```
# LCYS_2022 / L. M. Saxton

### Username and Password Registration and Login with Two Factor Authentication

### Purpose

This test code is meant to provide two functions:

1. register a username and password within a certain criteria
    - the criteria is meant to prevent the following attacks:
        - cross-site scripting
        - SQL injection
        - cross-site relay
        - man-in-the-middle

    - username criteria:
        - length between 5 and 16 characters
        - no spaces between characters
        - no special symbols

    - password criteria:
        - length between 8 and 16 characters
        - at least one uppercase character included
        - at least one digit included
        - no spaces between characters
        - at least one of the following special symbols included: ! @ # $ % &
        - produce a salted hash of the accepted password

2. provide a secure user login
    - the username must match with a case-specific cross-check
    - the password must match the salted hash archived
    - user must provide an OTP through Two Factor Authorization

### Installation

To install the test code, the code repository must be downloaded to a main repository on an OS. For example:

[Picture of main repo for code repo]

Because the test code does not have a GUI, it should be executed in a terminal, and can be done with the following steps:
1. Open the terminal program linked to the OS
2. Navigate to the code repository wherein the 'main.py' file is located
3. In the terminal, type the following:

    **Linux**

    `python3 main.py`

    **Powershell**

    ```
    python3
    >>> main.py
    ```

    MacOS

    `python3 main.py`

4. Press the 'enter' key

Code Construction

Object Oriented Programming

The code has been written with an object-oriented approach, as this approach "is defined by describing a collection of interacting objects via their data and behavior (Phillips , 2018:

The following classes are used to create a username and password:

    CreateUsername
    CreatePassword
    EncryptPass

These classes and functions can be viewed indepth in create_user_classes.py

The login program uses two separate classes:

    VerifyUser
    VerifyPass

These classes and functions can be viewed indepth in login_classes.py

Main Program File

The main.py program file has the following important characteristics:

- Imported classes along with `install pyotp` to implement Two Factor Authorization.

 - An empty dictionary `user_1 = {}` which stores the user profile input and provides cross-check information for the subsequent login program.

- while loops for username creation, password creation, username login, and password login

- A value of 1 assigned to successful input

- An empty list `success = []` which adds positive integer increment to the variable `success_count`

- A counter `success_count = 0`, which accepts or rejects input based on the the input value

- A print mechanism for hashed input

Information stored in the user dictionary is used to cross-check the username and password upon user login. A successful entry of the password is required to engage Two Factor Authori

Execution

Execution of the program results in the following:

![Program Snapshot](coding_project/program_snapshot.png)

The program also provides instructions for input that does not meet the username or password criteria for registration and/or login:

![Criteria Snapshot](coding_project/criteria_snapshot.png)

Testcode Data

Testcode performed for the various functions and the main program can be found in the repository `coding_project/testcode`.

References

desthuilliers, b. (2013) ValueError: invalid literal for int() with base 10: 'stop'. [online] Stack Overflow. Available at: https://stackoverflow.com/questions/16742432/valueerror-inv
```

```
Li, G. (2021) Byte string, Unicode string, Raw string â  A Guide to all strings in Python. [online] Medium. Available at: https://towardsdatascience.com/byte-string-unicode-string-r

Matthes, E. (2019) Python Crash Course. 2nd ed. San Francisco: No Starch Press.

Phillips, D. (2018) Python 3 Object Oriented Programming. 3rd ed. Birmingham: Packt Publishing, pp.7-33.

Python Pool. (2021) [Solved] RecursionError: maximum recursion depth exceeded while calling a Python object. [online] Available at: https://www.pythonpool.com/recursionerror-maximum-r

Rochlin, M. (2012) What's the difference between the print and the return functions? | Codecademy. [online] Codecademy. Available at: https://www.codecademy.com/forum_questions/518ffb

W3schools.com.(n.d.) Python Inheritance. [online] Available at: https://www.w3schools.com/python/python_inheritance.asp [Accessed 4 September 2022].
```