

User: Help me complete this coding assignment. This assignment will help me: Determine which built-in modules can be used to address the prompt (like math, random, string, urllib), Effectively use functions and objects from various built-in modules, and Practice python fundamentals and best practices. This is what the code should do: Create a password generator, you can choose if you want it as an object or a function. Regardless, the generator should ask the format for password type, either memorable (covered in part a) or random (covered in part b).

A memorable password is generated by selecting n random words, adding a random 1 digit number to each word, and then concatenating them using a hyphen (-). A memorable password should include arguments specifying the number of words and the available cases for those words.

A random password is generated by selecting n random characters from among a variety of different character types: lower case, upper case, numbers, and or punctuation symbols. A random password should include arguments specifying the length of the password, if they want punctuation symbols included, and any characters that aren't allowed in the password.

Add some additional functionality to your password generator. When a password is generated the password as well as the time it was created (day, date, and time) should be appended to a file titled "Generated_Passwords.txt". Both types of passwords (memorable or random) should have their own "Generated_Passwords.txt" files, contained in different directories titled "Memorable" and "Random" respectively. If one of those directories doesn't exist the code should make them. Confirm that this works by generating 1000 passwords with the type chosen randomly between memorable and random.

GitHub Copilot: I'll open assignment1.py to see what's already implemented, then add a complete password generator and tests.

Read [(file:///Users/kyal0513/Desktop/B211%20assignment%20fun/assignment1.py)]

I'll create a short plan of implementation steps before editing the file.

Created 3 todos

Now I'll add assignment1.py with the password generator implementation and a `--demo` runner. Now I'll write assignment1.py with the generator, logging, and a `--demo` runner. I'll add the file to the workspace.

Using "Apply Patch"

...

...

I'll update the todo list to mark implementation and demo runner complete.

Starting: *Run demo* (3/3)

I'll run the script's demo to generate 1000 passwords and create the output files.

Ran terminal command: `python3 '/Users/kyal0513/Desktop/B211 assignment fun/assignment1.py' --demo`

Now I'll mark the demo run as completed in the todo list.

Completed: *Run demo* (3/3)

Done — I added assignment1.py, implemented the generator and logging, and ran the demo to create 1000 entries. Files created:

- Generated_Passwords.txt
- Generated_Passwords.txt

Next steps: run interactive mode with ``python3 'assignment1.py'`` or inspect the generated files. Do you want me to open one of the generated files?

Made changes.