ABW505D DATA PROGRAMMING AND PREDICTIVE ANALYTICS FOR BUSINESS

TS. DR. KHAW KHAI WAH

Name: Kareem Mustafa Mohamed Eldemerdash

Email: kareemmostafa15@student.usm.my

Matric No: P-EMO112/22

Project: password code generator

The Random Password Generator

Introduction

In this report we will develop a program using python generates a powerful random password depends on the length that entered by user.

We will use RANDOM MOUDULE and STRING MOUDULE.

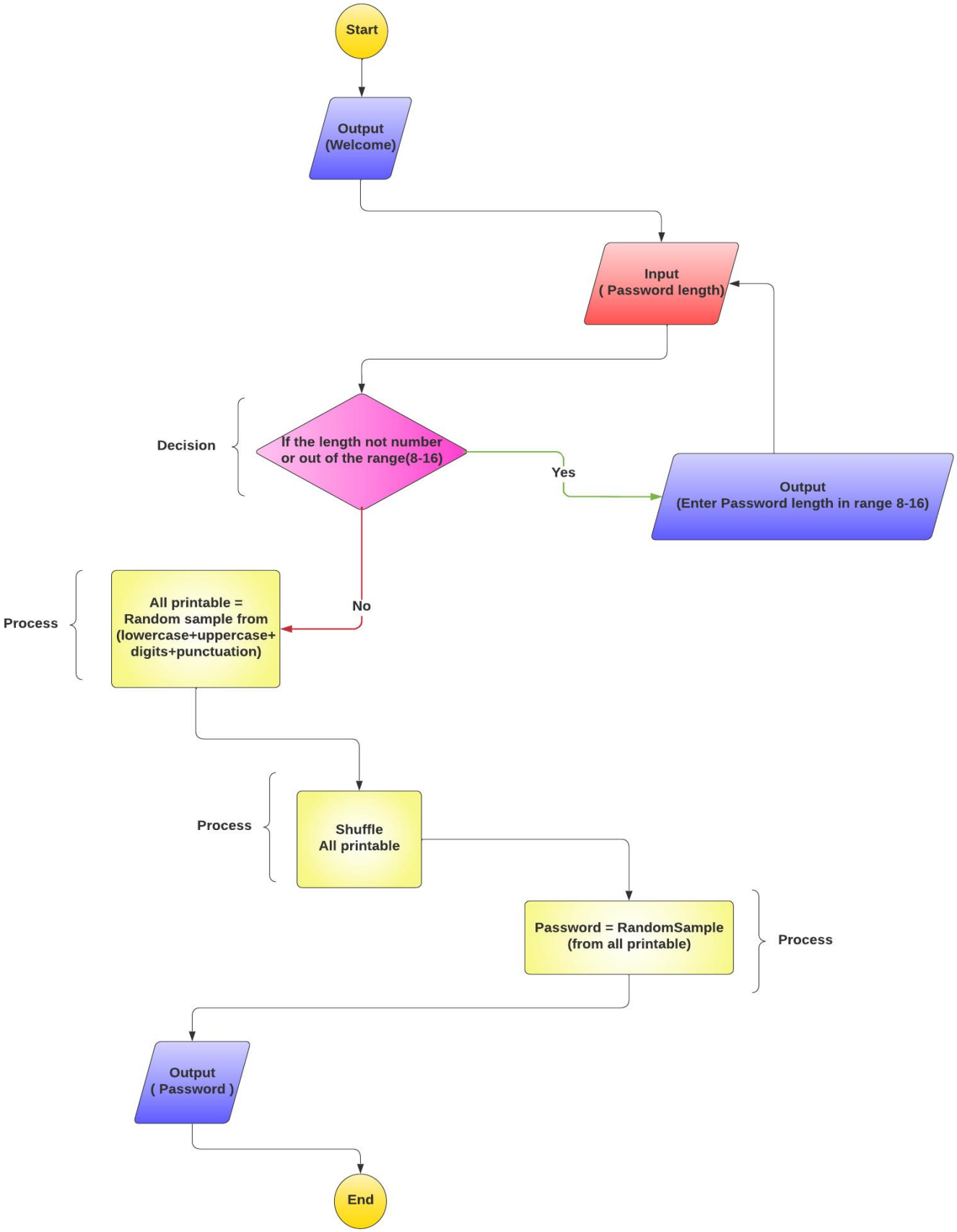
That will explain in:

- Algorithm development
- Pseudocode
- Flowcharts
- Coding
- Conclusion

Algorithm development

- Start
- Saying Welcome to the user.
- Enter the length of the password from the user in range (8-16).
- If the length is not number or out of the range, print and correcting message with one more trial to generate another password.
- If the length in range (8 -16) breaks the loop
- Make a random list from **all characters can be printed**.
- Shuffle the list.
- develop the (**Password**) from **random sample** from this list with a sample size that the same of the length the user entered before.
- End

Flowcharts:


Start
Output (Welcome)
Input (Password length)
Decision
If the length not number or out of the range(8-16)
Yes
Output (Enter Password length in range 8-16)
No
Process
All printable = Random sample from (lowercase+uppercase+ digits+punctuation)
Process
Shuffle All printable
Password = RandomSample (from all printable)
Process
Output (Password)
End

Pseudocode

Program starts.

Print ‘Welcome, Your Random Password Generator! '.

Ask user to enter a Password Length

While True

If length not number or less than 8 or more than 16 then

    ‘password length should be more than 8 and less than 16’

    Input length

Else

    Change the type of the input to integer

    Break while loop

Create a random list from lowercase, uppercase, digits and punctuation.

Shuffle the list

Create a password from Random sample from the list with length of input.

Print 'your password is (Password)

Coding

Random Password Generator

n [5]:

```
import random
import string

#greeting the user
print('Welcome, Your Random Password Generator! ')
#input the lenght of the password
password_lenth = input('enter the lenth(8-16) of your password please!: ')


#while loop for checking about the input
while True:
    #if it is not number in the available range or not that return a correction message and one more traial
    if not(password_lenth.isnumeric()) or not (8<= int(password_lenth) <=16):
        print ('password lenth shoud be more than 8 and less than 16 ')
        password_lenth = input('enter the lenth of your password please!: ')
    #change the input from string to integr and break the loop
    else:
        password_lenth=int(password_lenth)
        break



#creating a list from all caracters can be printed
all_printable = random.sample(string.ascii_lowercase,(password_lenth+3)//4) + random.sample(string.ascii_uppercase,(password_lent


#shuffle the list
random.shuffle(all_printable)


#random sample from the the list to create the password
password =''.join(random.sample(all_printable,password_lenth))


#output the password
print('your password is:▼▼▼\n>>>>> ',password)
```

```
Welcome, Your Random Password Generator!
enter the lenth(8-16) of your password please!: 8
your password is:▼▼▼
>>>>>  5^.kF8Nz
```

Coding

```
import random
import string
#greeting the user
print('Welcome, Your Random Password Generator! ')
#input the lenght of the password
password_lenth = input('enter the lenth(8-16) of your password please!: ')
#while loop for checking about the input
while True:
#if it is not number in the available range or not that return a correction message a
nd one more traial
  if not(password_lenth.isnumeric()) or not (8<= int(password_lenth) <=16):
    print ('password lenth shoud be more than 8 and less than 16 ')
    password_lenth = input('enter the lenth of your password please!: ')
#change the input from string to integr and break the loop
  else:
    password_lenth=int(password_lenth)
    break
#creating a list from all caracters can be printed
all_printable = random.sample(string.ascii_lowercase,(password_lenth+3)//4) + random.
sample(string.ascii_uppercase,(password_lenth+3)//4) + random.sample(string.digits,(p
assword_lenth+3)//4) + random.sample(string.punctuation,(password_lenth+3)//4)
#shuffle the list
random.shuffle(all_printable)
#random sample from the the list to create the password
password =''.join(random.sample(all_printable,password_lenth))
#output the password
print('your password is:▼▼▼\n>>>>> ',password)
```

Conclusion

- When we have to make a decision, we can use (IF) of choosing between more than one alternative.

- When we need to repeat any function we can use (While) and use (break)
  For breaking this loop.

- Random module helps us to making a random selection from a set of data with the function of **random.sample(the data set , the sample size).**

- String module helps us to classify any type of characters in one of this group

  string.ascii_letters

  The concatenation of the **ascii_lowercase** and **ascii_uppercase** constants described below. This value is not locale-dependent.

  string.ascii_lowercase

  The lowercase letters 'abcdefghijklmnopqrstuvwxyz'.

  This value is not locale-dependent and will not change.

  string.ascii_uppercase

  The uppercase letters 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'.

  This value is not locale-dependent and will not change.

  string.digits

  The string '0123456789'.

  string.hexdigits

  The string '0123456789abcdefABCDEF'.

  string.octdigits

  The string '01234567'.

  string.punctuation

  String of ASCII characters which are considered punctuation characters in the C locale: !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~.

string.printable

String of ASCII characters which are considered printable.

This is a combination of **digits, ascii_letters, punctuation,** and **whitespace**.

string.whitespace

A string containing all ASCII characters that are considered whitespace. This includes the characters **space**, **tab**, **linefeed**, **return**, **formfeed**, and **vertical tab**

References:

random — Generate pseudo-random numbers — Python 3.10.4 documentation

string — Common string operations — Python 3.10.4 documentation