Assessment 2: Programming Exercise

QC1003: Programming 1
30% weighting

Password Generator

You design and write a Python program that generates secure passwords. The key requirements include:

1. Users have the option to either create their own password or have the program generate one randomly.
2. If users create their own password, they must confirm it by entering the same password twice for it to be accepted.
3. The password must meet certain criteria: a minimum length, inclusion of uppercase letters, numbers, special characters, and lowercase letters.
4. The password must be unique and not previously used.
5. Your password generator must be of good quality. Show how you demonstrate this.

What you need to submit:

A zipped folder containing:

1. Your source code: i.e. Python code.
2. A 1 page (max 2 pages) pdf document where you discuss your code design and issues you have identified and resolved. In this pdf document you discuss mainly requirement 5 and any other additional aspects of your submissions.

Submission Instructions

1. You must submit your code via MyAberdeen before 23.59 on 06/11/2024. Make sure your solution is ready to run before then.
2. Any changes to the submission instructions will be communicated to you via MyAberdeen.

Marking criteria

For reference: This is the official University of Aberdeen marking scale:

[Common Grading Scale (CGS) | Students | The University of Aberdeen (abdn.ac.uk)]

A Excellent	Implements working code for all 1 to 5 requirements. All code is commented according to the Python Coding Style Your program runs efficiently, uses fewer computing resources. Demonstrate how you achieve this. Program accounts for users creating incorrect passwords, and hints at what the error(s) were, and allows further attempts.

	Program enables users to retrieve their password, example: Forgot Your Password? Submitted work demonstrates additional feature(s) beyond what is asked for in the 5 requirements.
B Very Good	Implements working code for all 1 to 5 requirements. All code is commented according to the Python Coding Style Your program runs efficiently, uses fewer computing resources. Demonstrate in detail how you achieve this. Program accounts for users creating incorrect passwords, and hints at what the error(s) were, and allows further attempts.
C Good	Implements working code for all 4 requirements. All code is commented according to the Python Coding Style Your program runs efficiently, uses fewer computing resources. Demonstrate how you achieve this. Program accounts for users creating incorrect passwords, and hints at what the error(s) were, and allows further attempts.
D Pass	Partial implementation of requirements 1 to 2 Some of the code is commented.
E Marginal Fail	Minimal implementation of requirements 1 to 2. Some of the code is commented. Code has errors.
F Fail	Program is minimal and has errors. Poor design.