

# AI Assisted Coding

## Assignment-1.2

**Name: Mohammed Rahil Raza**

**Ht.no: 2303A51433**

**Batch. No: 21**

---

### **1Q) Task 1: AI-Generated Logic Without Modularization (Factorial without Functions)**

#### **Prompt :**

write a python code to calculate the factorial of a number without using any functions and input should be taken from user.

#### **Code:**

```
1  '''write a python code to calculate the factorial of a number without
2  using any functions and input should be taken from user.'''
3  number = int(input("Enter a number to calculate its factorial: "))
4  factorial = 1
5  for i in range(1, number + 1):
6      factorial *= i
7  print(f"The factorial of {number} is {factorial}")
8
9
```

#### **Output:**

```
● PS C:\Users\rahil> python "C:\Users\rahil\OneDrive\Documents\assignment 1.2.py"
Enter a number to calculate its factorial: 5
The factorial of 5 is 120
PS C:\Users\rahil>
```

#### **Justification:**

By doing this task, the factorial of a number is calculated without using any functions.

The program takes input from the user and uses a single for loop to compute the factorial.

All steps are written directly in the main program using iterative logic.

## 2Q) Task 2: AI Code Optimization & Cleanup (Improving Efficiency)

### Prompt used:

optimize the above code by adding comments, readability and with improved logic.

### Code:

```
1
2 '''optimize the above code by adding comments, readability andwithimproved logic.'''
3 # Taking input from the user
4 number = int(input("Enter a number to calculate its factorial: "))
5 # Initializing factorial variable to 1
6 factorial = 1
7 # Looping from 1 to the entered number (inclusive)
8 for i in range(1, number + 1):
9     factorial *= i # Multiplying current value of factorial with i
10 # Displaying the result
11 print(f"The factorial of {number} is {factorial}")
12
```

### Output:

```
PS C:\Users\rahil> python "C:\Users\rahil\OneDrive\Documents\assignment 1.2.py"
Enter a number to calculate its factorial: 4
The factorial of 4 is 24
❖PS C:\Users\rahil> 
```

### Justification:

By doing this task, the factorial program is optimized for better readability and understanding.

Meaningful comments and clear variable usage are added while keeping the logic simple and efficient.

The program still uses a single loop and user input, with all steps written in the main program.

## 3Q) Task 3: Modular Design Using AI Assistance (Factorial with Functions)

## Prompt used:

- write a python code to calculate the factorial of a number using meaningful user defined functions and taking the input from the user.
- if user gives negative numbers then this code should handle that case as well.

## Code:

```
C:\> Users > rahil > OneDrive > Documents > assignment1.2.py > ...
1  ''' write a python code to calculate the factorial of a number using meaningful user defined functions and taking the input from the user'''
2
3  def calculate_factorial(num):
4      """Function to calculate the factorial of a given number."""
5      factorial = 1
6      for i in range(1, num + 1):
7          factorial *= i
8      return factorial
9  # Taking input from the user
10 number = int(input("Enter a number to calculate its factorial: "))
11 # Calling the function and storing the result
12 result = calculate_factorial(number)
13 # Displaying the result
14 print(f"The factorial of {number} is {result}")
15
16 '''if user gives negative numbers then this code should handle that
17 case as well'''
18 def calculate_factorial(num):
19     if num < 0:
20         return "Factorial is not defined for negative numbers."
21
22     factorial = 1
23     for i in range(1, num + 1):
24         factorial *= i
25     return factorial
26
27
28 number = int(input("Enter a number to calculate its factorial: "))
29 result = calculate_factorial(number)
30
31 print(f"The factorial of {number} is {result}")
32
```

## Output:

```
Enter a number to calculate its factorial: -4
The factorial of -4 is Factorial is not defined for negative numbers.
```

## Justification:

By doing this task, the factorial logic is implemented using a meaningful user-defined function.

The program takes input from the user and handles negative numbers with proper validation.

Modular design improves code readability, reusability, and error handling.

#### **4Q) Task 4: Comparative Analysis – Procedural vs Modular AI Code (With vs Without Functions)**

<b>Criteria</b>	<b>Without Functions (Procedural Code)</b>	<b>With Functions (Modular Code)</b>
<b>Logic Clarity</b>	Logic is written linearly in one place, which is easy to follow for small programs but becomes cluttered as code grows.	Logic is clearly separated into meaningful functions, making the flow easier to understand.
<b>Reusability</b>	Code cannot be reused easily; the same logic must be rewritten if needed again.	Functions can be reused multiple times across the program or in other programs.
<b>Debugging Ease</b>	Debugging is harder because all logic is mixed in the main program.	Easier to debug since errors can be isolated within specific functions.
<b>Suitability for Large Projects</b>	Not suitable for large projects due to poor structure and maintainability.	Highly suitable for large projects because of clean structure and modular design.
<b>AI Dependency Risk</b>	Higher risk, as AI-generated logic may be copied blindly without understanding due to lack of structure.	Lower risk, as modular design encourages understanding of logic, inputs, and outputs.

#### **Conclusion:**

Procedural code is useful for small, simple tasks and beginners, while

modular code using functions is better suited for scalable, maintainable, and professional software development.

## 5Q) Task 5: AI-Generated Iterative vs Recursive Thinking

- An iterative version of the logic

### Prompt used:

write a python code to calculate the factorial of a number using iterative approach with meaningful user defined functions and taking the input from the user.

### Code:

```
1  '''write a python code to calculate the factorial of a number using iterative
2  approach with meaningful user defined functions and taking the input from the user. '''
3  def calculate_factorial_iterative(number):
4      """
5      Calculate the factorial of a given number using an iterative approach.
6
7      Parameters:
8      number (int): The number to calculate the factorial for.
9
10     Returns:
11     int: The factorial of the number
12     """
13     if number < 0:
14         return "Factorial is not defined for negative numbers."
15
16     factorial = 1
17     for i in range(1, number + 1):
18         fac (variable) factorial: int
19     return factorial
20
21
22 # Get number input from user
23 num = int(input("Enter a number to calculate its factorial: "))
24
25 # Calculate and display the factorial
26 result = calculate_factorial_iterative(num)
27 print(result)
28
```

### Output:

```
PS C:\Users\rahil> -2
-2
PS C:\Users\rahil>
```

- **A recursive version of the same logic**

### **Prompt used:**

write a python code to calculate the factorial of a number using recursive approach with meaningful user defined functions and taking the input from the user.

### **Code:**

```
1  '''write a python code to calculate the factorial of a number using recursive
2  approach with meaningful user defined functions and taking the input from the user. . '''
3  def calculate_factorial_recursive(number):
4      """
5      Calculate the factorial of a given number using a recursive approach.
6
7      Parameters:
8      number (int): The number to calculate the factorial for.
9
10     Returns:
11     int: The factorial of the number
12     """
13     if number < 0:
14         return "Factorial is not defined for negative numbers."
15     if number == 0 or number == 1:
16         return 1
17     return number * calculate_factorial_recursive(number - 1)
18
19
20 # Get number input from user
21 num = int(input("Enter a number to calculate its factorial: "))
22
23 # Calculate and display the factorial
24 result = calculate_factorial_recursive(num)
25 print(result)
26
27
28
```

### **Output:**

```
Enter a number to calculate its factorial: -2
Factorial is not defined for negative numbers.
```

### **Justification:**

By doing this task, factorial logic is implemented using both iterative and recursive approaches.

The iterative version uses a loop for step-by-step multiplication, while

the recursive version uses function calls and base conditions.  
This comparison helps understand differences in readability, stack usage, performance, and when recursion should be avoided.