## ELDO-HUB - DATA SCIENCE – Python Programming

**Class Assignment – Topic 3 Iteration**

## Challenge 1: Digit Frequency Counter

**Goal:** Practice for **loops**, string handling, and dictionaries to count how often each digit appears.

### 🔍 Description

1. Prompt the user:

num\_str = input("Enter a positive integer: ")

1. **Validate** input is all digits (str.isdigit())
2. Initialize a **dictionary** counts = {}.
3. **Iterate** over each character in the string (for ch in num\_str:) :
   * Convert ch to an integer if desired, or keep as string key.
   * Use counts[ch] = counts.get(ch, 0) + 1.
4. After the loop, **print** each digit and its frequency in ascending order:

python

CopyEdit

for digit in sorted(counts):

print(f"Digit {digit}: {counts[digit]} time(s)")

1. **Example Run:**

css

CopyEdit

Enter a positive integer: 12234331

Digit 1: 3 time(s)

Digit 2: 2 time(s)

Digit 3: 3 time(s)

Digit 4: 1 time(s)

## Challenge 2: Guess‑the‑Number Game with Limited Attempts

**Goal:** Combine while **loops**, **branching**, and **input/output** to create an interactive game.

### 🔍 Description

1. **Import** the built‑in random module.
2. **Generate** a random integer between 1 and 50:

import random

secret = random.randint(1, 50)

1. Set a **counter** attempts = 0 and a **maximum** max\_attempts = 5.
2. Use a while **loop** (while attempts < max\_attempts:) to allow up to 5 guesses:
   * Prompt:

guess = int(input(f"Attempt {attempts+1}/{max\_attempts}: Your guess? "))

* + **Increment** attempts += 1.
  + **Branch**:
    - If guess == secret:

print("🎉 Correct! You guessed the number in", attempts, "tries.")

break

* + - Elif guess < secret:

print("Too low! Try again.")

* + - Else:

print("Too high! Try again.")

1. After the loop, if the user **didn’t guess** correctly:

else:

print(" Out of attempts. The number was", secret)

1. **Example Run:**

Attempt 1/5: Your guess? 25

Too low! Try again.

...

 Correct! You guessed the number in 4 tries.

### References

1. Python string methods: str.isdigit()
2. For loop syntax: for x in iterable:
3. While loop overview: “Python’s while loop repeats as long as condition is true”
4. Branching examples in loops: GeeksforGeeks
5. Dictionary .get() for counting
6. random.randint() documentation
7. Sorting dictionary keys: Python sorted() built-in
8. F‑strings for output formatting
9. Break and else clause on loops
10. Input conversion: int(input())