Hitakshi Verma

August 21, 2020

Foundations of Programming: Python

Assignment 07

<https://github.com/hitakshi01/-IntroToProg-Python-Mod07/>

**Create a script using exception error handling and pickling**

**Introduction:**

Under the guidance of my professor, Randal Root, I have learnt how to create scripts using custom functions, files and structured error handling. As a part of my assignment, I am going to share my knowledge on structured error handling and pickling; sharing some resource links with you folks for your better understanding about these topics.

Also, I am going to create a script that demonstrates python’s structural error handling and pickling concept.

I believe you all are ready and curious to know these concepts. So, let’s get started.

**Pickling and Unpickling in Python:**

Python’s pickle module is used for serializing and de-serializing python object structures. The process to converts any kind of python objects (list, dict, etc.) into byte streams (0s and 1s) is called *pickling or serialization or flattening or marshalling*. We can convert the byte stream (generated through pickling) back into python objects by a process called as *unpickling*.

In real world scenarios, “pickling” is needed when we have to transfer data from one server/system to another and then store it in a file or database.

It is advisable not to “unpickle” data received from an untrusted source as they may pose security threat. Pickle module has no way of knowing or raising alarm while pickling malicious data.

To pickle the data, first we have to import the “pickle” module in our python script.

Pickle module has two main functions: *dump* and *load*. The first one is ***dump***, which dumps an object to a file object and the second one is ***load***, which loads an object from a file object.

To understand the basics of picking and unpickling in more detail. Please navigate to the following links provided below as these links not only gives you *basic understanding* but also helps you to get familiar with other useful *functions* and *exception handling* presented in pickle module.

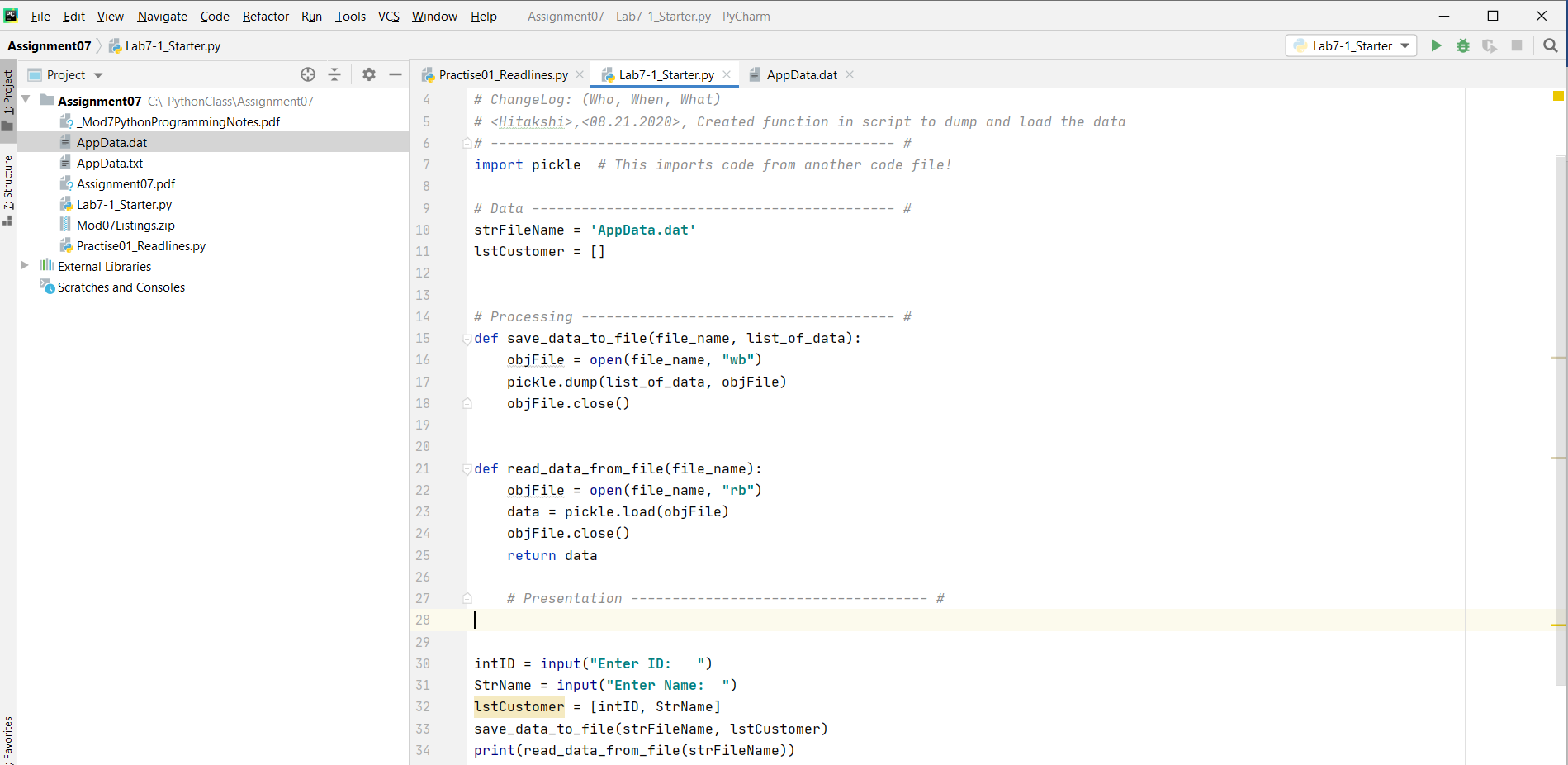
* <https://www.tutorialspoint.com/python-pickling>
* <https://thepythonguru.com/pickling-objects-in-python/>
* <https://docs.python.org/3/library/pickle.html>

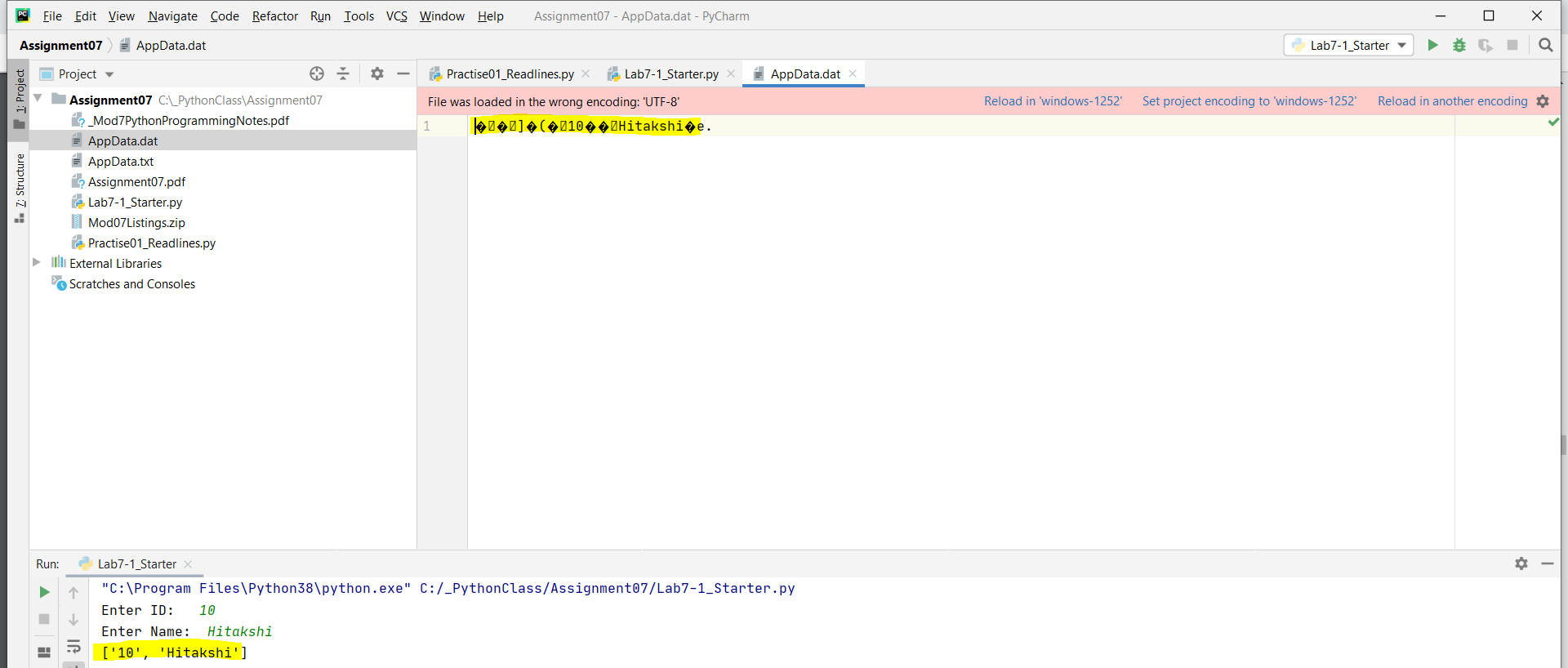
I am also sharing a link with you if you are interested from “*software* *security* perspective.

* <https://www.synopsys.com/blogs/software-security/python-pickling/>

**Working on LAB 7-1:**

In this lab, I am going to create a function that saves data to a binary file using dump() function and loads object from file using load() function. For this, I am using my professor, Randal Root’s starter file.





*Figure1 to 3 demonstrates the data saved into binary file using dump and loaded from file using load function*

**Structured Error Handling:**

Generally, there are two types of errors in programming- syntax errors and exceptions. Here we are more interested in knowing exceptions and how we handle exceptions.

When a statement or expression is syntactically correct but an error is caused during its execution, then it is called as “exception”. When that error occurs, Python details what type of exception error was encountered. Exceptions are convenient in many ways for handling errors and special condition in a program. When we think that we have a code which can produce an error then we can use exception handling. We can raise an exception in your own program by using the “raise” exception statement.

Raising an exception breaks current code execution and returns the exception back until it is handled. We can trap errors using try- catch block when we think there is some condition that might cause an error and puts that condition in exception block.

"Exception" is a built-in python class used to hold information about an error. Python automatically creates an Exception object when an error occurs. The Exception object automatically fills with information about the error that caused the exception.

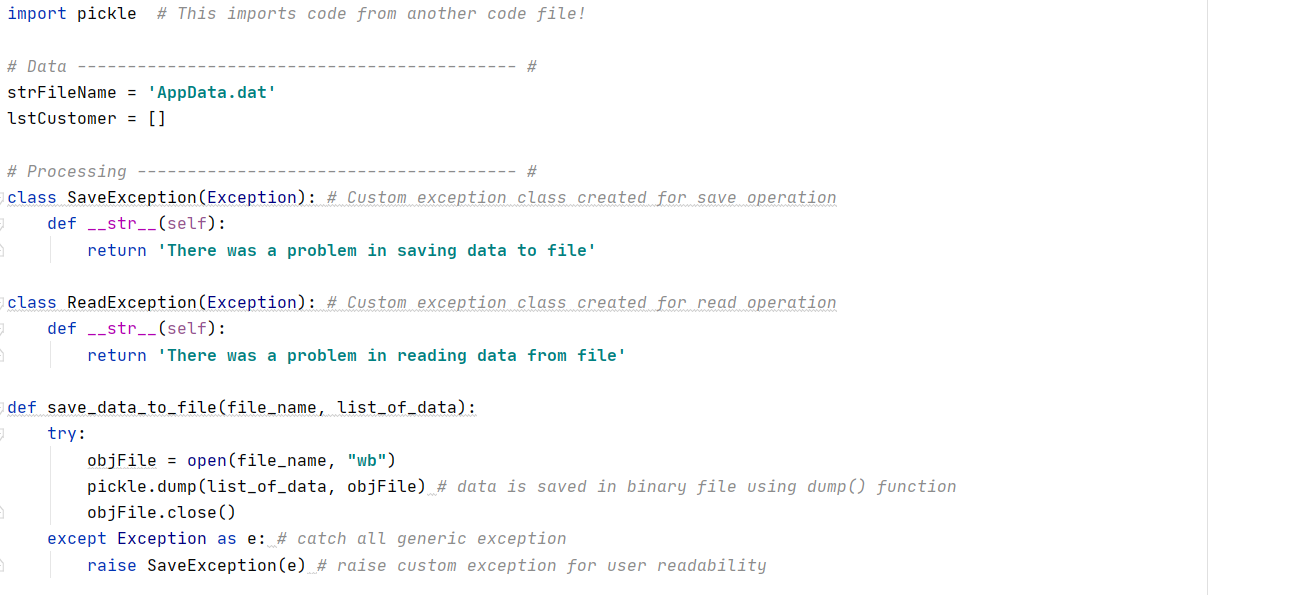
For more details about structural handling, folks, you can try these links which helps you to learn how to do exception handling in more details.

* <https://realpython.com/python-exceptions/>
* <https://www.programiz.com/python-programming/exception-handling>

**Assignment 7 - Creating a python script that demonstrates python’s pickling and exception error handling**

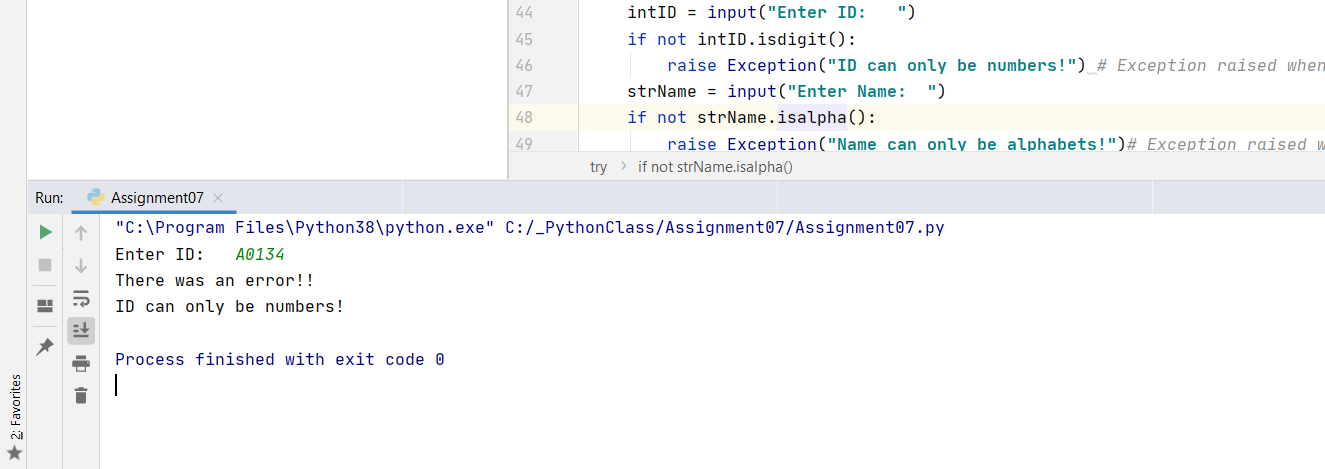
Since we did not have a starter file or any specific problem statement; for this assignment, I have re-used the code for lab 7-1 and added error handling using try-except.

I created 2 custom exception classes *SaveException* and *ReadException* and defined a new function \_\_str\_\_ which returns a print statement with a custom error message.



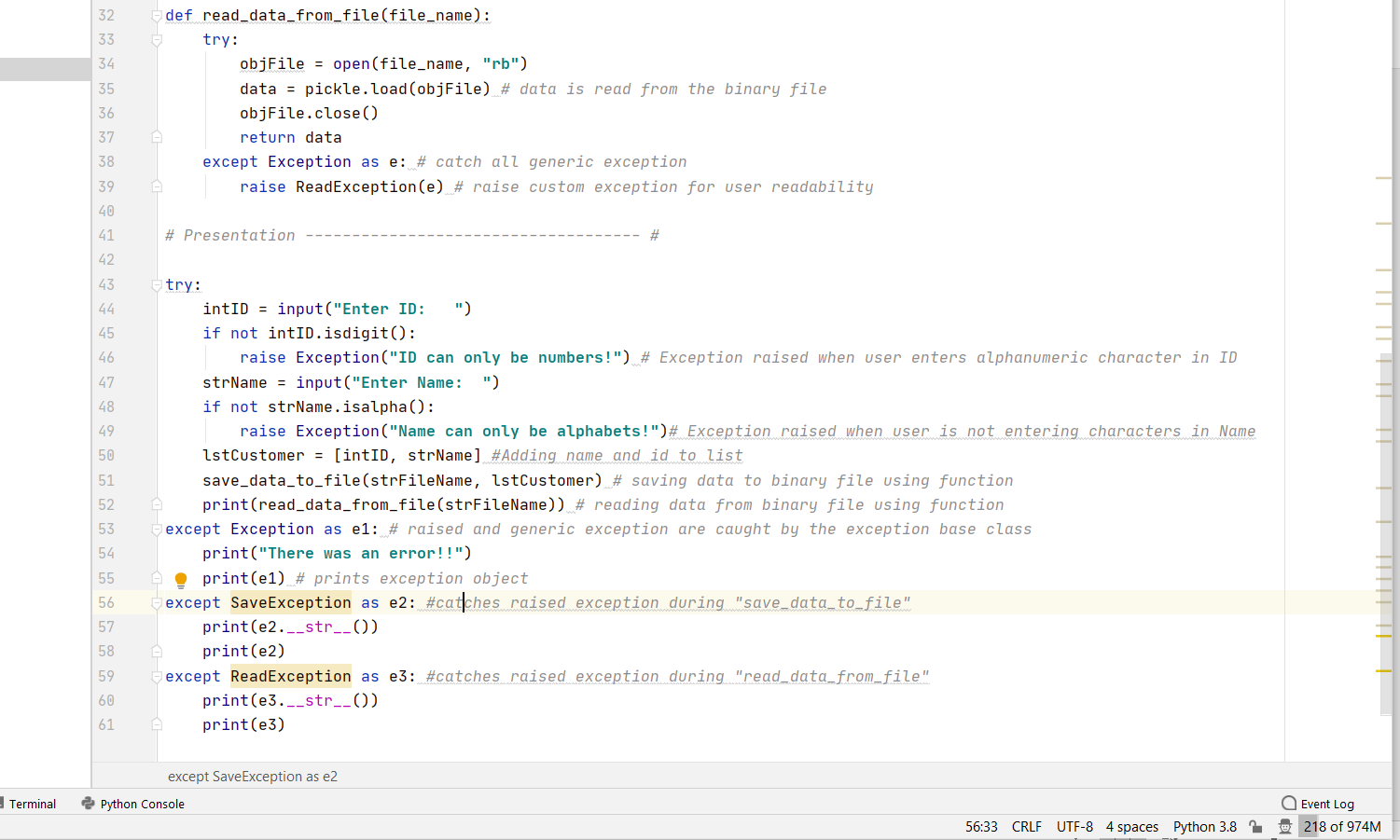
*Figure 1: Define custom exceptions to be used in individual functions*

In the main body of the script, the user inputs the “ID” and “Name”. I assumed “ID” can only be numbers/ digits and “name” can only be alphabets. Hence, I used isdigit() and isalpha() functions respectively. I used an if statement to compare the input with the above functions and raise an exception.



*Figure 2: Throw error if alphanumeric ID entered*

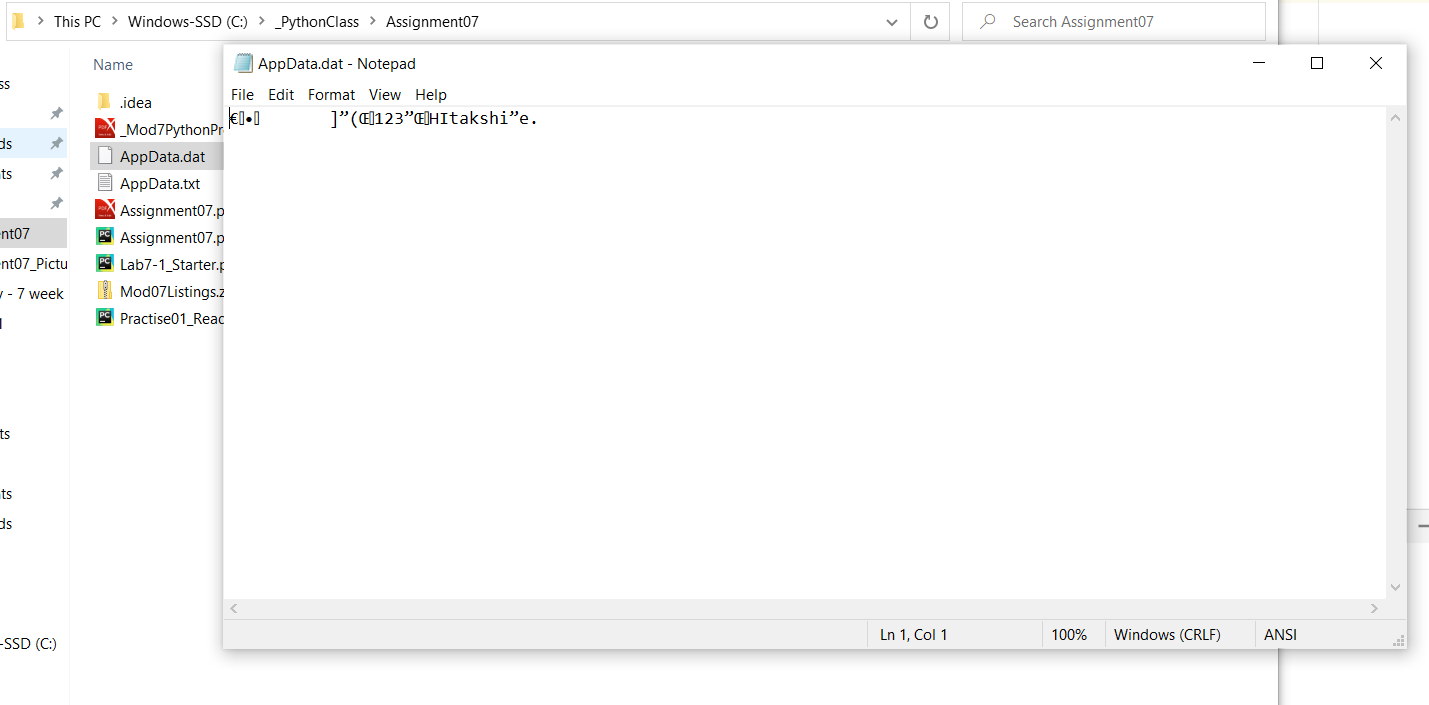
I added the “Name” and “ID” into a list and used the function save\_data\_to\_file to save the list into a binary file. Within the function, I opened the file using the *open* function and used dump function provided by pickle to store the data. I have put the code within try- except code to catch all the exceptions and throw a custom exception SaveException as defined above.



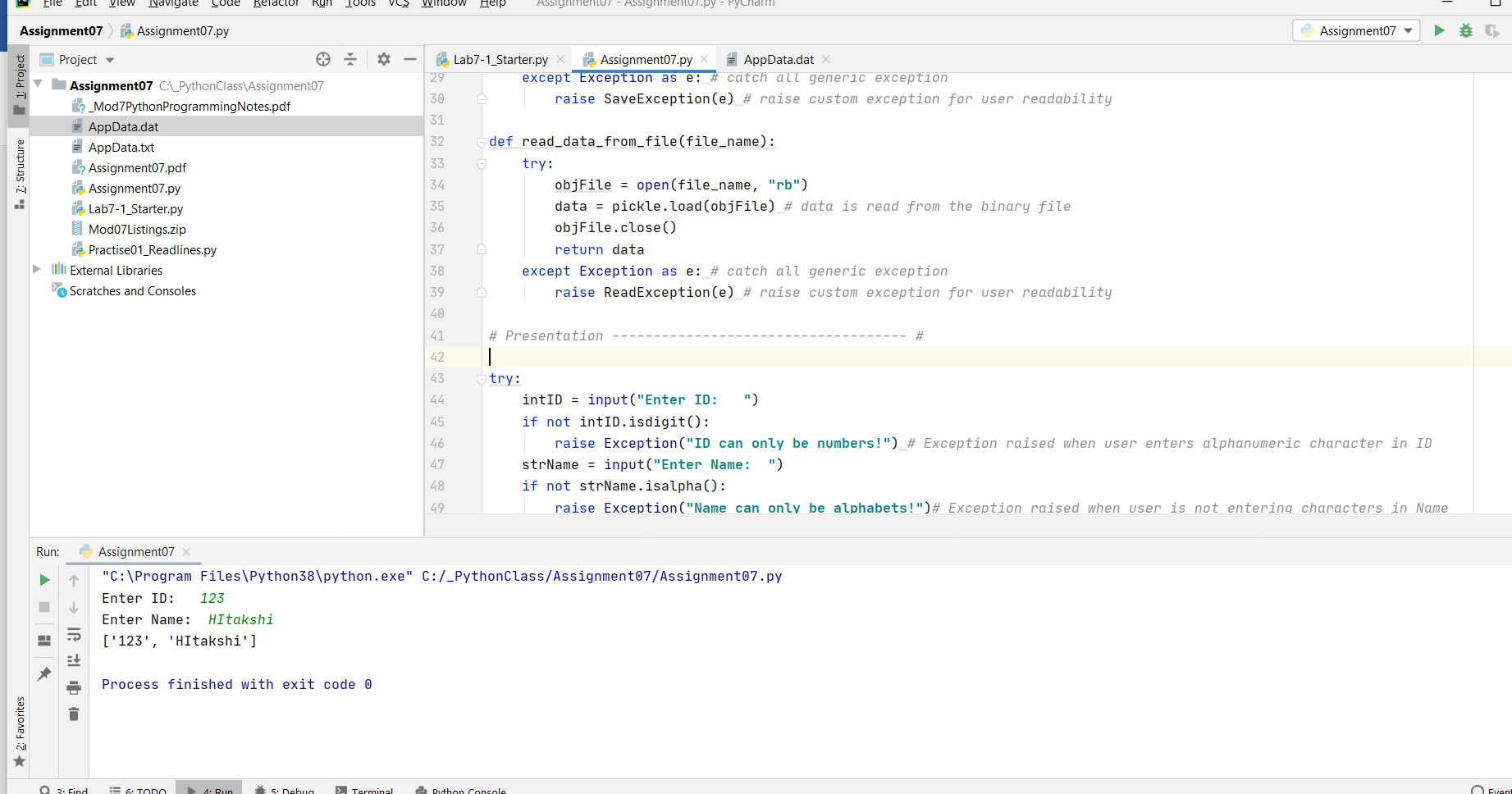
*Figure 3: Main body of the script*

In the next step, I created a function read\_data\_from\_file which opens the file, and used the load function provided by pickle and return the data. Similar to save, I wrapped the code in try -except block to catch any exceptions and then raise a custom error – ReadException.

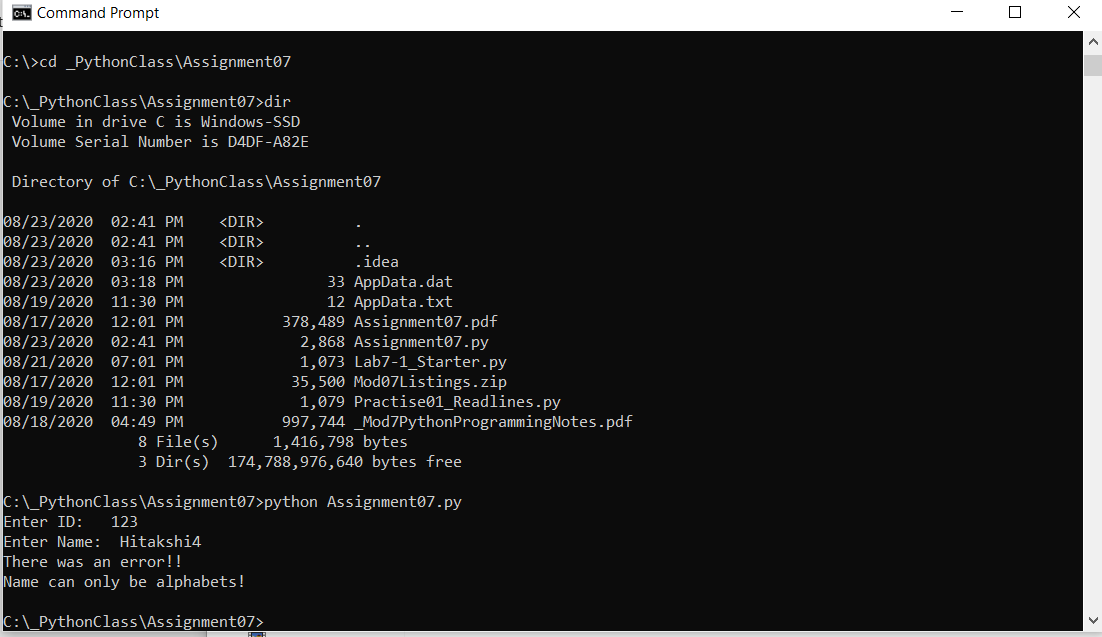
Within the main body, I catch the exceptions thrown from the individual functions and display a user friendly message. I used 3 except statements Exception, SaveException and ReadException and close the script.



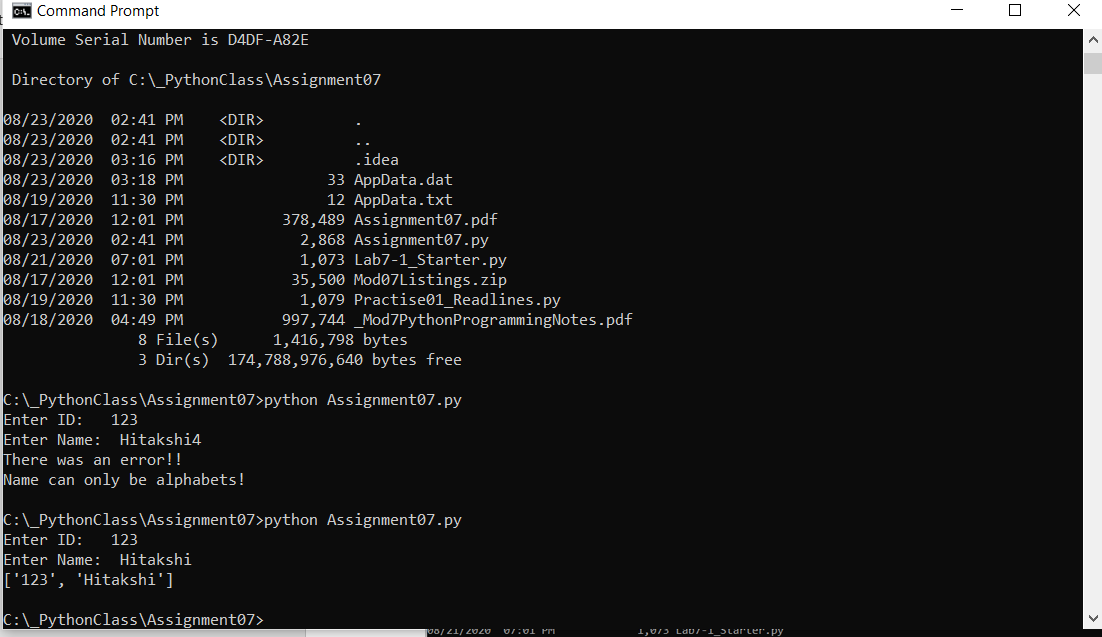
*Figure 4: AppData.dat file which is a binary file*



*Figure 5: Running the script from PyCharm*



*Figure 6: Running script from command prompt with error*



*Figure 7: Running the script from PyCharm*

**Summary:**

After knowing how to do professional scripting, here i have learnt how to handle errors that are not syntactic ones but exceptions that usually comes when the program runs. I have created scripts and performed lab work where I handled custom and python generated exceptions using exception class and custom exception class. I think it's important for a programmer to write each piece of code in try- except block for efficient scripting and error handling. I have also created a GitHub web page where i have posted all my assignment work, found this way quite easy and quick.