Jeff Wichmann

June 1, 2020

IT FDN 100 A Sp 20: Foundations of Programming: Python

Assignment 7

<https://github.com/jwichmann51/IntroToProg-Python-Mod07>

Python: How to use exception handling and the pickling module

# Introduction

This paper introduces the concepts of exception handling and pickling. I explain both concepts and provide examples using both the Michael Dawson textbook (Dawson M., Python Programming, Third Edition, Course Technology, a part of Cengage Learning, 2010) and certain websites. The assignment also provides a program that demonstrates exception handling and pickling. My goal is to create a document that a Python user can utilize to build functions as necessary to complete this coding exercise. Additionally, I can use this as a reference to refresh my memory in the future, if necessary. A subsequent draft of this assignment will be completed with the corrected script. I am using Mac OS.

# Exception handling

Exception handling is when a program is purposefully designed to respond in a certain way when Pythion runs into an error. As Michael Dawson explains, when Python runs into an error, the program “raises an exception” (Dawson M., Python Programming, Third Edition, Course Technology, a part of Cengage Learning, 2010). When Python runs into an exception, it stops what it is doing and provides details on the error. Figure 1 shows example of this.

A screenshot of a cell phone

Description automatically generated

**Figure 1: Example Python code in PyCharm created by Michael Dawson**

Figure 2 shows the error code in Python. This error code is not very helpful to anyone except a programmer. Non-programmers would do better with a gentler response or a hint about what they should do differently. Exception handing provides this.

A screenshot of a cell phone

Description automatically generated

**Figure 2: Output in PyCharm**

As explained by the website Tutorialspoint, “Python provides two very important features to handle any unexpected error in your Python programs and to add debugging capabilities in them −

Exception Handling − This would be covered in this tutorial. Here is a list [of] standard Exceptions available in Python: Standard Exceptions.

Assertions − This would be covered in Assertions in Python tutorial” (tutorialspoint, 2020, <https://www.tutorialspoint.com/python/python_exceptions.htm>) (external site). I thought this comment was helpful because it makes the point that exception handling is not the only way to manage unexpected errors.

The most basic way to deal with an exception is with a *try* statement with an *except* clause. As Dawson succinctly says, a programmer can use a try statement to “section off some code that could potentially raise an exception”. Dawson continues that you can then “write an except clause with a block of statements that are executed… if an exception is raised”.

Figure 3 shows example code using the *try* statement with an *except* clause.

A screenshot of a cell phone screen with text

Description automatically generated

**Figure 3: Example Python code in PyCharm written by Michael Dawson**

The code in figure 3 asks the user for a number. If a string is provided (as opposed to a number), the message “Something went wrong!” is returned. The program then asks for a number again. If another string is entered, the program provides a better hint, saying “That was not a number!” (Figure 4).

**A screenshot of a cell phone

Description automatically generated**

**Figure 4: Output in PyCharm**

Different kinds of errors cause different types of exceptions (Dawson M., Python Programming, Third Edition, Course Technology, a part of Cengage Learning, 2010). Dawson explains there are over two dozen exception types. Tutorialspot provides a list of 30 exception types at <https://www.tutorialspoint.com/python/python_exceptions.htm> (external site).

# Pickling

Text files are user friendly because you can read and write to them, but they are not appropriate for storing a complex set of data (Dawson M., Python Programming, Third Edition, Course Technology, a part of Cengage Learning, 2010). Pickling is a tool in Python to preserve a complex set of data. Michael Dawson provides the example code pickle\_it.py which I edited to some degree for clarity. This code is shown in figure 5.

A screenshot of a cell phone

Description automatically generated

**Figure 5: Example Python code in PyCharm written by Michael Dawson**

The first thing Dawson did was to import two modules, the pickle module and the shelve module (Figure 6). The pickle module enables the user to pickle and store complex data in a file. The shelve module allows you to store and access pickled objects. When using a text file, you write a string of characters into the text file. With a binary or pickled file, you are writing an object to a binary file (Root R., Assignment07 IT FDN 100 A Sp 20: Foundations of Programming: Python, University of Washington, 2020).

A picture containing food

Description automatically generated

**Figure 6: Example Python code in PyCharm written by Michael Dawson**

Dawson then creates three lists (Figure 7) which will be pickled and written to a binary file.

A black sign with white text

Description automatically generated

**Figure 7: Example Python code in PyCharm written by Michael Dawson**

The next step is to open the file (Figure 8).



**Figure 8: Example Python code in PyCharm written by Michael Dawson**

The data is then pickled and stored. The file is then closed (Figure 9).

A close up of a sign

Description automatically generated

**Figure 9: Example Python code in PyCharm written by Michael Dawson**

The program then proceeds to retrieve and unpickle the lists (Figure 10). The program reads the three pickled objects in the file which are variety, shape, and brand.

A black sign with white text

Description automatically generated

**Figure 10: Example Python code in PyCharm written by Michael Dawson**

# The Assignment

The script assignment for Module 7 is to create a script which demonstrates both exception handling and the pickling module (Root R., Assignment07 IT FDN 100 A Sp 20: Foundations of Programming: Python, University of Washington, 2020). This script follows in figure 11.

# ------------------------------------------------- #  
# Title: Assignment07  
# Description: Exception handling and pickling demo  
# ChangeLog: (Who, When, What)  
# JWichmann,5/31/2020,Created Script  
# JWichmann,6/1/2020, final review of the script  
# ------------------------------------------------- #  
  
# Data -------------------------------------------- #  
strTicker1 = str(input("Enter a Ticker: ")) # The user is asked to enter three stock tickers.  
strTicker2 = str(input("Enter a 2nd Ticker: "))  
strTicker3 = str(input("Enter a 3rd Ticker: "))  
try:  
 intPrice1 = int(input("Enter a price: ")) # The user is asked to enter three stock prices.  
except:  
 print("Please restart the program and enter prices only.") # Exception handing encourages the user enter a number and not a string.  
  
try:  
 intPrice2 = int(input("Enter a 2nd price: "))  
except:  
 print("Please restart the program and enter prices only.")  
  
try:  
 intPrice3 = int(input("Enter a 3rd price: "))  
except:  
 print("Please restart the program and enter prices only.")  
  
# Process -------------------------------------------- #  
  
import pickle  
  
print("Research lists.")  
prices = [intPrice1, intPrice2, intPrice3]  
ticker = [strTicker1, strTicker2, strTicker3]  
f = open("coverage.dat", "wb")  
pickle.dump(prices, f)  
pickle.dump(ticker, f)  
f.close()  
  
print("\nUnpickling lists.")  
f = open("coverage.dat", "rb")  
prices = pickle.load(f)  
ticker = pickle.load(f)  
  
# Presentation -------------------------------------------- #  
  
print(ticker)  
print(prices)  
f.close()  
  
input("\n\nPress the enter key to exit.")

**Figure 11: Example Python Script**

# Images showing the script working

The output file shows the program working correctly (Figure 12).

A screenshot of a cell phone

Description automatically generated

**Figure 12: Output in PyCharm**

# Running the script in OS command

Figure 13 shows the script running in OS command.

A screenshot of a cell phone

Description automatically generated

**Figure13: Output in OS command**

# Summary

In summary, this document outlines the steps and process to create a programming script. The goal was to introduce and explain exception handling and the pickling module and create code showing both. This document acts as guide to a Python user or for me, if I need a refresher at some point in the future.