# TIE2030

# Programming Methodology with Python

# Laboratory Exercise (LAB-II)

**Date submission due: Thursday 14 October 2021**

**Grading: There are only two questions in this exercise. Your programs will be graded out of 50 marks and the final weight of this assignment is 25%. The guidelines explained in the template will also carry marks. So please adhere to the guidelines.**

**Useful Concepts -** This exercise mainly uses program loops & control. For processing and handling any data, you may consider using data structures in Python like, lists, tuples, arrays, dictionaries, etc.

## Question 1: (25 marks) Password verifier

In this program you will develop a password verifier. Follow the rules to generate a password.

- Your password must be at least 10 characters in length
- There must be **at least** one numerical value, one uppercase English alphabet, one lowercase English alphabet, and one symbol
- There must not be 3 consecutive identical chars
- Limit your passwords to the following symbol listing: @ # $ % ^ & * + _ !? ~ > < ; , . / \ | (Of course, you may use more but this is the recommended set)
- If the entered password is correct, display the number of textual, numerical and special characters as shown below in the second sample output
- **First draw a high-level flowchart of your code – Your code must follow this design and will be verified.**

**Give the user 5 trials** to input a password that satisfies your rules. If he/she fails continuously, then you may quit with a message. See the sample outputs. You must display as prescribed here.

In your report, paste the screenshot of your code and the screenshots of the runs that show the following cases:

1. The user inputs invalid passwords for 5 times.
2. The user inputs a valid password in the first trial.
3. The user input invalid passwords in the first 4 trials and a valid password in the last trial.

**Note - We will test many more cases, not limited to the above 3, so you should test your code thoroughly to make sure it gives the correct outputs.**

**Sample Output 1:**

```
The rules for creating a password here are:
 The minimum length should be of 10 characters including
 at least one numerical value, one uppercase English alphabet,
 one lowercase English alphabet, and one symbol
 and no 3 consecutive chars

Enter a password: 555&$Dea28
Breakdown of the password
Length of your password is: 10
Plese don't put 3 consecutive identical chars.
Invalid password. Try again please.

Want to set your password? Type 1 for yes and 0 for no: 1
Enter a password: 786T&&&q12
Breakdown of the password
Length of your password is: 10
Plese don't put 3 consecutive identical chars.
Invalid password. Try again please.

Want to set your password? Type 1 for yes and 0 for no: 1
Enter a password: uyrtsnhtg829
Breakdown of the password
Length of your password is: 12
Invalid password. Try again please.

Want to set your password? Type 1 for yes and 0 for no: 1
Enter a password: uyUY5
Breakdown of the password
Length of your password is: 5
Invalid password. Try again please.

Want to set your password? Type 1 for yes and 0 for no: 1
Enter a password: 8976tRR218
Breakdown of the password
Length of your password is: 10
Invalid password. Try again please.
You have exceeded the max number of chances! Bye.

The rules for creating a password here are:
 The minimum length should be of 10 characters including
 at least one numerical value, one uppercase English alphabet,
 one lowercase English alphabet, and one symbol
 and no 3 consecutive chars
```

5th time

**Sample Output 2:**

```
The rules for creating a password here are:
 The minimum length should be of 10 characters including
 at least one numerical value, one uppercase English alphabet,
 one lowercase English alphabet, and one symbol
 and no 3 consecutive chars

Enter a password: YT56%$dEnt
Breakdown of the password
Length of your password is: 10
Valid password.
There are 2 digits in your password
There are 6 texture characters in youur password
There are 2 symbols in your password
Valid! Great!
```

## Question 2: (25 marks) Vowel Character Checker

In this program, you will develop a Vowel Character Checker for given statements. You may store the statements as strings in a list, if you wish. Report the following **for every statement**. Sample output is also given.

- Display the statement
- Total number of characters in the statement
- Total number of words in your statement
- Frequency of occurrence of each vowel a, e, i, o, u (**case-insensitive**)
- Total number of characters other than alphabets (**do not include space character**)

Given test statements. Use exactly as they are given.

1. PythOn was conceived in the late 1980s by GuIdo van Rossum at CEntrum WIskunde & InformaticA
2. Python uses the words and, or, not for its boOlean opErators rather than the Symbolic &&, ||, ! used in Java and C

From the output given below, you may observe that this program recognizes only lower-case letters. **In your version of the program you need to count upper case vowels too** (your vowel character checker is case-insensitive). Note that the statement in the sample output is different from the test statements given above.

**Sample Output (**& is considered a character and a word) **:**

```
Total number of statements: 2
Your statement:
Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica
Total number of characters in the given statement: 92
Total number of words in the statement: 16
The frequency of vowel a is 6
The frequency of vowel e is 6
The frequency of vowel i is 5
The frequency of vowel o is 5
The frequency of vowel u is 4
Total number of characters other than alphabets: 5

Your statement:
<your second statement>
...
```