**Password Generator Documentation**

**1. Overview:**

This password generator is implemented using object-oriented programming (OOP) concepts in Python. It allows the user to generate multiple passwords of varying lengths, with each password containing a combination of lowercase letters, numbers, and uppercase letters.

**2. PasswordGenerator Class:**

* **Attributes:**
  + **alphabet**: A string containing all lowercase letters of the English alphabet.
* **Methods:**
  + **\_\_init\_\_(self)**: Initializes the **PasswordGenerator** object by setting the **alphabet** attribute to the string of lowercase letters.
  + **generate\_password(self, length)**: Generates a password of the specified length. The password is created by randomly selecting lowercase letters from the **alphabet**, replacing some characters with numbers using **\_replace\_with\_number()** method, and replacing some characters with uppercase letters using **\_replace\_with\_uppercase\_letter()** method.
  + **\_replace\_with\_number(self, password)**: Replaces some characters in the password string with random digits (0-9).
  + **\_replace\_with\_uppercase\_letter(self, password)**: Replaces some characters in the password string with uppercase letters.

**3. Main Function:**

* **User Interaction:**
  + Prompts the user for the number of passwords to generate and their respective lengths.
  + Ensures that the minimum length of each password is 3.
  + Creates an instance of the **PasswordGenerator** class.
  + Calls the **generate\_password()** method for each password length and prints the generated passwords.

**4. Design Choices:**

* **Object-Oriented Approach:**
  + The password generator is implemented as a class (**PasswordGenerator**) to encapsulate related functionality and attributes.
  + This approach enhances modularity, reusability, and readability of the code.
* **Randomization:**
  + Randomization is utilized to generate passwords with varying characters, making them more secure.
  + Random characters are selected from the alphabet, and some characters are replaced with numbers and uppercase letters to add complexity to the passwords.

**5. Conclusion:**

This password generator provides a convenient way to generate multiple passwords with varying lengths, ensuring a balance between security and usability. By leveraging object-oriented design principles and randomization techniques, the generator produces passwords suitable for various applications while offering flexibility and customization options to the user.