Laboratory-08

COMSC-122

Fall 2017

Laboratory 08

- We will be rewriting the two functions in Program8-6 to accomplish two goals, which are shown in Part 1 and Part 2 below:
- Part 1:
  - Program 8-6 contains a function called, valid_password(), which determines if the password you type in passes these requirements:
    - Is at least 7 characters long
    - Contains at least one upper case character
    - Contains at least one lower case character
    - Contains at least on numeric digit
  - You are asked to increase the strength of the password's ability to resist hacking by adding three additional requirements to the password:
    - It must now contain at least 8 characters.
    - It must contain at least one non_Alpha_Numeric character.
    - It must not contain any spaces.
- You will use Program 8-7 to drive the module in Program 8-6 that contains the function, valid_password(), in order to test it.
- When you have Part 1 working, call the Instructor to demonstrate.

Laboratory 08

- Part 2:
  - Program 8-6 contains another function called, get_login_name(), which generates a student's ID.
    - It uses the first three letters of the First Name
    - Followed by the first three letters of the Last Name
    - Followed by the last three numbers of the ID Number.
  - You are asked to change get_login_name() so that it conforms to the standards of Los Medanos College, which are the following:
    - It uses the first letter of the first name.
    - Followed by the entire last name of the student.
    - Followed by the last three numbers of the ID Number.
- Using Program8-4, as the function driver, test your amended get_login_name() function to be sure it delivers the Los Medanos version of the Student ID, using your FirstName, LastName, and ID Number.
- When you have Part 2 working, call the Instructor to demonstrate the full operation of your revised Program8-6 which contains the two modified functions.

Program 8-6 (login.py)

```
1  # The get_login_name function accepts a first name,
2  # last name, and ID number as arguments. It returns
3  # a system login name.
4
5  def get_login_name(first, last, idnumber):
6      # Get the first three letters of the first name.
7      # If the name is less than 3 characters, the
8      # slice will return the entire first name.
9      set1 = first[0 : 3]
10
11     # Get the first three letters of the last name.
12     # If the name is less than 3 characters, the
13     # slice will return the entire last name.
14     set2 = last[0 : 3]
15
16     # Get the last three characters of the student ID.
17     # If the ID number is less than 3 characters, the
18     # slice will return the entire ID number.
19     set3 = idnumber[-3 :]
20
21     # Put the sets of characters together.
22     login_name = set1 + set2 + set3
23
24     # Return the login name.
25     return login_name
```

Program 8-6

The Function: get_login_ name()

```
27  # The valid_password function accepts a password as
28  # an argument and returns either true or false to
29  # indicate whether the password is valid. A valid
30  # password must be at least 7 characters in length,
31  # have at least one uppercase letter, one lowercase
32  # letter, and one digit.
33
34  def valid_password(password):
35      # Set the Boolean variables to false.
36      correct_length = False
37      has_uppercase = False
38      has_lowercase = False
39      has_digit = False
40
41      # Begin the validation. Start by testing the
42      # password's length.
43      if len(password) >= 7:
44          correct_length = True
45
46          # Test each character and set the
47          # appropriate flag when a required
48          # character is found.
49          for ch in password:
50              if ch.isupper():
51                  has_uppercase = True
52              if ch.islower():
53                  has_lowercase = True
54              if ch.isdigit():
55                  has_digit = True
```

Program 8-6

The Function: valid_password()

Program 8-6

The Function: valid_password() concluded

```
56
57      # Determine whether all of the requirements
58      # are met. If they are, set is_valid to true.
59      # Otherwise, set is_valid to false.
60      if correct_length and has_uppercase and \
61          has_lowercase and has_digit:
62          is_valid = True
63      else:
64          is_valid = False
65
66      # Return the is_valid variable.
67      return is_valid
```

Program 8-7 (validate_password.py)

```
1  # This program gets a password from the user and
2  # validates it.
3
4  import login
5
6  def main():
7      # Get a password from the user.
8      password = input('Enter your password: ')
9
10     # Validate the password.
11     while not login.valid_password(password):
12         print('That password is not valid.')
13         password = input('Enter your password: ')
14
15     print('That is a valid password.')
16
17 # Call the main function.
18 main()
```

Program Output (with input shown in bold)

```
Enter your password: bozo Enter
That password is not valid.
Enter your password: kangaroo Enter
That password is not valid.
Enter your password: Tiger9 Enter
That password is not valid.
Enter your password: Leopard6 Enter
That is a valid password.
```

Program 8-7

The **main()** Function

Programs 8-6 and 8-7 create a complete Password Checking Application

Program 8-4: The Main Function – generate_login.py

Program 8-4 (generate_login.py)

```
 1  # This program gets the user's first name, last name, and
 2  # student ID number. Using this data it generates a
 3  # system login name.
 4
 5  import login
 6
 7  def main():
 8      # Get the user's first name, last name, and ID number.
 9      first = input('Enter your first name: ')
10      last = input('Enter your last name: ')
11      idnumber = input('Enter your student ID number: ')
12
13      # Get the login name.
14      print('Your system login name is:')
15      print(login.get_login_name(first, last, idnumber))
16
17  # Call the main function.
18  main()
```

Useful String Methods

Table 8-1 Some string testing methods

Method	Description
`isalnum()`	Returns true if the string contains only alphabetic letters or digits and is at least one character in length. Returns false otherwise.
`isalpha()`	Returns true if the string contains only alphabetic letters and is at least one character in length. Returns false otherwise.
`isdigit()`	Returns true if the string contains only numeric digits and is at least one character in length. Returns false otherwise.
`islower()`	Returns true if all of the alphabetic letters in the string are lowercase, and the string contains at least one alphabetic letter. Returns false otherwise.
`isspace()`	Returns true if the string contains only whitespace characters and is at least one character in length. Returns false otherwise. (Whitespace characters are spaces, newlines (`\n`), and tabs (`\t`).
`isupper()`	Returns true if all of the alphabetic letters in the string are uppercase, and the string contains at least one alphabetic letter. Returns false otherwise.