Alexandra Rochester

March 1st 2021

Intro to Programming Python

Assignment07

Files and Exceptions

# Introduction

-readlines() function returns data as a list, whereas read() function returns data as a string.

- Pickling saves data in binary format instead of as plain text. It can obscure the file’s content and reduce its size. It does not encrypt it.

- try-except blocks of code allow you customize how your program handles errors instead of just letting Python do that for you. It is helpful whenever you think human interaction might cause a problem in the programming.

- 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. You can capture the Exception object in the except section of a try-except block and extract the error message.

# Step 1

I found Chapter 7 in Python Programming for the Absolute Beginner really helpful in breaking down what pickling is. I also looked up pickling on the web and like that this [article](https://www.geeksforgeeks.org/understanding-python-pickling-example/) explained the benefits of pickling. Some of those are that It serializes the objects so later objects won’t be serialized again and that pickle stores the object once, and so all other references point to that master copy anywhere in the code.

# Step Two

I followed along with the examples given in chapter 7 but changed the object names to my own list. The code below creates lists under object names, then dumps those lists into a .dat file using pickle.dump()

import pickle, shelve  
  
print("Types of Fruit")  
fruit = ["apple, orange, pear"]  
shape = ["whole", "sliced", "diced"]  
origin = ["local", "global", "gmo"]  
  
f = open("fruit1.dat", "wb") # this will create and write to a binary file. if the file exists it will overwrite data  
# pickle.dump puts each dataset into the file as one object  
pickle.dump(fruit, f)  
pickle.dump(shape, f)  
pickle.dump(origin, f)  
f.close()

# Step Three

This code opens the data from the file back into lists, then prints them.

Text

Description automatically generated

# Step Four

I converted the pickling functions into a custom function so that it would be called whenever input is entered.

Text

Description automatically generated

# Step Five

I created while loops for the try except examples. I wanted to add more fruit and shape strings to the fruit and shape lists, and a new integer to the cost list.

Text

Description automatically generated

# Step Six

I had to change the tabbing and remove the break for the integer try statement for the cost input. I also printed some of the python automated error messages to see how it handles structured errors.

Text

Description automatically generated

# Step Seven

I then appended the lists with the input, printed them and then printed the appended lists using the pickling custom function.

# We then check the input has been acknowledged by printing them  
print(fruitInput + "," , shapeInput + ", ", costInput)  
  
# those inputs are appended onto the fruit, shape and cost lists  
fruit.append(fruitInput)  
shape.append(shapeInput)  
cost.append (int(costInput)) # this converts the input into an integer  
  
# we print the appended lists using the pickle function  
print("This is the new appended set of lists" '\n')  
save\_and\_load\_data(fruit,shape,cost)

# Step Eight

I then ran the code in PyCharm and IDLE to ensure it ran in both systems.

Text

Description automatically generated

Graphical user interface, application, Word

Description automatically generated