**Samir Abad**

**May 29, 2022**

**IT FDN 110A Sp. 22**

**Assignment 7**

**Github url-** [mailsamir/ITFnd100-Mod07 (github.com)](https://github.com/mailsamir/ITFnd100-Mod07)

**Github webpage -** [Exception handling and Pickling in Python | ITFnd100-Mod07 (mailsamir.github.io)](https://mailsamir.github.io/ITFnd100-Mod07/)

**Exception handling in Python**

**Resources:**

[**https://www.tutorialsteacher.com/python/exception-handling-in-python**](https://www.tutorialsteacher.com/python/exception-handling-in-python)

This website shows how to apply Exception handling in python. It shows some examples to catch basic error types such as division by zero. It then extends the concept to Multiple except blocks.

[**https://www.w3schools.com/python/python\_try\_except.asp**](https://www.w3schools.com/python/python_try_except.asp)

This website shows simple examples on try, except, else and finally blocks. I learnt that you can define as many exception blocks as you want, e.g. if you want to execute a special block of code for a special kind of error. You can use the else keyword to define a block of code to be executed if no errors were raised. The finally block, if specified, will be executed regardless if the try block raises an error or not.

<https://www.pythontutorial.net/python-oop/python-raise-exception/>

This website shows how to raise an exception. As a Python developer you can choose to throw an exception if a condition occurs. To throw (or raise) an exception, use the raise keyword.

**Pickling in Python**

.

**Resources:**

**https://www.datacamp.com/tutorial/pickle-python-tutorial**

I found this website useful because it did a great job of explaining fundamentals associated with pickling in python. It started with description of pickling and how it is different from compression. It briefly explained how is pickling useful and where and when we should use pickling. It provides a brief summary of various object types that can be pickled. It also provided some basic examples on how to pickle a dictionary using the dump () method. The unpickling files was demonstrated using the load () method.

[**https://realpython.com/python-pickle-module/**](https://realpython.com/python-pickle-module/)

I found this website especially useful since it explained the difference between pickle, json and marshal. It also lists the protocols that Python pickle module can use. It also describes the difference between the 2 pickling methods- dump () and dumps ().The only difference between dump () and dumps () is that the first creates a file containing the serialization result, whereas the second returns a string. It also briefly covers risks associated with pickling.

[**https://towardsdatascience.com/do-not-use-python-pickle-unless-you-know-all-these-facts-d9e8695b7d43**](https://towardsdatascience.com/do-not-use-python-pickle-unless-you-know-all-these-facts-d9e8695b7d43)

This website goes into more detail on what are the pros and cons of pickling. It also describes objects that cannot be pickled.

**Applying knowledge:**

**Pickling**-The function that writes to a binary file is shown below. The parameters defined for this function are a) name of file and a list that has the items defined. The dump() method writes the list of items in to the file in the binary format.

import pickle # This imports code from another code file!  
  
# Data -------------------------------------------- #  
# Declare variables and constants  
fName = 'newfile.dat'  
lstItemsPrice = []  
  
def write\_to\_file(filename, list\_of\_items):  
 file=open(filename,"wb")  
 pickle.dump(list\_of\_items,file)  
 file.close()

**Exception handling**- Two examples are shown below. The first one is to ask the user to input the name of an item. If the name is not defined by alphabets than an exception error is raised. The print command shows various options of printing the error in addition to a customized message.

The second example gets the price for the user. If the input is not an integer then an Exception error is raised and a customized message is printed asking the user to only provide integer values.

try:  
 item = str(input("Enter name of item:"))  
 if not item.isalpha():  
 raise Exception('Do not use numbers for the item name. No special characters(Space, #, $, etc.) are allowed as input!')  
except Exception as e:  
 print('There was a non-specific error!')  
 print("Built-In Python error info: ")  
 print(e, e.\_\_doc\_\_, type(e), sep='\n')  
 exit()  
  
else:  
 try:  
 price=input("Enter item price in $:")  
 if not price.isnumeric():  
 raise Exception('Use integer values to input the price. No special characters(Space, #, $, etc.) are allowed as input!')  
 except Exception as e:  
 print('There was a non-specific error!')  
 print("Built-In Python error info: ")  
 print(e, e.\_\_doc\_\_, type(e), sep='\n')  
 exit()

Graphical user interface, application

Description automatically generated

Figure1- Binary file created using pickling

Text

Description automatically generated

Figure2- Code successfully ran in pycharm

**Summary:**

The various online resources helped firm the understanding on how to apply Exception handling and pickling. The simple python script was used to apply and demonstrate the knowledge. Overall, this assignment was very helpful in introducing basic python scripting to a novice programmer.