Cows and Bulls

- Create a program that will play the "cows and bulls" game with the user.
- The game works like this:
- Randomly generate a 4-digit number.
- Ask the user to guess a 4-digit number.

Cows and Bulls - Continued

- For every digit that the user guessed correctly in the correct place, they have a "cow".
- For every digit the user guessed correctly in the wrong place is a "bull".

Cows and Bulls - Continued

- Every time the user makes a guess, tell them how many "cows" and "bulls" they have.
- Once the user guesses the correct number, the game is over.
- Keep track of the number of guesses the user makes throughout the game and tell the user at the end.

Cows and Bulls - Continued

- Say the number generated by the computer is 1038.
- An example interaction could look like this: Welcome to the Cows and Bulls Game! Enter a number: »> 1234 2 cows, 0 bulls »> 1256 1 cow, 1 bull ... Until the user guesses the number.

Cows and Bulls Game - Instructions

- Ensure the game would work for any number of digits
- Game is treated as complete if cows count is equal to number of digits
- After a game is complete, repeat the next game as long as user wants to play

Secret Number

- Generate Secret Number using random number generator module random
- Ensure secret number has no duplicate digits

Possible Methods for Cows and Bulls Class

- Initialization (__init__()) to maintain properties or variables required for the cows and bulls object
- _count_bulls(), to count number of bulls (ensure to subtract number of cows from the bulls count)
- _count_cows(), to count number of cows

Possible Methods for Cows and Bulls Class - Continued

- check_cows_bulls(), to call bull and cow count methods (ensure to call cows count before bull count)
- finished() to check whether game is complete
- score() to know the current score i.e., number of cows and number of bulls