>>>> Day 5:

Topics Covered:

1. Multiplication Table of any number
2. Fibonacci Series up to N terms
3. Check if a number is prime
4. Find all prime numbers in a given range

1️⃣ Multiplication Table

- **Purpose:** Print multiplication table of any number.
- **Core Concept:** `for` loop and arithmetic operation (`*`).
- **Syntax Example:**

```
num = int(input("Enter a number: "))
for i in range(1, 11):
    print(f"{num} x {i} = {num * i}")
```

2️⃣ Fibonacci Series

- **Purpose:** Print Fibonacci series up to N terms.
- **Core Concepts:**
    - Loops for iteration
    - Variable swapping
- **Syntax Example:**

```
n = int(input("Enter number of terms: "))
a, b = 0, 1
print(a, b, end=" ")
for _ in range(2, n):
    c = a + b
    print(c, end=" ")
    a, b = b, c
```

3️⃣ Check Prime Number

- **Purpose:** Check if a given number is prime.
- **Core Concepts:**
  - Conditional statements (`if-else`)
  - Modulo operator `%` to check divisibility
  - Loop optimization using `sqrt(n)`
- **Syntax Example:**

```
num = int(input("Enter a number: "))
if num <= 1:
    print("Not prime")
else:
    is_prime = True
    for i in range(2, int(num**0.5)+1):
        if num % i == 0:
            is_prime = False
            break
    print("Prime" if is_prime else "Not prime")
```

4️⃣ Prime Numbers in a Range

- **Purpose:** Print all prime numbers between a given range.

- **Core Concepts:**
  - Loops for iterating through range
  - Nested loops for checking primes
- **Syntax Example:**

```
start = int(input("Start: "))
end = int(input("End: "))
for num in range(start, end+1):
    if num > 1:
        is_prime = True
        for i in range(2, int(num**0.5)+1):
            if num % i == 0:
                is_prime = False
                break
        if is_prime:
            print(num, end=" ")
```

✅ Core Concepts Summary

Concept	Usage
Loops (`for`)	Repeated execution, iterating ranges, generating tables/series
Conditional Statements	`if-else` for logic (prime check, input validation)
Arithmetic Operations	`+`, `*`, `%` for calculations and divisibility checks
User Input	`input()` + `int()` for numeric input
Optimization	Checking primes up to `sqrt(n)` for efficiency
Variable Swapping	Used in Fibonacci series (`a, b = b, a+b`)