Programming with Loops and Functions

Password Verification

Make sure you have read and understood

* ***Week 3 Module Contents***, *and*
* [Coding Standard for our Lab AssignmentsPreview the documentView in a new window](https://foothillcollege.instructure.com/courses/3607/files/499103/download?wrap=1)

**before** submitting this assignment. Hand in only one submission.

**Understand the Application**

With so much of our information stored online, having a secure password is a very important practice.  In this lab we’ll write a program that validates a new password following certain rules that will be stated in the program specification below.

**The Program Spec**

Write a program using loops and functions that asks for a password, and then asks the user to confirm the password chosen.  Your solution will verify that both user passwords entered match.  Your program must also validate that the new password follows these rules:

* The password must be at least 8 characters long
* The password must have at least one uppercase and one lowercase letter
* The password must have at least one digit

Re-prompt the user if either 1) the two passwords entered do not match or if  2) the password does not satisfy all the rules stated above.

**Deliverable:  yournameLab3.py**  Your source code solution and a copy of the run pasted into your source submission file.  Be sure to comment out your run so that your **.py** file will still run in the grader test bed.

**Input Error Checking:**  All user input is valid data.  The two user input passwords do need to match each other.  A password does need to be of a minimum length as specified in the spec.  Your program does not need to test for an upper bound limit on the password length.

**Test Run Requirements:**  Submit **one run** that demonstrates at least one example of a password that does not meet each of the rules (expecting a minimum of 5 test cases; 3 demonstrating a password that does not satisfy the rules as well as a pair where the two passwords entered by the user do not match.  Make sure that your run isolates each rule one test case at a time to verify that your program is validating the specific rule.  Also, make sure you include at least one run that contains a valid password verification.

Here are some other requirements:

1. Create constants versus using literals in your source code (i.e. for minimum password length)
2. Include a function that checks whether a password is valid
3. Include a loop that will continue to prompt the user until a valid password is entered

Here is a sample run (remember you’ll need at least five test cases):

…

Enter your password: abcd

Re-enter your password: abcd

That password didn't have the required properties.

Enter your password: gL123456

Re-enter your password: gL123456

That pair of passwords will work.