THADOMAL SHAHANI ENGINEERING COLLEGE

DEPARTMENT OF INFORMATION TECHNOLOGY

Roll no:I-12 **7/2/2025**

3. Control Structure:LO2

Aim:

1)Write a Python program to print a triangle pattern (give any), emphasizing the transition from C to Python syntax

Theory:

i. Loops (for, while) control repeated execution.
ii. The range() function generates a sequence of numbers.
iii. Nested loops create structured patterns.
iv. print() with string multiplication ('*' * i) simplifies repeated characters.
v. Python uses indentation instead of {} like in C.
vi. Variables store user input dynamically.
vii. Pattern logic requires proper loop control

Program:

```
rows = int(input("Enter

number of rows: ")) for i in

range(1, rows + 1):

    print("*" * i)
```

Output :

```
Enter number of rows:5

*

**

***

****
```

*****

Conclusion:

In conclusion, Python simplifies pattern printing by using concise syntax, eliminating the need for semicolons and curly braces, and relying on indentation for code structure. This makes the code more readable and easier to write compared to C.

Aim:

2) Design a Python program to compute the factorial of a given integer N.

Theory:

1. Factorial (N!) is calculated as 1 × 2 × ... × N.
2. Loops (for or while) accumulate the product.
3. fact variable stores the computed factorial.
4. Factorial of 0 is always 1.
5. range(1, N+1) iterates from 1 to N.
6. int(input()) ensures user input is numeric.
7. The logic prevents errors for negative numbers.

Program:

```
num =
int(input("Enter a
number: ")) fact = 1
for i in
   range(1,
   num + 1):
   fact *= i
```

```
print(f"Factorial of {num} is {fact}")
```

Output :

Enter a number: 5

Factorial of 5 is 120

Conclusion :

In conclusion, the Python program to compute the factorial of a number effectively demonstrates the use of loops or recursion to multiply integers from 1 to NNN. It highlights how simple logic and basic arithmetic operations can be applied to solve mathematical problems in Python. This is a fundamental concept, useful for understanding iteration and recursion.