CS 1400 – Summer 2018
Assignment #10

Introduction
You will complete one program to give you experience with:

- **Classes**
- **Private data**
- **Special Methods on Strings**

Task 1
Go back to assignment #8, drawing the chessboard. Turn the program into an Object-Oriented program. Use the starter file main-task1.py. This is where your program starts, and it cannot be modified. Create a chessboard.py file that creates a Chessboard class. It should follow the pattern of Assn 8. However, drawing will be occur with the call of draw(), as indicated in main-tas1.py.

Rubric
5 pts: Software Development Lifecycle Plan (see assn #5 for description)
5 pts: Accurate UML diagram
10 pts: main-task1.py file is not changed
10 pts: All methods and data except draw() are private
10 pts: Proper nested loop to draw individual rectangles
10 pts: Follow pattern of using 3 methods to draw complete chessboard picture

Task 2
Programming Exercise 8.3

Read the exercise in the book first, so these instructions make sense. Add the following additional requirements for the password:

- **A password cannot contain the word 'password'**
- **A password cannot end with '123'**

Your program should define a class called Password, which is in its own file. Your program will prompt the user for a password, and after completing will ask the user if they want to enter another. The program should only ever create one instance of Password. Your Password class should have at least the following:

- **set_password() method**
- **is_valid() method**
  - **This should return a Boolean**
- **get_error_message() method**
  - **This should return a string that indicates the problems with the password**
  - **It should be called if is_valid() returns False**
  - **The is_valid() method can generate this string as it tests each password requirement**
    - **Hint: create a private instance variable called __message to save it**
  - **Example return string**
    - **"must have 8 characters\nmust have at least 2 digits\ncannot end in 123"**

Rubric
5 pts: Software Development Lifecycle Plan (see assn #5 for description)
5 pts: Accurate UML diagram
10 pts: Implementation meets all requirements listed in book and above
10 pts: Only a single instance of Password is created
15 pts: Private method for each password requirement test
5 pts: Proper output for all tested input

Starter

Use the main.py file to get started. Only replace the blanks and add missing methods. Do not modify the existing code otherwise

Helpers

Remember that you can find solutions to the even programming exercises online. Check Canvas for a link. These are suggestions for you to do. They are not part of the assignment, and you do not have to turn them in.

Exercises: 8.2, 8.4, 8.8

What/How To Turn In (READ THIS)

Submit your files on Canvas.

Due: June 14, 2018 (Note: This is Thursday)