Name: Mitsuyo Kawano
Date: May 31, 2023
Course: Foundations of Programming Python
Assignment 07
GitHub URL: https://github.com/mitsuyojp/IntroToProg-Python-Mod07
GitHub Webpage: https://mitsuyojp.github.io/IntroToProg-Python-Mod07/

The Steps in Performing the Assignment Script

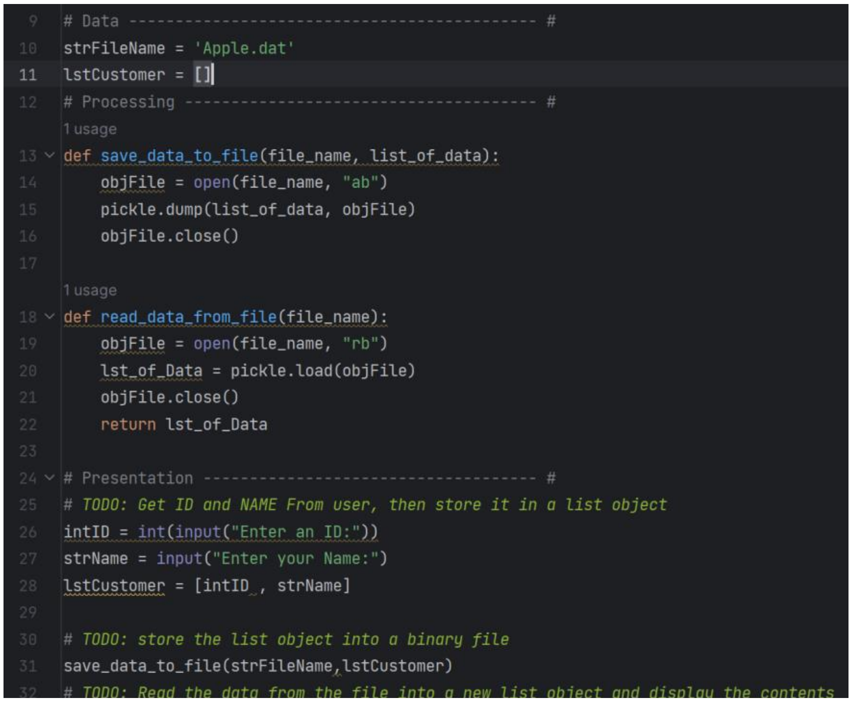
Introduction

In Module 7, we learned about working with Text Files, Binary Files, Error Handling, and Creating Advanced GitHub pages with Markdown.
Module 7 assignment is creating a script using Exception Handling and Pickling.
I'd like to explain the steps I took in performing this assignment.

What are steps?

1. Start with simple script using Lab7-1

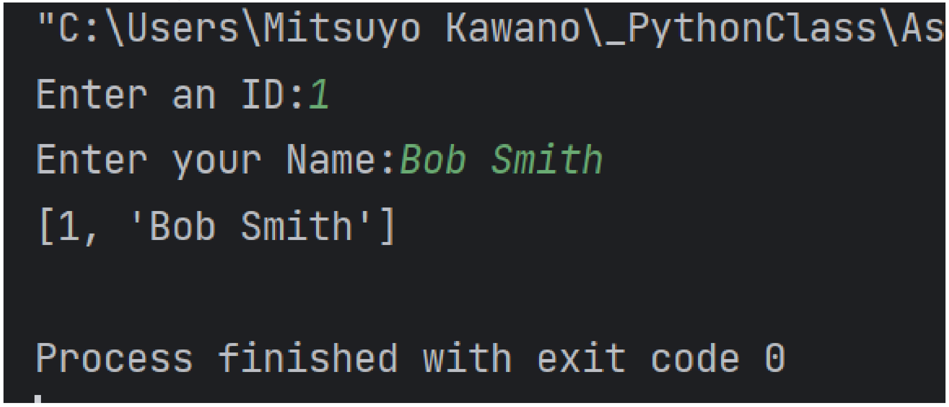
I have to include Pickling and Exception Handling, but first, I started with a simple Lab7-1 file to use Pickling.



```
# Data ------------------------------------------- #
strFileName = 'Apple.dat'
lstCustomer = []
# Processing ------------------------------------- #
1 usage
def save_data_to_file(file_name, list_of_data):
    objFile = open(file_name, "ab")
    pickle.dump(list_of_data, objFile)
    objFile.close()

1 usage
def read_data_from_file(file_name):
    objFile = open(file_name, "rb")
    lst_of_Data = pickle.load(objFile)
    objFile.close()
    return lst_of_Data

# Presentation ----------------------------------- #
# TODO: Get ID and NAME From user, then store it in a list object
intID = int(input("Enter an ID:"))
strName = input("Enter your Name:")
lstCustomer = [intID , strName]

# TODO: store the list object into a binary file
save_data_to_file(strFileName,lstCustomer)
# TODO: Read the data from the file into a new list object and display the contents
```

Figure 1 A Screenshot of Lab7-1 Script (PyCharm)

It is working fine!



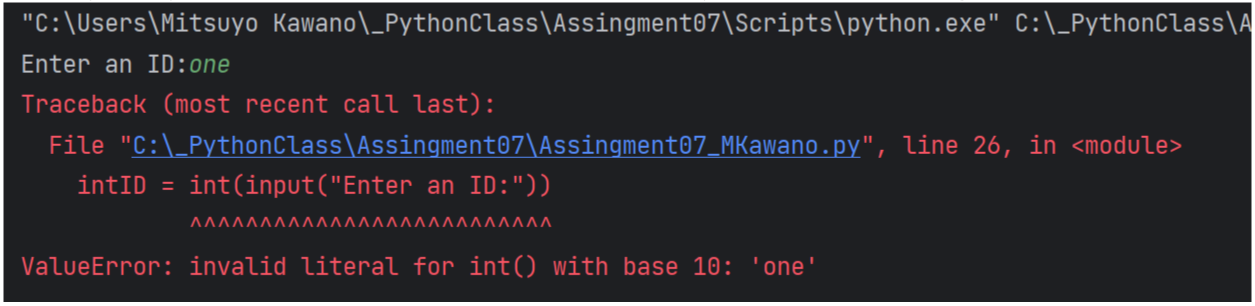
```
"C:\Users\Mitsuyo Kawano\_PythonClass\As
Enter an ID:1
Enter your Name:Bob Smith
[1, 'Bob Smith']

Process finished with exit code 0
```

Figure 2 A Screenshot of Lab7-1 Test Result (PyCharm)

2. Think about possible errors and modify the script.

Now I have to add Error Handling. What kind of errors can be created by user? One possibility is user input incorrect ID number, for example, not number. (int)

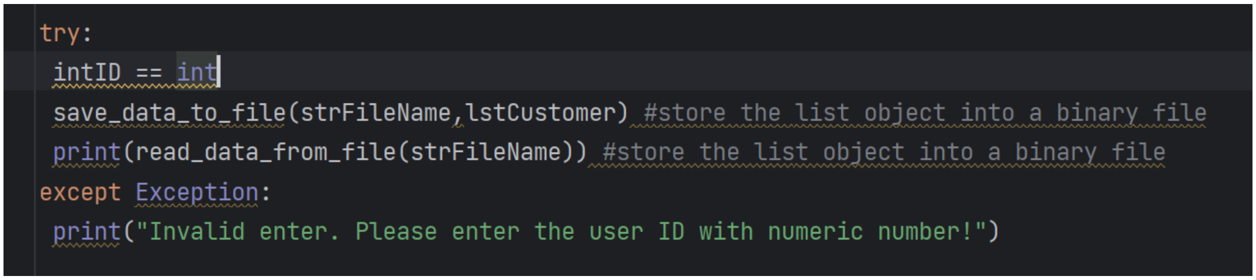
First, I just input “one” for the ID number, and this is Python’s error message.



```
"C:\Users\Mitsuyo Kawano\_PythonClass\Assingment07\Scripts\python.exe" C:\_PythonClass\A
Enter an ID:one
Traceback (most recent call last):
  File "C:\_PythonClass\Assingment07\Assingment07_MKawano.py", line 26, in <module>
    intID = int(input("Enter an ID:"))
            ^^^^^^^^^^^^^^^^^^^^^^^^^^
ValueError: invalid literal for int() with base 10: 'one'
```

Figure 3 A Screenshot of error message (PyCharm)

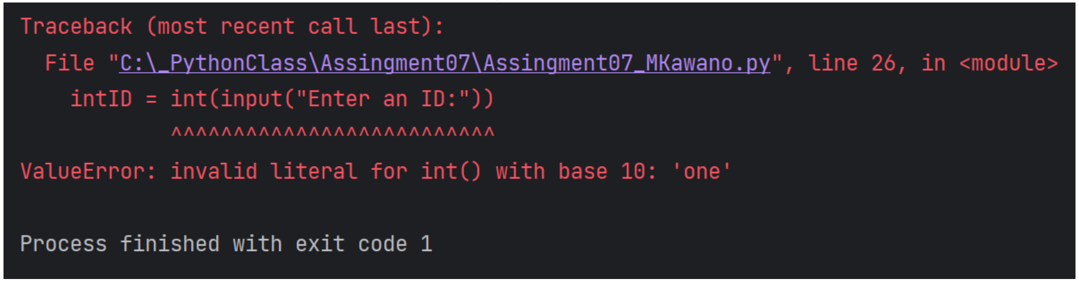
I added try and except as below into my script. If the user inputs numeric ID number, the entire script will be executed, otherwise print the error message to the user.



```
try:
 intID == int
 save_data_to_file(strFileName,lstCustomer) #store the list object into a binary file
 print(read_data_from_file(strFileName)) #store the list object into a binary file
except Exception:
 print("Invalid enter. Please enter the user ID with numeric number!")
```

Figure 4 A Screenshot of try except script (PyCharm)

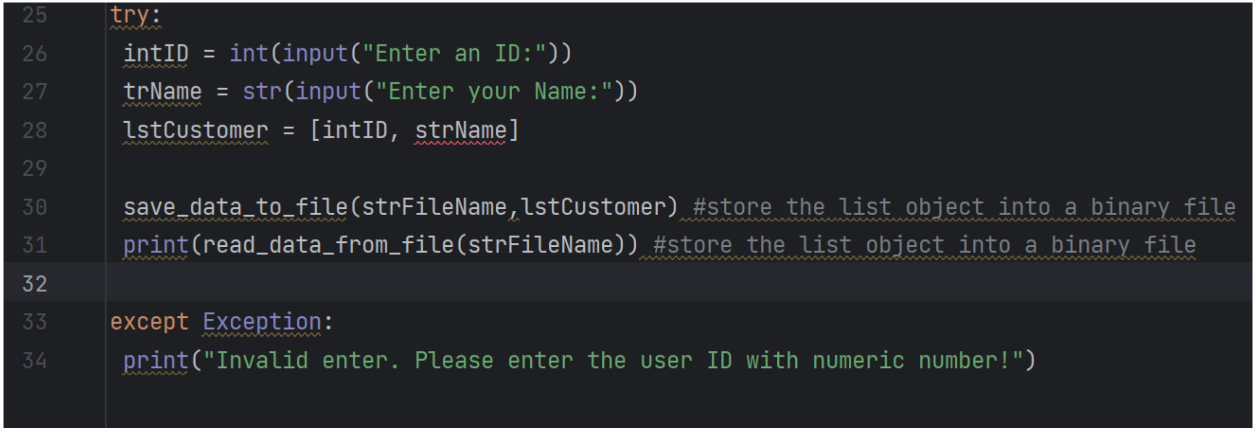
Here is the test result. It’s not working well.



```
Traceback (most recent call last):
  File "C:\_PythonClass\Assingment07\Assingment07_MKawano.py", line 26, in <module>
    intID = int(input("Enter an ID:"))
            ^^^^^^^^^^^^^^^^^^^^^^^^^^
ValueError: invalid literal for int() with base 10: 'one'

Process finished with exit code 1
```

Figure 5 A Screenshot of error message (PyCharm)

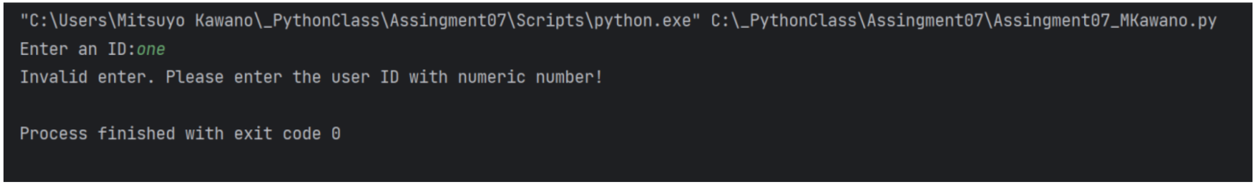
I realized I have to put all the script I want to try to run inside of the try block. And changed the script as below.



```
try:
    intID = int(input("Enter an ID:"))
    trName = str(input("Enter your Name:"))
    lstCustomer = [intID, strName]

    save_data_to_file(strFileName,lstCustomer) #store the list object into a binary file
    print(read_data_from_file(strFileName)) #store the list object into a binary file

except Exception:
    print("Invalid enter. Please enter the user ID with numeric number!")
```

Figure 6 A Screenshot of modified try and except script (PyCharm)

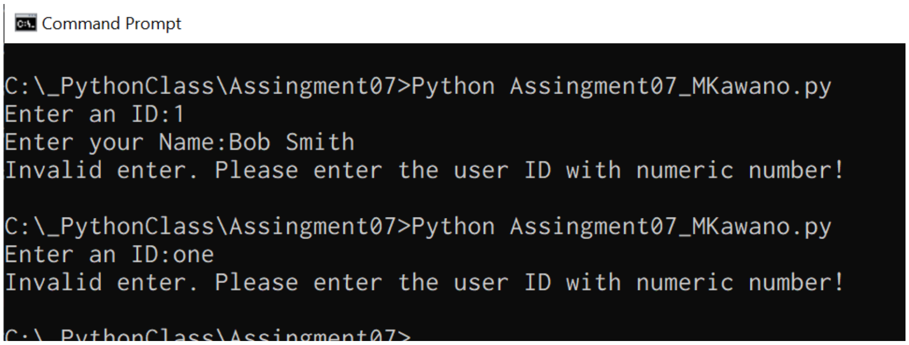
It ran successfully! Showing the user-friendly error message./



```
"C:\Users\Mitsuyo Kawano\_PythonClass\Assingment07\Scripts\python.exe" C:\_PythonClass\Assingment07\Assingment07_MKawano.py
Enter an ID:one
Invalid enter. Please enter the user ID with numeric number!

Process finished with exit code 0
```

Figure7 A Screenshot of test result (PyCharm)

It ran successfully in Command Line!



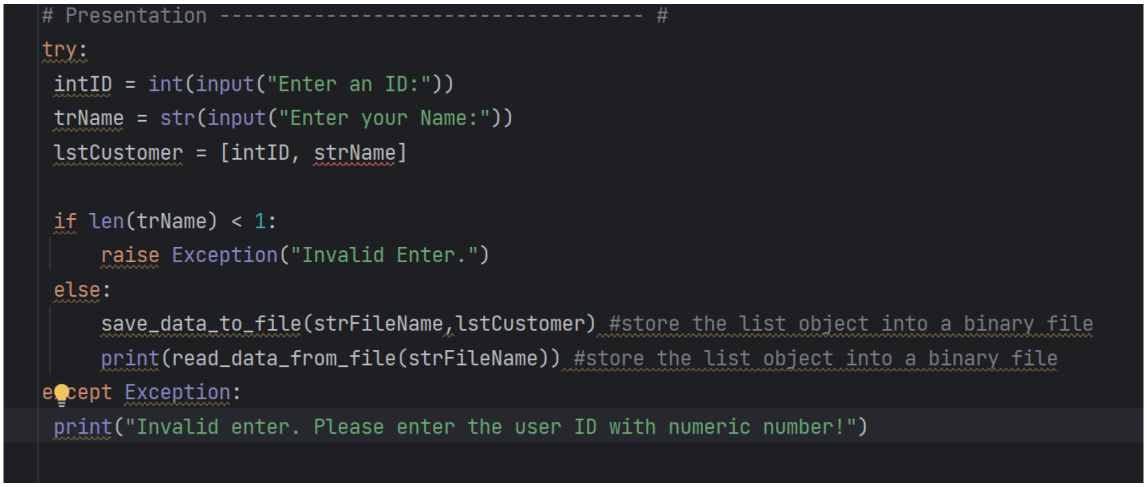
```
C:\_PythonClass\Assingment07>Python Assingment07_MKawano.py
Enter an ID:1
Enter your Name:Bob Smith
Invalid enter. Please enter the user ID with numeric number!

C:\_PythonClass\Assingment07>Python Assingment07_MKawano.py
Enter an ID:one
Invalid enter. Please enter the user ID with numeric number!

C:\_PythonClass\Assingment07>
```

Figure 8 A Screenshot of the test result (Command)

3. Think of another possible error and modify the script.

Another possible error I can think of is the user input the name incorrectly such us only one character. So, I have to add a script to check if the input name data is more than 2 characters.

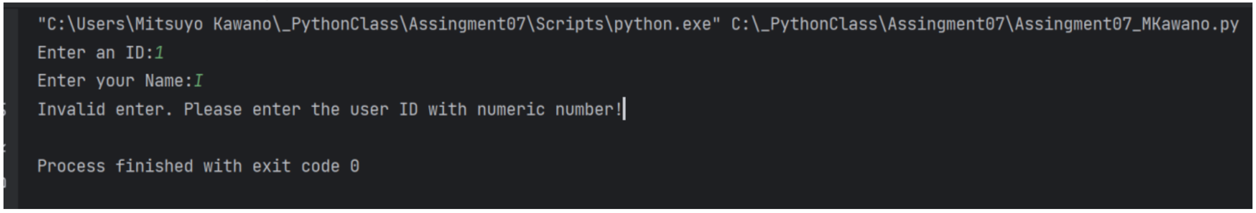
First modification is like below.



```
# Presentation ------------------------------------ #
try:
 intID = int(input("Enter an ID:"))
 trName = str(input("Enter your Name:"))
 lstCustomer = [intID, strName]

 if len(trName) < 1:
     raise Exception("Invalid Enter.")
 else:
     save_data_to_file(strFileName,lstCustomer) #store the list object into a binary file
     print(read_data_from_file(strFileName)) #store the list object into a binary file
except Exception:
 print("Invalid enter. Please enter the user ID with numeric number!")
```

Figure 9 A Screenshot of modified sript (PyCharm)

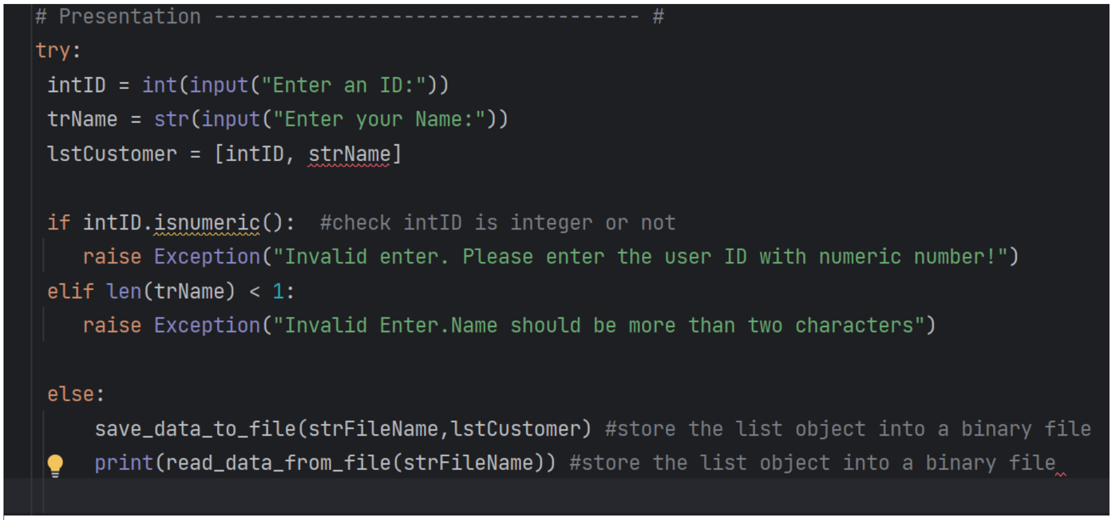
But the error message is not the one I wanted to show to the user.



```
"C:\Users\Mitsuyo Kawano\_PythonClass\Assingment07\Scripts\python.exe" C:\_PythonClass\Assingment07\Assingment07_MKawano.py
Enter an ID:1
Enter your Name:I
Invalid enter. Please enter the user ID with numeric number!

Process finished with exit code 0
```

Figure 10 A Screenshot of the error message (PyCharm)

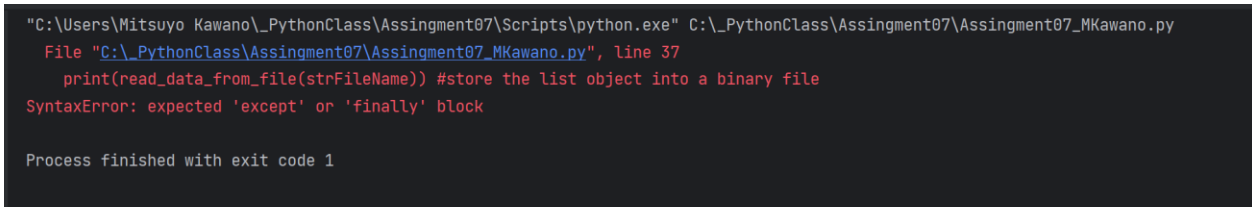
I revised it like this.



```
# Presentation ------------------------------------ #
try:
 intID = int(input("Enter an ID:"))
 trName = str(input("Enter your Name:"))
 lstCustomer = [intID, strName]

 if intID.isnumeric():  #check intID is integer or not
    raise Exception("Invalid enter. Please enter the user ID with numeric number!")
 elif len(trName) < 1:
    raise Exception("Invalid Enter.Name should be more than two characters")

 else:
     save_data_to_file(strFileName,lstCustomer) #store the list object into a binary file
     print(read_data_from_file(strFileName)) #store the list object into a binary file
```

Figure 11 A Screenshot of modified script (PyCharm)

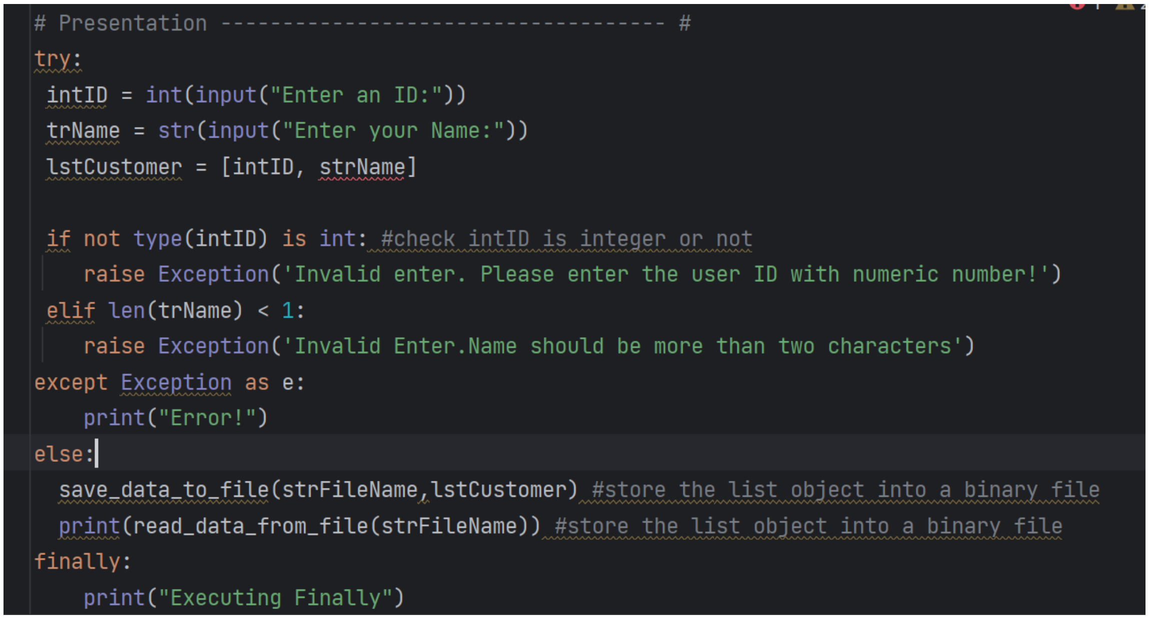
Now showing this error. expected 'except' or 'finally' block???



```
"C:\Users\Mitsuyo Kawano\_PythonClass\Assingment07\Scripts\python.exe" C:\_PythonClass\Assingment07\Assingment07_MKawano.py
  File "C:\_PythonClass\Assingment07\Assingment07_MKawano.py", line 37
    print(read_data_from_file(strFileName)) #store the list object into a binary file
SyntaxError: expected 'except' or 'finally' block

Process finished with exit code 1
```

Figure 12 A Screenshot of the error message (PyCharm)

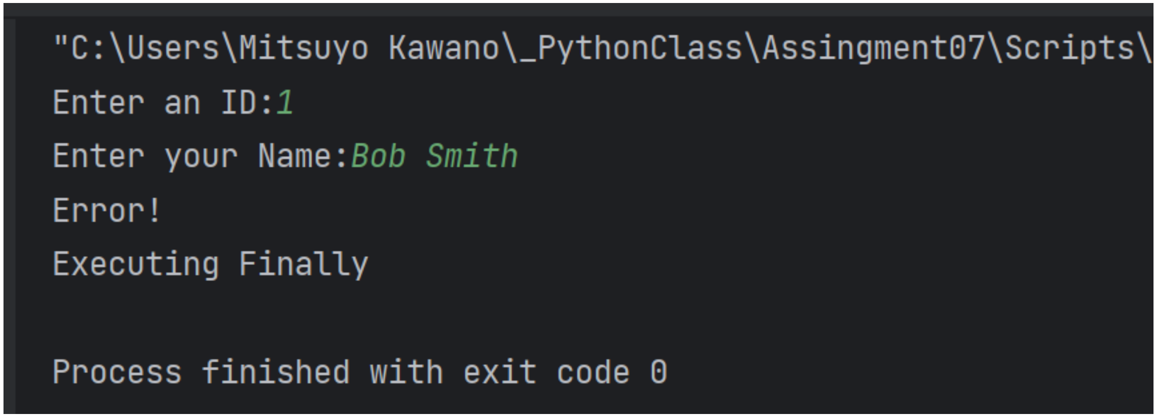
I revised it like this.



```
# Presentation ------------------------------------ #
try:
 intID = int(input("Enter an ID:"))
 trName = str(input("Enter your Name:"))
 lstCustomer = [intID, strName]

 if not type(intID) is int: #check intID is integer or not
    raise Exception('Invalid enter. Please enter the user ID with numeric number!')
 elif len(trName) < 1:
    raise Exception('Invalid Enter.Name should be more than two characters')
except Exception as e:
    print("Error!")
else:
  save_data_to_file(strFileName,lstCustomer) #store the list object into a binary file
  print(read_data_from_file(strFileName)) #store the list object into a binary file
finally:
    print("Executing Finally")
```

Figure 13 A Screenshot of the revised script (PyCharm)

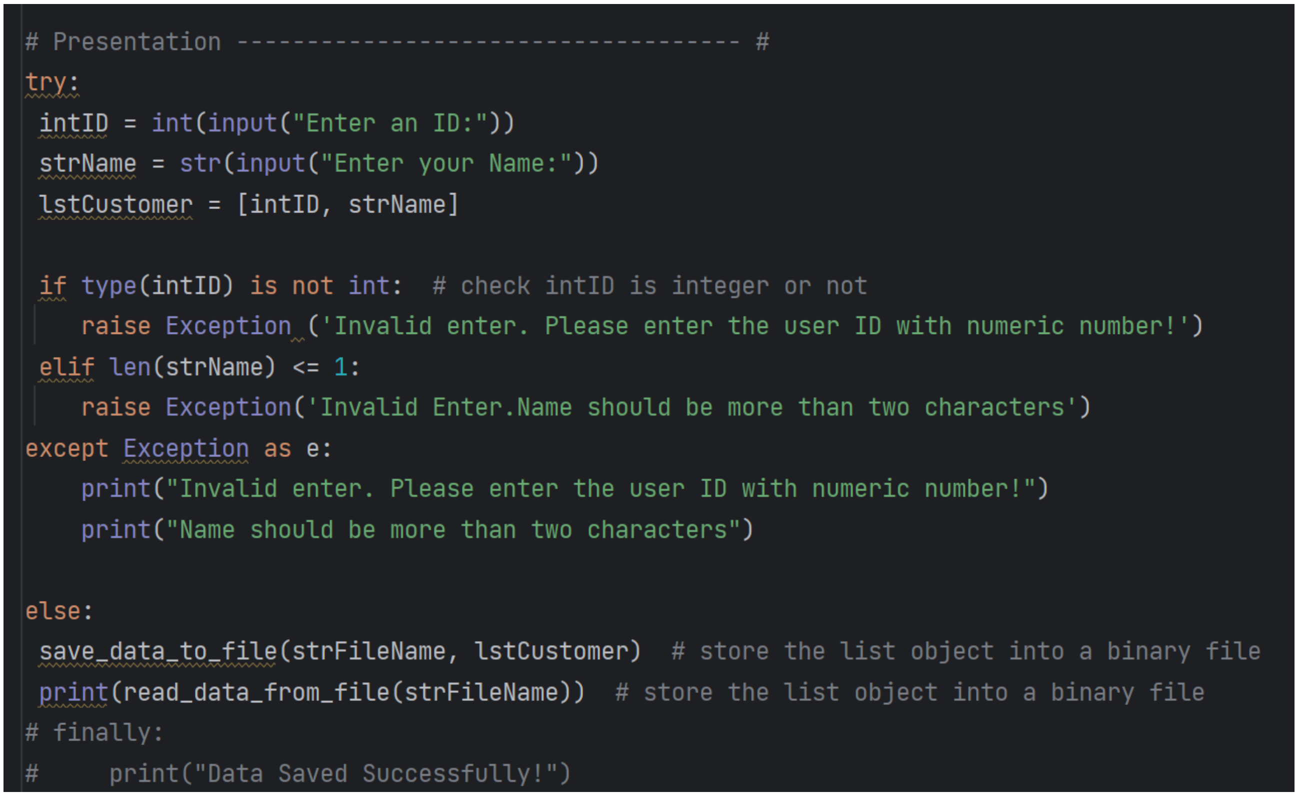
It runs well but shows error messages all the time.



```
"C:\Users\Mitsuyo Kawano\_PythonClass\Assingment07\Scripts\
Enter an ID:1
Enter your Name:Bob Smith
Error!
Executing Finally

Process finished with exit code 0
```

Figure 14 A Screenshot of the test result (PyCharm)

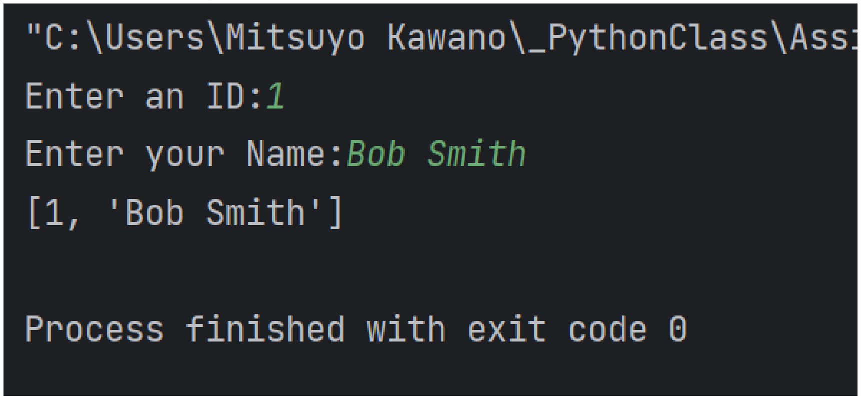
I did some research online, and here is the final script I wrote. I commented out the finally statement part since this part is always printed out no matter what, and I felt this is not necessary for this script.



```python
# Presentation ------------------------------------ #
try:
 intID = int(input("Enter an ID:"))
 strName = str(input("Enter your Name:"))
 lstCustomer = [intID, strName]

 if type(intID) is not int:  # check intID is integer or not
    raise Exception ('Invalid enter. Please enter the user ID with numeric number!')
 elif len(strName) <= 1:
    raise Exception('Invalid Enter.Name should be more than two characters')
except Exception as e:
    print("Invalid enter. Please enter the user ID with numeric number!")
    print("Name should be more than two characters")

else:
 save_data_to_file(strFileName, lstCustomer)  # store the list object into a binary file
 print(read_data_from_file(strFileName))  # store the list object into a binary file
# finally:
#     print("Data Saved Successfully!")
```

Figure 15 A Screenshot of the final script (PyCharm)

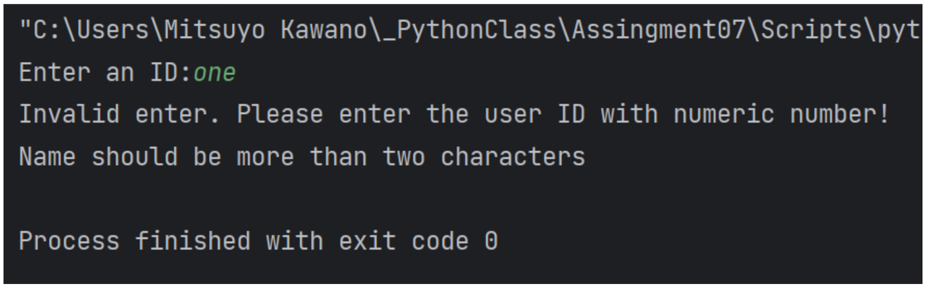
Runs successfully.



```
"C:\Users\Mitsuyo Kawano\_PythonClass\Ass
Enter an ID:1
Enter your Name:Bob Smith
[1, 'Bob Smith']

Process finished with exit code 0
```

Figure 16 A Screenshot of the test result (PyCharm)

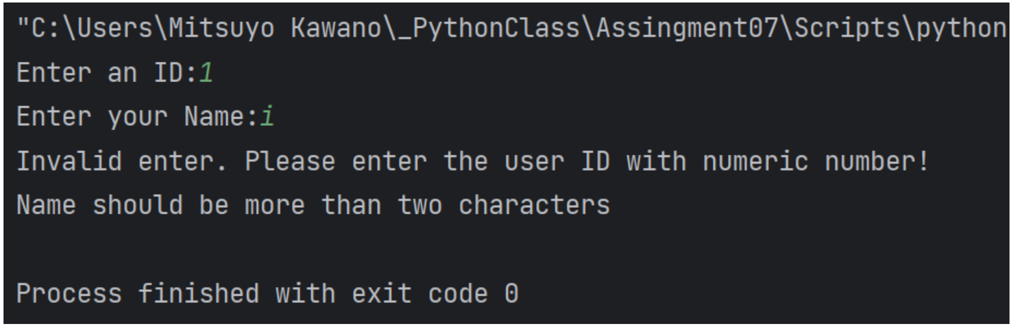
Runs successfully with invalid enter. (ID number was entered “one”)



```
"C:\Users\Mitsuyo Kawano\_PythonClass\Assingment07\Scripts\pyt
Enter an ID:one
Invalid enter. Please enter the user ID with numeric number!
Name should be more than two characters

Process finished with exit code 0
```

Figure 17 A Screenshot of the test result (PyCharm)

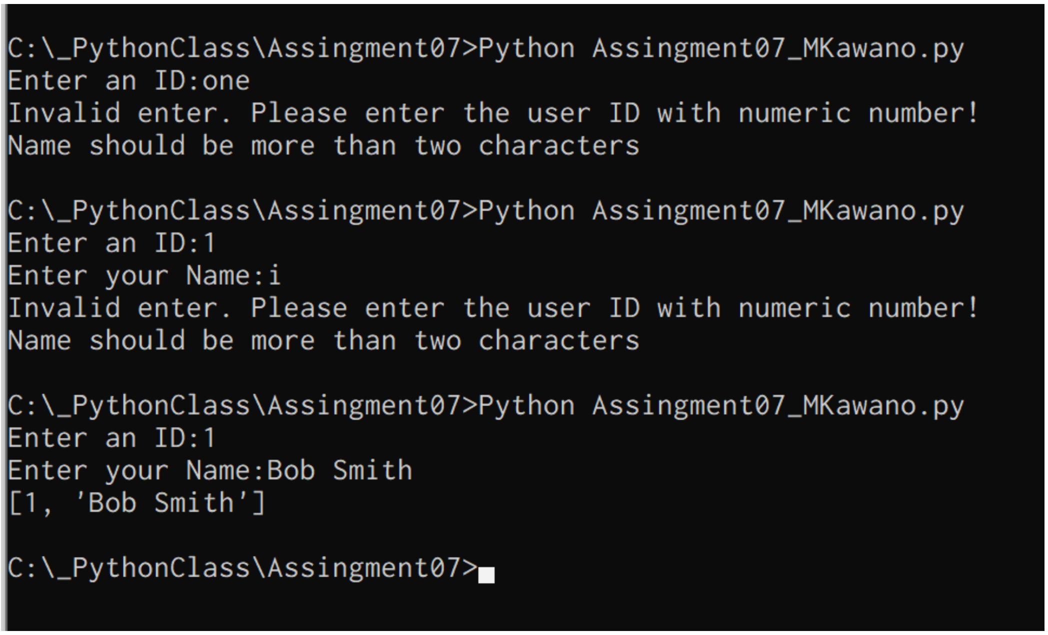
Runs successfully with invalid enter. (Name was entered “I” only one character)



```
"C:\Users\Mitsuyo Kawano\_PythonClass\Assingment07\Scripts\python
Enter an ID:1
Enter your Name:i
Invalid enter. Please enter the user ID with numeric number!
Name should be more than two characters

Process finished with exit code 0
```

Figure18 A Screenshot of the test result (PyCharm)

Runs successfully in Command Line.



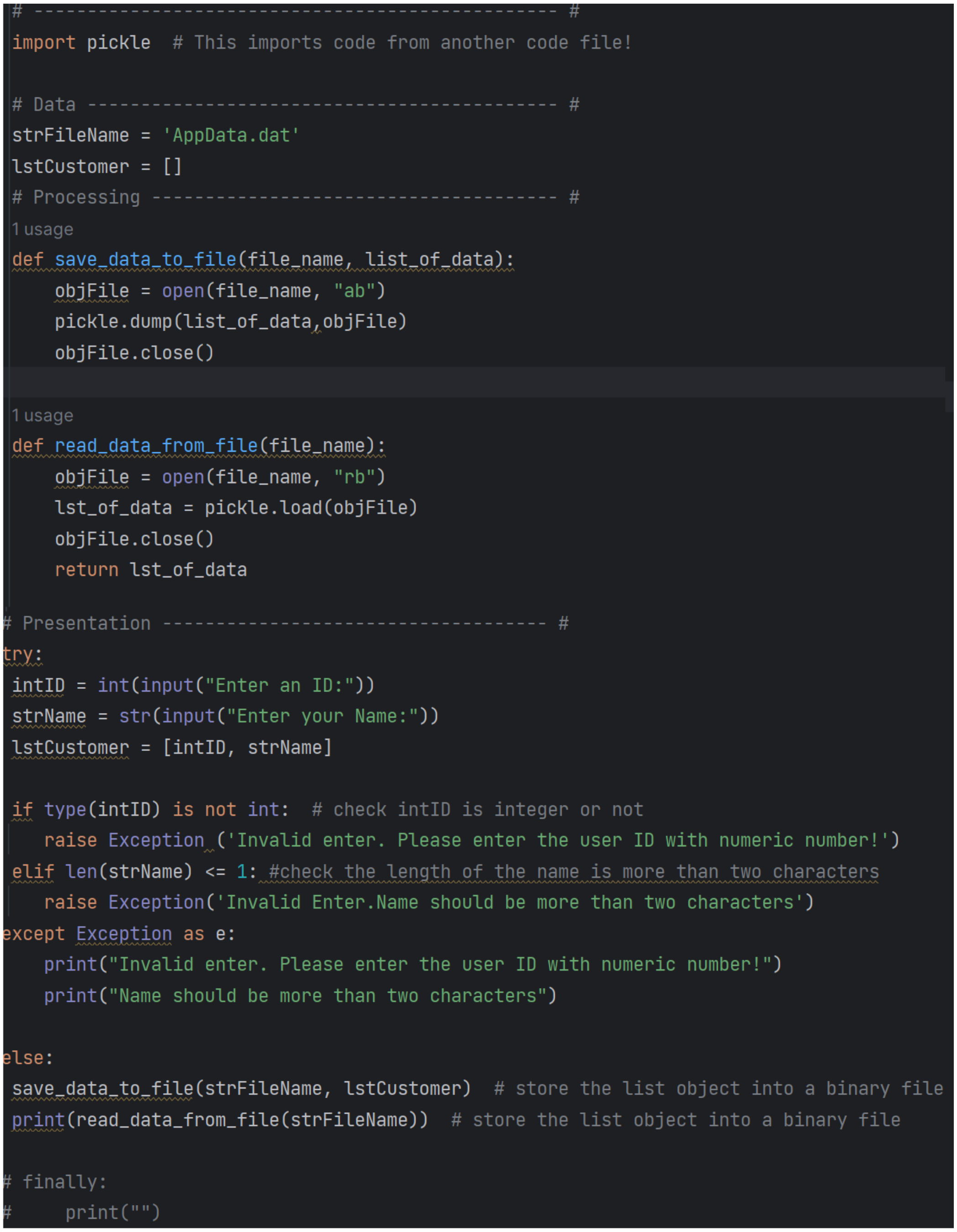
```
C:\_PythonClass\Assingment07>Python Assingment07_MKawano.py
Enter an ID:one
Invalid enter. Please enter the user ID with numeric number!
Name should be more than two characters

C:\_PythonClass\Assingment07>Python Assingment07_MKawano.py
Enter an ID:1
Enter your Name:i
Invalid enter. Please enter the user ID with numeric number!
Name should be more than two characters

C:\_PythonClass\Assingment07>Python Assingment07_MKawano.py
Enter an ID:1
Enter your Name:Bob Smith
[1, 'Bob Smith']

C:\_PythonClass\Assingment07>
```

Figure 19 A Screenshot the running script (Command Line)

Summary

Here is the summary of steps to complete this assignment.

1. Start with simple script using Lab7-1
2. Think about possible errors and modify the script (ID Number)
3. Think of another possible error and modify the script (Name)



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

# Data -------------------------------------------- #
strFileName = 'AppData.dat'
lstCustomer = []
# Processing -------------------------------------- #
1 usage
def save_data_to_file(file_name, list_of_data):
    objFile = open(file_name, "ab")
    pickle.dump(list_of_data,objFile)
    objFile.close()


1 usage
def read_data_from_file(file_name):
    objFile = open(file_name, "rb")
    lst_of_data = pickle.load(objFile)
    objFile.close()
    return lst_of_data

# Presentation ------------------------------------ #
try:
intID = int(input("Enter an ID:"))
strName = str(input("Enter your Name:"))
lstCustomer = [intID, strName]

if type(intID) is not int:  # check intID is integer or not
    raise Exception ('Invalid enter. Please enter the user ID with numeric number!')
elif len(strName) <= 1: #check the length of the name is more than two characters
    raise Exception('Invalid Enter.Name should be more than two characters')
except Exception as e:
    print("Invalid enter. Please enter the user ID with numeric number!")
    print("Name should be more than two characters")

else:
save_data_to_file(strFileName, lstCustomer)  # store the list object into a binary file
print(read_data_from_file(strFileName))  # store the list object into a binary file

# finally:
#     print("")
```

Figure 20 A Screenshot of the final script (PyCharm)