File Edit Format Run Options Window Help

```
#Code by Rajen chhetry -Python Training certificate course
#Project 2 : password manager

"""Objective: To develop a custom password manager using Python
   Domain:  Security

   Steps to perform:

    class BasePasswordManager
      members
        old_passwords: is a list that holds all of the user's past passwords.
        The last item of the list is the user's current password.

      methods
        get_password method that returns the current password as a string.
        is_correct method that receives a string and returns a boolean True or False depending on whether the string is equal to the current password or not.

    class PasswordManager
    This class inherits from BasePasswordManage
     methods
        set_password method that sets the user's password.
             Password change is successful only if:
                    - Security level of the new password is greater.
                    - Length of new password is minimum 6
        get_level method that returns the security level of the current password.
                 It can also check and return the security level of a new password passed as a string.
                   -level 0 - password consists of alphabets or numbers only.
                   -level 1 - Alphanumeric passwords.
                   -level 2 - Alphanumeric passwords with special characters. """
import re
class BasePasswordManager:

    def old_passwords(self):
        old_password = ['012345', '2134siph', '2134sip$']
        self.old_password = old_password[-1]
        return self.old_password

    def get_password(self):
        current_password = self.old_password
        self.current_password = current_password
        return "Current password is " + self.current_password
```

```
    def is_correct(self, password=input('please type in your password: ')):
        self.password = password
        print("New passsword is the same as the current password:",self.password == self.current_password)
        return self.password

class PasswordManager(BasePasswordManager):

    def get_level(self):
        self.security_level = 0
        check_alphabat = False
        check_number = False
        if self.password.isdigit():
            self.security_level = 0
            print('Security level is',self.security_level,' :WEAK')
            print('Password consists of Digits only')

        elif self.password.isalpha():
            self.security_level = 0 # For Security_Level 0
            print('Security level is',self.security_level,' :WEAK')
            print('Password consists of Alphabets only')

        elif check_alphabet == False and check_number == False and (bool(re.match('^[a-zA-Z0-9]*$', self.password)) == True):
            for i in self.password:
                if i.isalpha():
                    check_alphabet = True
            for j in self.password:
                if j.isnumeric():
                    check_number = True
                if check_alphabet == True and check_number == True:
                    self.security_level = 1 # For Security_Level 1
                    print('Security Level is', self.security_level, ': MODERATE')
                    print('Password is alphanumeric with NO special characters')
```

```
            elif check_alphabet == False and check_number == False:
                for i in self.password:
                    if i.isalpha():
                        check_alphabet = True
                for j in self.password:
                    if j.isnumeric():
                        check_number = True
                if check_alphabet == True and check_number == True and (bool(re.match('^[a-zA-Z0-9]*$', self.password)) == False):
                    self.security_level = 2 # For Security_Level 2
                    print('Security level is', self.security_level, ':STRONG')
                    print('Password is alphanumeric with special characters')
                else:
                    self.security_level = 1

                    print('Security level is', self.security_level, ': MODERATE')
                    print('Password contains special characters with either numbers or alphabets only')

    def set_password(self):
        if len(self.password) <8:
            print("New Password must have 8 characters or More")
            print("Password Change : UNSUCCESSFUL")
        elif self.security_level <2:
            print("New password must contain at least 1 special character with numbers and alphabets")
            print("Password Change : UNSUCCESSFUL")
        elif self.password == self.current_password:
            print("Password Change: No Changes Detected")
        else:
            print("Password Change:SUCCEDDFUL")
            print()

current = PasswordManager()
current.old_passwords()
current.get_password()
current.is_correct()
current.get_level()
current.set_password()
```