Name: Marita Quiroz
Date: May 31, 2023
Course: IT FDN 110 A Sp 23: Foundations Of Programming: Python
Assignment: 07
Github URL: https://github.com/mquiroz1/IntroToProg-Python-Mod07

Assignment 07 – INTRO TO PROGRAMMING (PYTHON)

Introduction

The purpose of this assignment is to modify create a program to demonstrate error handling in Python and Python pickling.

Program

The header section (Figure 1) provides information about the title, developer name, date of origination, and change log. The change log has been updated to include my name, date, and basic information regarding what was modified or updated.

Assignment07.py

```
1  # ------------------------------------------------ #
2  # Title: Assignment 7
3  # Description: A simple program to store gym equipment data in a binary file
4  # ChangeLog: (Who, When, What)
5  # Marita Quiroz,<5.31.2023>,Created Script
6  # ------------------------------------------------ #
```

Figure 1: Program header

Figure 2 displays the initial section of code where any Python modules are imported. Importing modules will allow access to additional features and definitions Python offers.

```
7
8  # Import pickle module
9  import pickle
10
```

Figure 3 shows the first section proceeding the header, which is where the data is initially handled. The Data section is where the variable declaration will occur. Here, the list, dictionary, and file data related variables are listed.

```
11  # Data ---------------------------------------------- #
12
13  strFileName = 'EquipInventory.dat'
14  eName = input("Enter equipment name: ")
15  # Specify exception type
16  # Ask user for quantity
17  try:
18      intQty = int(input("Enter quantity: "))
19      lstEquipment = [eName, intQty]
20  # If input is not a valid number (ValueError),
21  # message will be displayed back to user
22  except ValueError:
23      print("That was not a valid quantity!")
24      # Exit program
25      exit()
26
```

Figure 3: Data section with variable declaration

This section is also where an example of error handling is placed. After the equipment variable is declared, using user input, a try statement with an except clause is placed. The try statement will ask the user for the equipment quantity, which will then be converted to an integer type. If this is successfully input as an integer, the next line will be run. The next line adds the equipment name and quantity, as input by the user.

If the user does not enter a valid number, which may be anything other than a number, the ValueError exception type will be called. The message “That was not a valid quantity!” will then be displayed back to the user, and the program will exit.

The next section will begin the data processing (Figure 4). The binary file will be opened in write mode. The “wb” mode allows the file to be opened and written to, while overwriting any existing data within the file. If the file does not exist, the file will then be created. The pickle function, pickle.dump, will be called. Pickle.dump will take the equipment list and file as

parameters. The function will then write the objects to file. The file is closed once this has been completed.

```
28  # Processing ------------------------------------ #
29
30  # Open Equipment Inventory binary file in write mode
31  # If the file doesn't exist, one will be created
32  objFile = open("EquipInventory.dat", "wb")
33  # Use pickle.dump to write the object to file
34  pickle.dump(lstEquipment, objFile)
35  # Close file
36  objFile.close()
37
```

Figure 4: Open and write to binary file

The next function used is pickle.load (Figure 5). The file is opened again, but this time in read mode. Pickle.load will read the row of data in the file, and is saved as a string object. The file is then closed. The row of data is then displayed back to the user from the string variable. This verifies that the data was saved to file.

```
38  # Open Equipment Inventory binary file in read mode
39  objFile = open("EquipInventory.dat", "rb")
40  # Use pickle.load to load row of data
41  objFileData = pickle.load(objFile)
42  # Close file
43  objFile.close()
44  # Print current contents of file
45  print("Current file: ")
46  print(objFileData)
47
```

Figure 5: Open and read binary file

One of the challenges faced was exiting the program when the exception was raised. Since this is a simple demonstration, I did not use a menu for the user to go back to, so I needed the

program to exit. When the program exit function was not in, the program tried to continue to the file handling section, which caused an error for not receiving the correct value type (Figure 6).

```
/Users/maritaquiroz/Documents/_PythonClass/Assignment07/venv/bin/python /Users/maritaquiroz/Documents/_Py
Enter equipment name: barbell
Enter quantity: h
That was not a valid quantity!
Traceback (most recent call last):
  File "/Users/maritaquiroz/Documents/_PythonClass/Assignment07/Assignment07.py", line 25, in <module>
    lstEquipment = [eName, intQty]
                           ^^^^^^
NameError: name 'intQty' is not defined

Process finished with exit code 1
```

Figure 6: Exception raised without program exit

Figure 7 shows the code with the exit function included.

```
16  # Ask user for quantity
17  try:
18      intQty = int(input("Enter quantity: "))
19      lstEquipment = [eName, intQty]
20  # If input is not a valid number (ValueError),
21  # message will be displayed back to user
22  except ValueError:
23      print("That was not a valid quantity!")
24      # Exit program
25      exit()
26
```

Figure 7: Exit function added to except clause

Figure 8 shows the program handling the except clause and proceeding to exit.

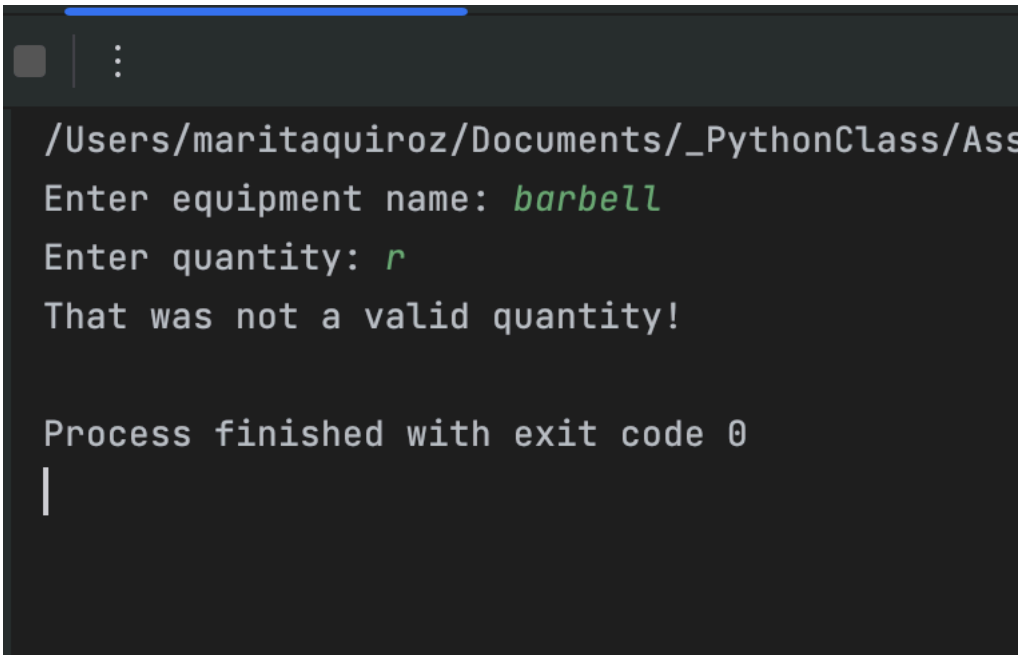

Figure 8: Program exiting after except clause is raised

Figure 9 shows the program run successfully in Pycharm, using a valid number for the equipment quanitity.

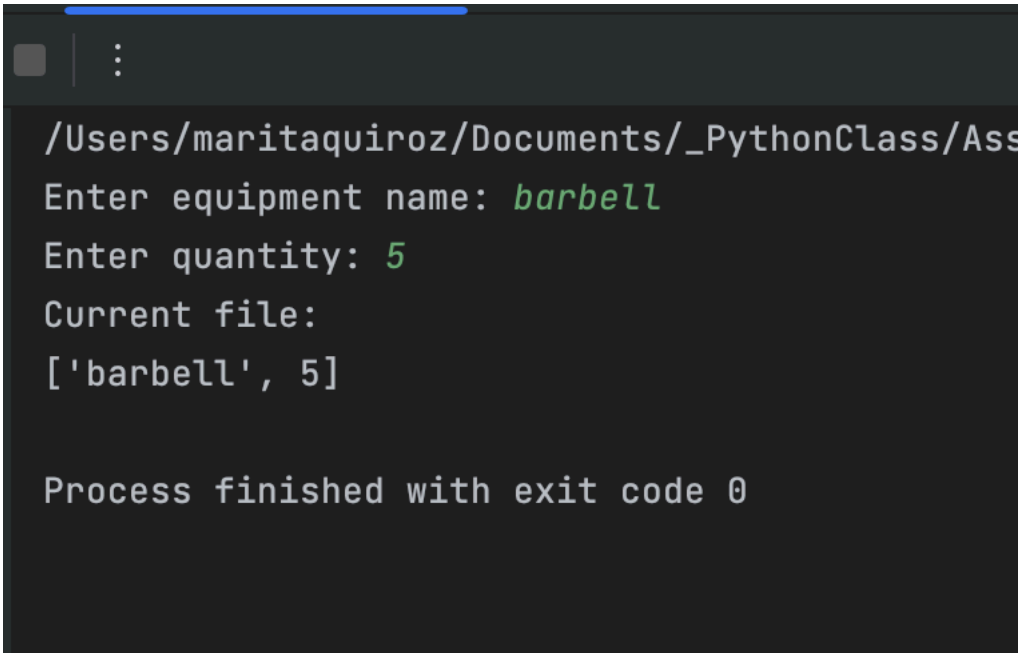

Figure 9: Program run successfully in Pycharm

The next figure (Figure 10) shows the basic program run successfully in the terminal/ console, using a number as input for quantity.

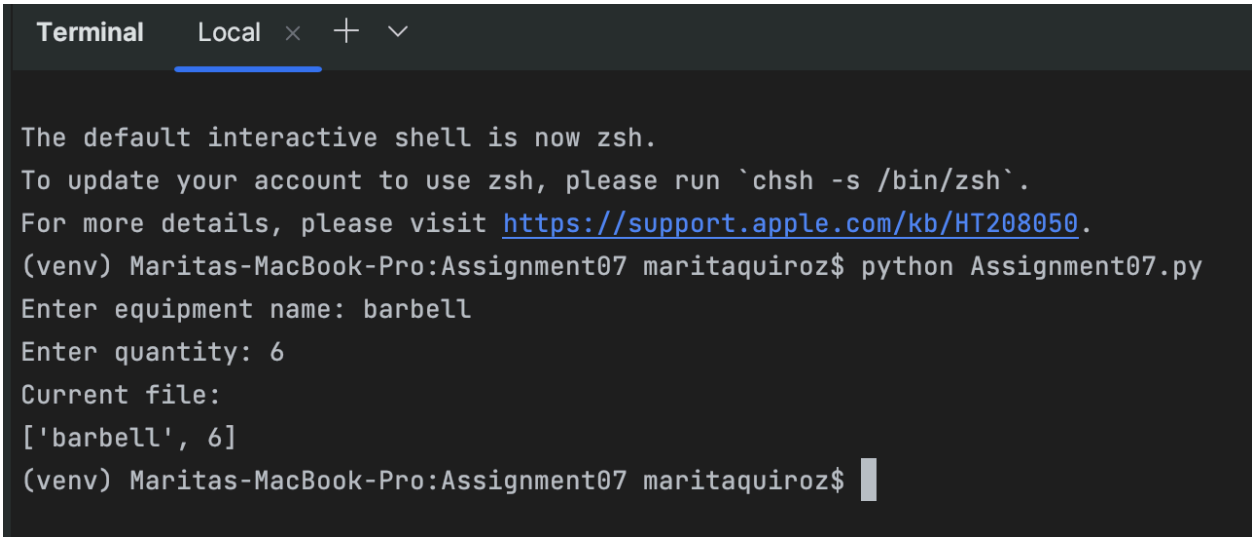

Figure 10: Program run successfully from the terminal

Summary

This program was create in a simple format to demonstrate usage of the Python pickle module. The program also provided an example of handling an exemption. For this assignment, I consulted the lecture, labs, the book, and the python documentation on the internet for file manipulation. I used the https://docs.python.org/ website to consult documentation about pickling and handling exemptions. I also referenced previous labs from other modules.