Python Password Generator

Summary:

The Password Generator script has a few issues related to user input validation and security concerns.

Steps to Reproduce:

1. Run the program.
2. Input various values for the number of passwords and their lengths.

Expected Behavior:

1. The program should generate the specified number of passwords, each with a length greater than 3.
2. Passwords should be generated securely and not rely solely on the `**random.sample**` function for cryptographic strength.
3. The program should provide clear feedback if an invalid password length is entered.

Observed Behavior:

1. If a password length less than or equal to 3 is entered, the program displays an error message, but the logic is flawed, as it still appends the invalid length to the `**passwords_length**` list.
2. The password generation method may not be cryptographically secure, and its reliance on `**random.sample**` may lead to potential security vulnerabilities.
3. The program lacks proper user input validation for non-integer inputs or invalid characters, which could result in unexpected behavior.

Environment:

- Operating System: [**Windows**]
- Python Version: [**Python 3.10.7**]

Additional Notes:

- Refactor the input validation to ensure that only valid password lengths are appended to the `**passwords_length**` list.
- Consider using a more secure method for password generation, such as the `**secrets**` module or a dedicated password hashing library.
- Implement additional input validation to handle non-integer inputs or invalid characters during the password length input.
- Provide clearer feedback to the user about the password generation process and any potential security concerns.