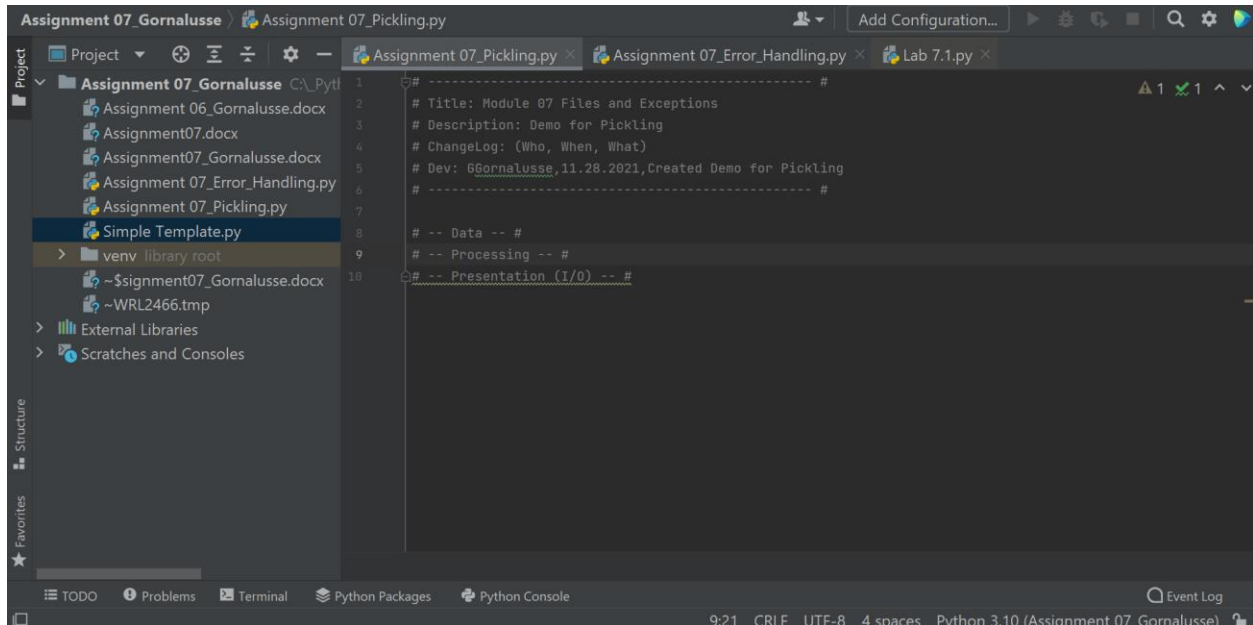
e) Select “Create” (lower right corner of your screen). [**Figure 2**]



**Figure 2** How to create a new project in `C:\_PythonClass\Assignment 07` subfolder using the IDE PyCharm

To do that:

- a) File> Open
- b) Select Assignment 07 subfolder
- c) Select either “This window” or “New Window”. Notice how, on the left-hand side, the “Assignment 07” subfolder shows up. In **Figure 3** I am illustrating this example.



**Figure 3.** PyCharm window showing current folder where your Python scripts for the Assignment07 will be saved

Step 3. Create two Python Scripts in the Project Folder: “Assignment07”: Assignment 07\_Pickling and Assignment 07\_Error\_Handling.

- a) Right click (Context menu) Assignment 07 folder
- b) File> New> Python File > Assignment 07\_Pickling
- c) File> New> Python File > Assignment 07\_Error\_Handling

You will start writing the header and comments, as indicated in the **Figure 3** above. And then, you add the code shown in the demos below.

#### Step 4. How to do Pickling in Python

I included a list of websites and videos that I think they are useful so you can learn how to do pickling in Python:

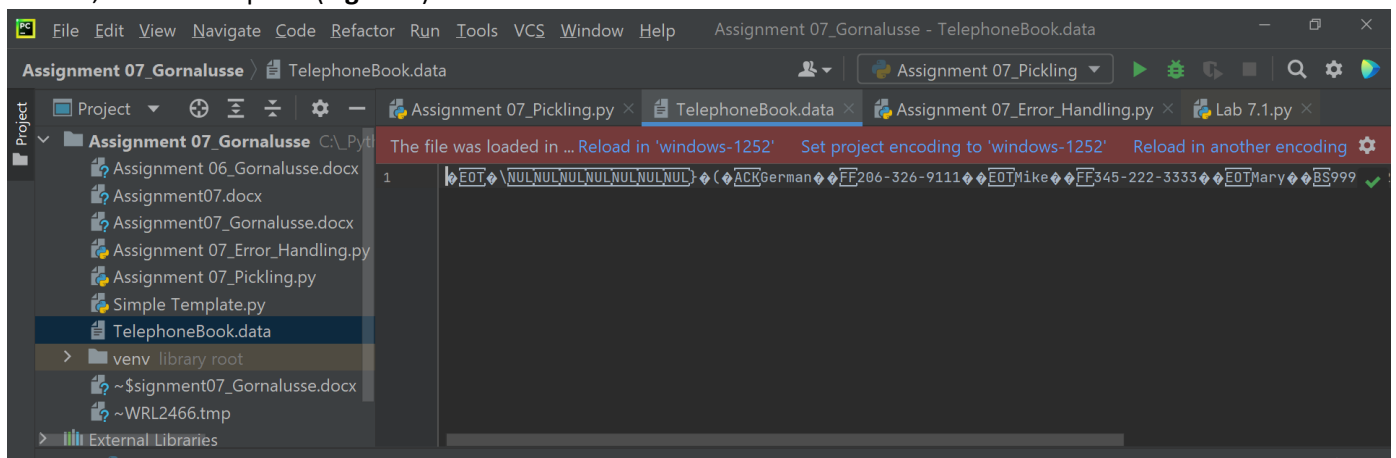
- <https://www.datacamp.com/community/tutorials/pickle-python-tutorial> (external website)

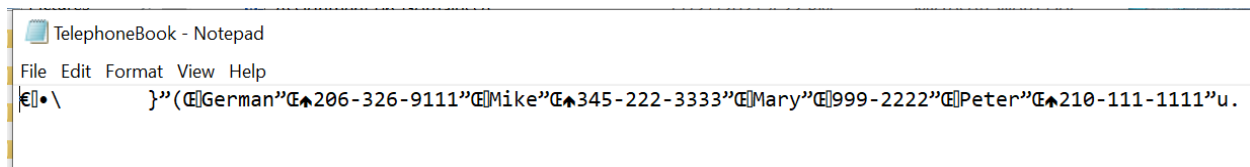
I included this website because I have used Datacamp in the past to learn R programming and I really like how the instructions are stated.

- [https://www.youtube.com/watch?v=D2e3\\_mPhQw0](https://www.youtube.com/watch?v=D2e3_mPhQw0). I chose this website because I only wanted to focus on dictionaries and not in other objects.
- <https://www.youtube.com/watch?v=Xc8Ss9JG2Pw> I liked that this video incorporates the pickling multiple objects and handling error (try/except module), which we will cover in the next section.

The steps that are described in the script explain **how to pickle a dictionary file**.

- 1) Start importing **pickle** package in Python
- 2) In this simple script, you will be pickling a simple dictionary that contains names and telephone numbers. You start declaring the dictionary in the Data section of the script.
- 3) Also, in the Data section, you also declare the name of the file you will be writing data to. I added the “dat” extension but you could have not added any extension.
- 4) In the Processing section, we define a function “**write\_data\_to\_file**”, which takes as parameters the file\_name (new “pickled”, binary file) and dic\_of\_data (file to be picked, in this case a dictionary).
- 5) To open the file for writing, simply use the **open()** function. The first argument should be the name of your file you will be writing to (new “pickled” file). The second argument is “wb”. The **w** means that you will be writing to the file, and **b** refers to binary mode. This means that the data will be written in the form of byte objects.
- 6) Once the file is opened for writing, you can use **pickle.dump()**, which takes two arguments: the object you want to pickle (dic\_of\_data) and the file to which the object has to be saved (outfile).
- 7) In the Presentation (Input/Output) section, you will call the **write\_data\_to\_file** function and define the arguments file\_name (new pickled file) and dic\_of\_data (original dictionary that will be pickled).
- 8) You will verify that a new file named Telephonebook.data should have appeared in the same directory as your Python script. The file should be in the same folder as your script when you used the correct, relative file path. (**Figure 4**).



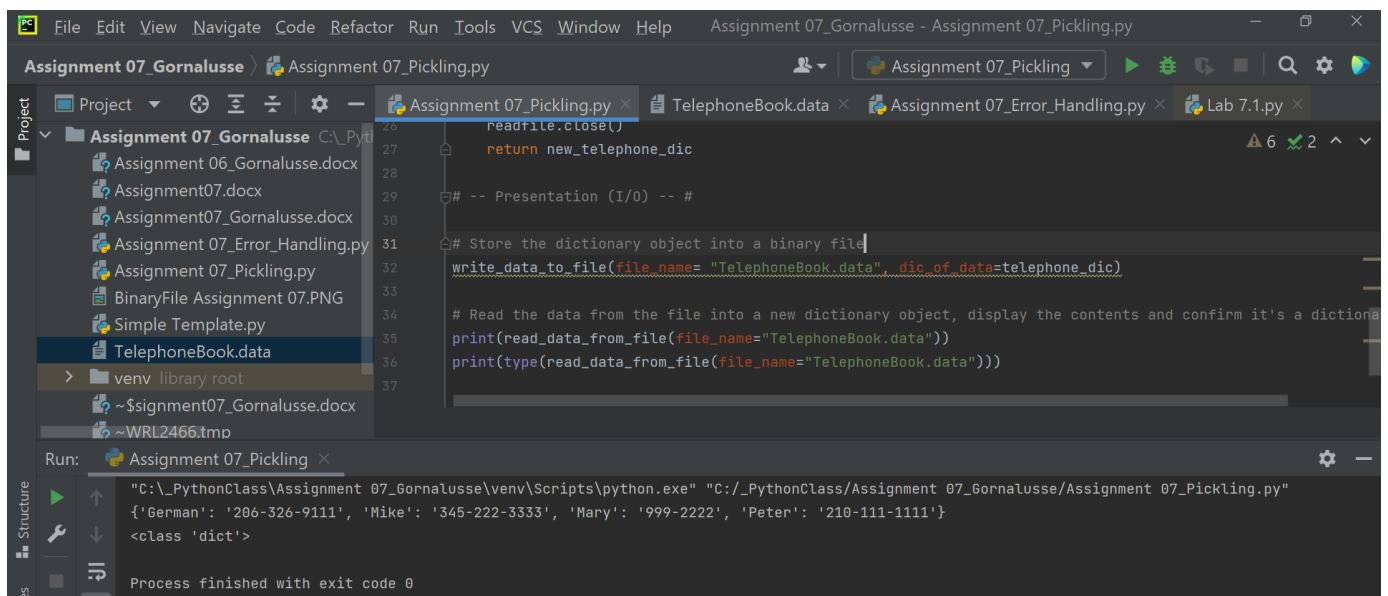


**Figure 4. Top panel: PyCharm window showing the binary file “TelephoneBook.data”. Notice the special characters showing how the code got “obscured”, but not totally encrypted. Also, notice that the location of this file. Bottom panel: binary file when opened with text editor Notepad.**

Below, I describe the steps to **how to unpickle a binary file into a new dictionary file.**

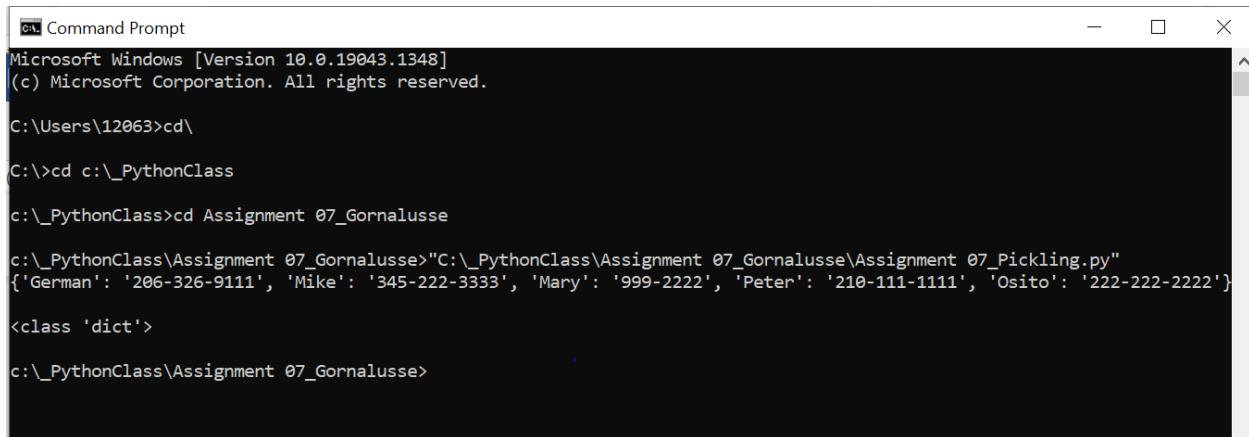
- 1) In the processing section, you will create the function “**read\_data\_from\_file**” to unpickle the binary file. The function takes as parameter the “pickled” binary file we created in the previous step.
- 2) We use the **open()** function again, but now with '**rb**' as second argument. The r stands for read mode and the b stands for binary mode, since you'll be reading a binary file. Assign this to a new object “readfile”.
- 3) Next, use **pickle.load()**, with readfile as argument, and assign it to a new dictionary “new\_telephone\_dic”. Again, you'll need to close the file at the end.
- 4) In the Presentation (I/O) section, you will read the data from the file into a new dictionary object. To do that, simply call the function “read\_data\_from\_file” and use as argument the binary file TelephoneBook.data you created in the previous section. Print this statement.
- 5) Confirm that the new object is a dictionary using the command **type()**.

**Figure 5**, below, verifies that this IO worked successfully.



**Figure 5. PyCharm console window verifies that the “unpickling” of the binary file into a new dictionary file occurred successfully. Please notice the new dictionary elements and how type() statement confirm that it is a dictionary object: <class ‘dic’>**

In the following **Figure 6** below, I illustrate how the assignment 07 worked properly once it is ran on OS command shell:



```
Command Prompt
Microsoft Windows [Version 10.0.19043.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\12063>cd\

C:\>cd c:\_PythonClass

c:\_PythonClass>cd Assignment 07_Gornalusse

c:\_PythonClass\Assignment 07_Gornalusse>"C:\_PythonClass\Assignment 07_Gornalusse\Assignment 07_Pickling.py"
{'German': '206-326-9111', 'Mike': '345-222-3333', 'Mary': '999-2222', 'Peter': '210-111-1111', 'Osito': '222-222-2222'}

<class 'dict'>

c:\_PythonClass\Assignment 07_Gornalusse>
```

*Figure 6. Assignment 07\_Pickling ran on OS command shell*

## Step 5. Structure Error Handling in Python

Before trying out the demo, please visit these websites:

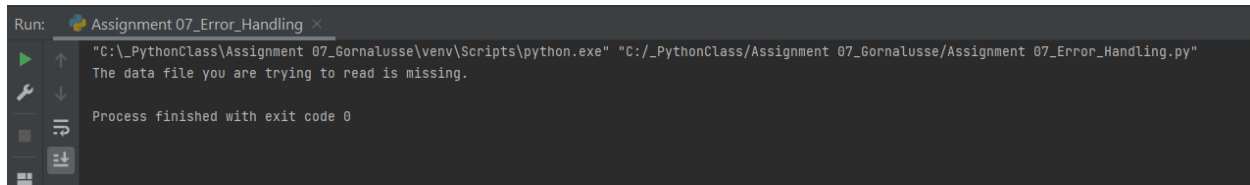
- <https://www.geeksforgeeks.org/python-exception-handling/> This website is a hub that contains many useful links in the bottom, wherein they explained separate exceptions like EOFError. The explanations are super clear.
- <https://www.techbeamers.com/use-try-except-python/> Ideal for beginners.
- <https://www.programiz.com/python-programming/exception-handling> I liked how the flow of error handling is explained in this website (what comes first, what is skipped, etc.)
- [https://www.w3schools.com/python/python\\_try\\_except.asp](https://www.w3schools.com/python/python_try_except.asp) I liked the simplicity of the explanations of the W3schools and the fact that you can try it out in any computer (even if you don't have Python actually installed).

When there are mistakes in your code, Python raises a runtime **exception**. In this demo “Assignment 07\_Error\_Handling”, we will practice.

First, you will learn what happens when your file is absent so Python cannot access it.

a) Write a **try**: statement. You will try to read the file “German’s file.text”, assign it to the object `file_data` and print it out. Because this file is missing, Python will internally raise an error of the type: `FileNotFound`.

b) Because this exception occurs, the code in the try’s suite terminates and then the code in the try’s except suite runs. The code is indented under the “except” clause and *only* executes if the “`FileNotFoundError`” exception is raised. The **Figure 7** below shows what happens when you run this code in Pycharm.



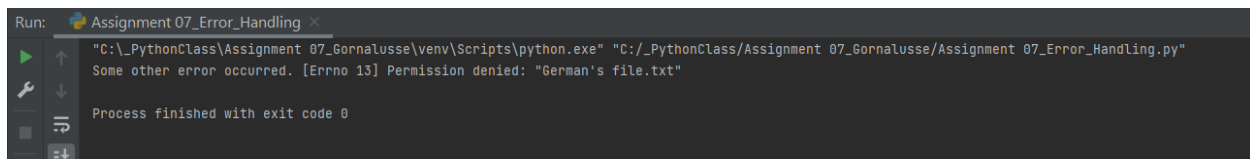
```
Run: Assignment 07_Error_Handling x
"C:\_PythonClass\Assignment 07_Gornalusse\venv\Scripts\python.exe" "C:/_PythonClass/Assignment 07_Gornalusse/Assignment 07_Error_Handling.py"
The data file you are trying to read is missing.

Process finished with exit code 0
```

**Figure 7.** The new version of the code produces a much friendlier message thanks to the “try” and “except” block that triggers the `FileNotFound` built-in class.

Now, you will create a **folder** in Assignment 07 with this name: “German’s file.txt” and you will run the code again.

In this case, the line #16 of your code “except Exception as err” will be triggered. Unlike the “non-specific” catch-all statement “except:”, this one arranges for the exception object to be assigned to the “err” variable and prints it out (line #17), as shown in **Figure 8** below:



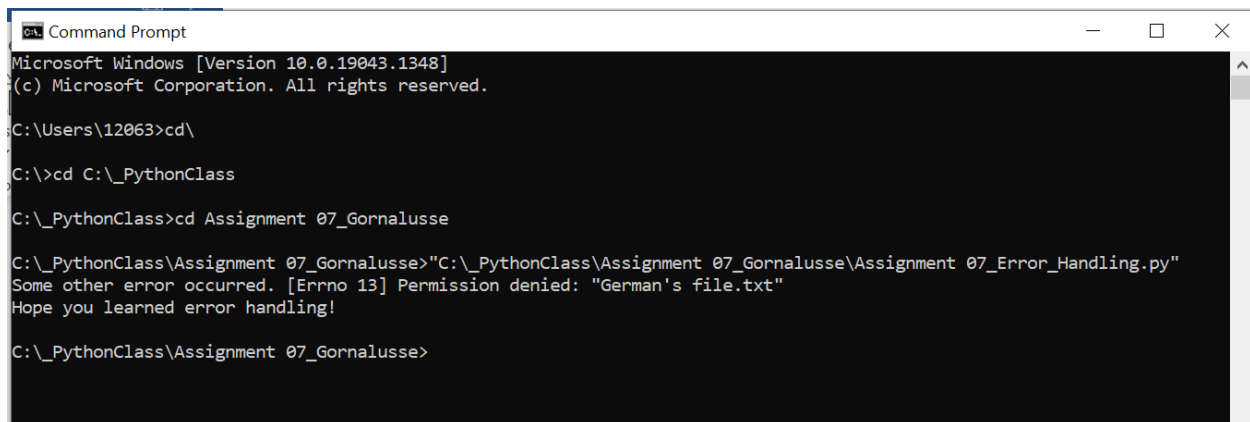
```
Run: Assignment 07_Error_Handling x
"C:\_PythonClass\Assignment 07_Gornalusse\venv\Scripts\python.exe" "C:/_PythonClass/Assignment 07_Gornalusse/Assignment 07_Error_Handling.py"
Some other error occurred. [Errno 13] Permission denied: "German's file.txt"

Process finished with exit code 0
```

**Figure 8.** The catch-all exception handler can assign the error to a variable that can be printed out

The last statement **finally** executes the command no matter how error handling happened in the try:except block of code.

In the following **Figure 9** below, I illustrate how the assignment 07\_Error Handling worked properly once it is ran on OS command shell:



```
Command Prompt
Microsoft Windows [Version 10.0.19043.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\12063>cd\

C:\>cd C:\_PythonClass

C:\_PythonClass>cd Assignment 07_Gornalusse

C:\_PythonClass\Assignment 07_Gornalusse>"C:\_PythonClass\Assignment 07_Gornalusse\Assignment 07_Error_Handling.py"
Some other error occurred. [Errno 13] Permission denied: "German's file.txt"
Hope you learned error handling!

C:\_PythonClass\Assignment 07_Gornalusse>
```

**Figure 9.** Assignment\_07 Error Handling ran under the command line OS shell

## Summary

In this paper, you were introduced to two useful Python features: pickling and error handling using try:except blocks. Pickling” is the process whereby a Python object (list, dic, etc.) is converted into a byte stream (0s and 1s), and “unpickling” is the inverse operation. The idea is that this binary-coded

character stream contains all the information necessary to reconstruct the object in another python script. The main advantage of pickling/unpickling is it allows to transfer files of reduced size from a server to another. In the second part of this paper, I illustrated how the try:except block can be used to spot errors in the script. You will also learn how to assign a non-specific exception as an error and how to use it.