

Paul Mitchell

December 1, 2020

IT FDN 110 A - Intro to Python Programming

Assignment 7

<https://github.com/pwm94/IntroToProg-Python-Mod07>

## Assignment 7 - Using Exceptions & Pickles

### **Introduction & Problem Statement**

The task set forth in this assignment was to create a script that utilized both exceptions and pickles and finding a way to integrate both into the same script. For this assignment, I decided to continue the trend of using a menu of numerical options, using the exception function to ensure proper input, and using pickling and unpickling to save and store data in an external file.

### **Creating the Menu**

First I decided to model my menu off of a Dog Adoption Database. To do so, I created the menu with the following options: (1) Adding Dogs to the List; (2) Displaying the Current Dogs; (3) Unpickle & Reload List of Dogs from File; (4) Pickle and Save to File; (5) Exit Program. See Figure 1 for Menu display. User input was collected as a string input and saved as Variable UserInput.

```
OptionVar = str(0)
StoredData = []
while (True):
    print("\nDatabase of Adoptable Dogs & Their Ages:")
    print("1) Add Dogs to List")
    print("2) Display Current Dogs")
    print("3) Unpickle & Reload List of Dogs from File")
    print("4) Pickle and Save to File")
    print("5) Exit Program")
    UserInput = str((input("Which option would you like to perform:| [1] to [5]? ")))
```

Figure 1

## **Exceptions**

I decided to utilize the exception function to ensure the user only input 1 through 5 as options in the menu. I accomplished this by converting the UserInput string to integer variable y. I declared variable x to be 5.0 and set variable z to be  $x - \text{abs}(y)$ . This would ensure any number greater than 5 would return a negative value. Combined with a squareroot function of z, this would test for an exception. Similarly I set a testforzero variable to equal  $1 / y$  in case the user input 0. I realize this would allow a -1 through -5 input, but decided that those would still be allowed to correlate to their positive inverses. See Figure 2.

```

try:
    import math
    x = int(5)
    y = int(UserInput)
    z = x - abs(y)
    test = math.sqrt(z)
    testforzero = 1 / y
    OptionVar = UserInput
    if (OptionVar == str(1)):...
    elif (OptionVar == str(2)):...
    elif (OptionVar == str(3)):...
    elif (OptionVar == str(4)):...
    elif (OptionVar == str(5)):...

```

Figure 2

For the Exception messages, I used ValueError and ZeroDivisionError. See Figure 3.

```

except ValueError:
    print("\nError: Invalid Input.")
    print("Please Ensure your Selection is Between [1] and [5]")
    print()
    continue
except ZeroDivisionError:
    print("\nError: Invalid Input.")
    print("Please Ensure your Selection is Between [1] and [5]")
    print()
    continue

```

Figure 3

### **Pickling and Menu Execution**

If the user correctly input 1 thorough 5 (or negative 1 through 5) as their menu choice, no exception would occur, and the While(True) loop would run. It contained if and elif statements for each menu item. Menu Items (1), (2), and (5) are the same as previous assignments, so I will not go into detailed descriptions. However, for Menu Items (3) and (4), new pickling script was

added to save the data as a pickle instead of a list or dictionary, as we did previously. See Figure 4.

```
elif (OptionVar == str(3)):
    import pickle
    pickle_in = open("file.pickle", 'rb')
    StoredData = pickle.load(pickle_in)
    print("Reload Complete!")
    print("***** The Current Dogs Available For Adoption Are [Name (Age)]: *****")
    for row in StoredData:
        print(row[0] + " (" + row[1] + ")")
    print("*****")
    print()
elif (OptionVar == str(4)):
    print("Would you like to save & pickle your data?")
    Var3 = str(input("Enter 'y' or 'n' "))
    if (Var3 == str("y")):
        import pickle

        pickle_out = open("file.pickle", 'wb')
        pickle.dump(StoredData, pickle_out)
        pickle_out.close()
    continue
```

Figure 4

### **Final Code**

Final code is attached and a demonstration of it working in an IDLE shell is below (Figures 5 and 6). Figure 7 shows the picked data saved externally.

```

Database of Adoptable Dogs & Their Ages:
1) Add Dogs to List
2) Display Current Dogs
3) Unpickle & Reload List of Dogs from File
4) Pickle and Save to File
5) Exit Program
Which option would you like to perform:z [1] to [5]? 2
***** The Current Dogs Available For Adoption Are [Name (Age)]: *****
*****

Database of Adoptable Dogs & Their Ages:
1) Add Dogs to List
2) Display Current Dogs
3) Unpickle & Reload List of Dogs from File
4) Pickle and Save to File
5) Exit Program
Which option would you like to perform:z [1] to [5]? 3
Reload Complete!
***** The Current Dogs Available For Adoption Are [Name (Age)]: *****
Shasta (1)
Samson (2)
Apollo (3)
Doug (4)
*****

```

Figure 5

```

Database of Adoptable Dogs & Their Ages:
1) Add Dogs to List
2) Display Current Dogs
3) Unpickle & Reload List of Dogs from File
4) Pickle and Save to File
5) Exit Program
Which option would you like to perform:z [1] to [5]? 1

Enter the Name and Age for the new dog below
Enter a Name: Teddy
Enter an Age: 5

Database of Adoptable Dogs & Their Ages:
1) Add Dogs to List
2) Display Current Dogs
3) Unpickle & Reload List of Dogs from File
4) Pickle and Save to File
5) Exit Program
Which option would you like to perform:z [1] to [5]? 4
Would you like to save & pickle your data?
Enter 'y' or 'n' y

Database of Adoptable Dogs & Their Ages:
1) Add Dogs to List
2) Display Current Dogs
3) Unpickle & Reload List of Dogs from File
4) Pickle and Save to File
5) Exit Program
Which option would you like to perform:z [1] to [5]? 3
Reload Complete!
***** The Current Dogs Available For Adoption Are [Name (Age)]: *****
Shasta (1)
Samson (2)
Apollo (3)
Doug (4)
Teddy (5)
*****

```

Figure 6

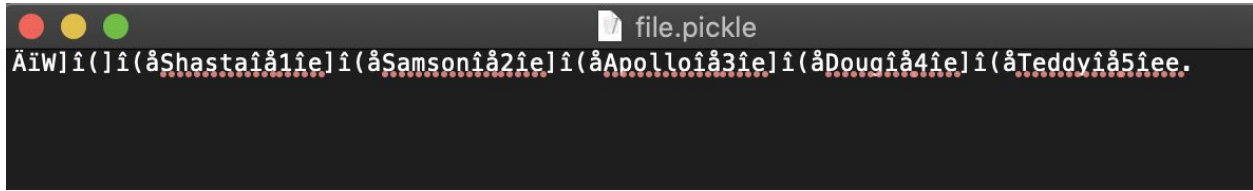


Figure 7

## **Resources**

The resources I used to complete this assignment are as follows:

- <https://docs.python.org/3/library/exceptions.html#ValueError>
  - I liked this resource because it was quickly searchable and had great description of many different types of exceptions
- <https://medium.com/dev-genius/python-error-handling-8bed3f5b5769>
  - I liked this resource because of the way it demonstrated multiple exception stacking
- <https://www.journaldev.com/33500/python-valueerror-exception-handling-examples>
  - I liked this resource because it used clear examples
- <https://www.youtube.com/watch?v=2Tw39kZlbhs>
  - I liked this resource because it gave a step by step guide on how to use the pickle and unpickle functions
- <https://medium.com/swlh/pickling-in-python-ac3c7a045ae5>
  - I liked this resource because it clearly outlined what pickles are (seems similar to blockchain) and make them easily understandable