**Kate Pollock**

**November 28, 2021**

**Foundations of Programming: Python**

**Assignment07**

**GitHubURL <** [**https://github.com/katepollock/IntroToProg-Python-Mod07**](https://github.com/katepollock/IntroToProg-Python-Mod07)**>**

# **Password Creation – Assignment07**

# Introduction

In Module 7, we learned about exception handling and pickling, both through review of the module and our own web research. Unlike other modules we made our own scripts to demonstrate understanding of these concepts. I chose to create a script in which a user enters or changes a password, demonstrating both of these concepts.

# Research

I found the following websites to be helpful regarding exception handling:

[Python Exceptions Handling (w3schools.in)](https://www.w3schools.in/python-tutorial/exception-handling/)

[Python Exception Handling Using try, except and finally statement (programiz.com)](https://www.programiz.com/python-programming/exception-handling)

I found the above websites to be helpful regarding exception handling because:

* W3 Schools provided a good overview of exceptions in Python including types of statements, why and when they are used as well as examples of different exceptions.
* Programwhiz provided a good overview as well as videos describing built in exceptions, how to handle exceptions and user defined/custom exceptions.

I found the following website to be helpful regarding pickling:

[The Python pickle Module: How to Persist Objects in Python – Real Python](https://realpython.com/python-pickle-module/)

I felt that the above site had a good explanation of the serialization process. It also provided a description of the different modules for serialization in Python.

# Planning my “Password Creation ” Script

In planning my password creation script, I tried to utilize mostly functions in my code as many of the tasks were repetitive. Additionally, I created two custom exceptions to check the validity of the password entered and whether the passwords match when creating a new password. It was necessary tom import pickle as I would be saving the passwords in a binary format (***Figure 1***).

*# ---------------------------------------------------------------------------- #  
# Title: Assignment 07 - Password Generator  
# Description: Working with exceptions and pickling on a password  
# program, when program starts, import pickle,  
# create functions to read and save to file,  
# check password validity, then call functions,  
# use exceptions when no file to read and password  
# valid or doesn't agree.  
# ChangeLog (Who,When,What):  
# KPollock, 11.28.2021  
# ---------------------------------------------------------------------------- #  
  
# Data ---------------------------------------------------------------------- #  
# Declare variables, exception classes***import** pickle  
  
  
**class** PasswordDoesNotMeetCriteriaException(Exception):  
 **pass  
  
  
class** PasswordMismatch(Exception):  
 **pass**strFileName = **'PassData.dat'**

***Figure 1 – Script Heading, Declare Classes and Variable***

Main Body of Password Script

The following are some of the main features of my script (***See Figure 2***):

* Function enforce\_password\_validity checks for each requirement or rule for password generation. If any of the criteria are not meant, the custom exception PasswordDoesNotMeetCriteria is raised.
* Function save\_data\_to\_file saves the password in binary format to the file. I used write in order to write over the previous password with a new one.
* Function read\_data\_from\_file loads the password back into memory.
* Function main is the function that will be called in the output section of the script. It uses a try-except block to catch FileNotFound error which indicates a new user. In this case, the function set password is called. It additionally offers the user a choice of menu options.
* Functions set\_password, change\_password and authenticate are utilized for menu options 2 and 3). The custom exceptions are raised when a user does not enter a password that meets the criterion or they do not replicate their initial password entry in the case of setting a new password or changing a password (PasswordMismatch).

*# Processing --------------------------------------------------------------- #  
# Rules for password generation  
# Password must be 8 characters or more  
# Password must contain both letters (upper and lower)  
# numbers and special characters, 1 of each***def** enforce\_password\_validity(password):  
 *""" Check that password agrees with "rules"* **:param** *password: (string) user password  
 """* count\_upper = 0  
 count\_lower = 0  
 count\_digit = 0  
 count\_special = 0  
 **for** ch **in** password:  
 **if** ch.isupper():  
 count\_upper += 1  
 **elif** ch.islower():  
 count\_lower += 1  
 **elif** ch.isdigit():  
 count\_digit += 1  
 **else**:  
 count\_special += 1  
  
 **if not** (count\_upper > 0 **and** count\_lower > 0 **and** count\_digit > 0 **and** count\_special > 0 **and** len(password) >= 8):  
 **raise** PasswordDoesNotMeetCriteriaException  
  
  
**def** save\_data\_to\_file(file\_name, password):  
 *""" Saves password to file using pickle* **:param** *file\_name: (string) with name of file* **:param** *password: (string) user password  
 """* file = open(file\_name, **"wb"**)  
 pickle.dump(password, file)  
 file.close()  
  
  
**def** read\_data\_from\_file(file\_name):  
 *""" Reads data from a file into a variable (string)* **:param** *file\_name: (string) with name of file* **:return***: password: (string) user password  
 """* file = open(file\_name, **"rb"**)  
 password = pickle.load(file)  
 file.close()  
 **return** password  
  
  
**def** print\_menu\_choices():  
 *""" Display a menu of choices to the user* **:return***: choice: (string) user choice  
 """* print(**"""  
 Please select from menu options:  
 1) Authenticate user   
 2) Change password  
 3) Exit  
   
 """**)  
 choice = input(**'Enter menu choice: '**)  
 **return** choice  
  
  
**def** main():  
 *""" calls menu choice and proceeds based on choice  
 """* **try**:  
 read\_data\_from\_file(strFileName)  
 **except** FileNotFoundError:  
 print(**'Welcome new user. No password is on file. Please enter your initial password.'**)  
 set\_password()  
  
 **while True**:  
  
 strMenuChoice = print\_menu\_choices()  
  
 **if** strMenuChoice == **'1'**:  
 authenticate()  
 **elif** strMenuChoice == **'2'**:  
 change\_password()  
 **elif** strMenuChoice == **'3'**:  
 print(**'Goodbye!'**)  
 **break  
 else**:  
 print(**'Please enter [1 to 3]'**)  
  
  
**def** change\_password():  
 *""" calls authenticate and if True, calls set\_password  
 """* **if** authenticate():  
 set\_password()  
  
**def** set\_password():  
 *""" user inputs password and functions password\_validity  
 is checked and password is reentered to see if it matches.  
 If not, mismatch exception is passed.  
 """* **while True**:  
 **try**:  
 password = input(**'Enter new password: '**)  
 enforce\_password\_validity(password)  
 **if** password != input(**'Re-Enter new password: '**):  
 **raise** PasswordMismatch  
 print(**'New password accepted'**)  
 save\_data\_to\_file(strFileName, password)  
 **return  
 except** PasswordDoesNotMeetCriteriaException:  
 print(**'Password does not meet requirements.'**)  
 **except** PasswordMismatch:  
 print(**'Passwords did not match.'**)  
  
**def** authenticate():  
 *""" Checks if password in file and  
 if it is the same, if not passes PasswordMismatch* **:return***: true or false (bool)  
 """* **try**:  
 password\_exist = input(**'Please enter your current password: '**)  
 **if** password\_exist != read\_data\_from\_file(strFileName):  
 **raise** PasswordMismatch  
 print(**'Password accepted'**)  
 **return True  
 except** PasswordMismatch:  
 print(**'Passwords did not match.'**)  
 **return False***#Presentation I/O code*main()

***Figure 2 – Processing and Output code***

# Results of Script

I ran the code in the command prompt and the results were as expected (***Figure 3***).

Text

Description automatically generated

***Figure 3: Output in Command Prompt***

# Summary

I have written the Python program above by utilizing the new concepts learned in Module 7 of this course. These concepts include opening and writing data from and to a file in binary format (pickling) and using try-except error handling. I’m looking forward to learning about classes next week.