AI ASSISTED CODING

LAB ASSIGNMENT -10.2

Program	: B-TECH
Specialization	: AIML
Name of Student	: AKULA MANOJ
Enrollment-No	: 2403A52031
Batch No	: 02
Date	: 07-10-2025

TASK DESCRIPTION:1

AI-Assisted Code Review (Basic Errors)

-Write a Python program as shown below.
- Use an AI assistant to review and suggest corrections

PROMPT:- CORRECT THE MISTAKES IN THE FOLLOWING CODE.

GIVEN CODE:-

```
def calcFact(n):
  result=1
  x=0
  for i in range(1,n):
      result=result*i
  return result

def main():
      num = 5
      FACT = calcFact(num)
      print("the factorial of",num,"is",FACT)
      t=10
      if FACT>10:
             print("BIG Number")
      else:
       print("small number")

main()
```

-OBSERVATION ON GIVEN CODE:-

I found the simple mistakes, like the code is not running properly in the interpreter.

I found in mistake that it is using calcFact(num), using calFact is wrong.

Mainly Function is not properly written.

-GENERATED CODE BY AI WITH OUTPUT:-

```
def calculator(n):
    """Calculates the factorial of a non-negative integer n."""
    result = 1
    # The factorial of 0 is 1
    if n == 0:
        return 1
    # Calculate factorial for n > 0
    for i in range(1, n + 1):
        result = result * i
    return result

def main():
    # Set the number for which to calculate the factorial
    num = 5
    # Call the calculator function to get the factorial
    FACT = calculator(num)
    # Print the result
    print("the factorial of", num, "is", FACT)

    # Check if the factorial is a "BIG Number" or a "small number"
    if FACT > 10:
        print("BIG Number")
    else:
        print("small number")

# Execute the main function
main()
```

```
the factorial of 5 is 120
BIG Number
```

-OBSERVATION ON CORRECTED CODE:-

As expected, the code was corrected by using the correct calculator Function.

OVERALL OBSERVATION:

I found that it is throwing errors in the given code, and AI perfectly found the error and corrected it by generating a fresh new code with the correct Output.

TASK DESCRIPTION:2

Automatic Inline Comments.

•Write the Python code for Fibonacci as shown below and execute.

• Ask AI to improve variable names, add comments, and apply PEP8 formatting (cleaned up).

• Students evaluate which suggestions improve readability most. one.

Expected Output#2: Clean format Python code with much readability.

GIVEN CODE:-

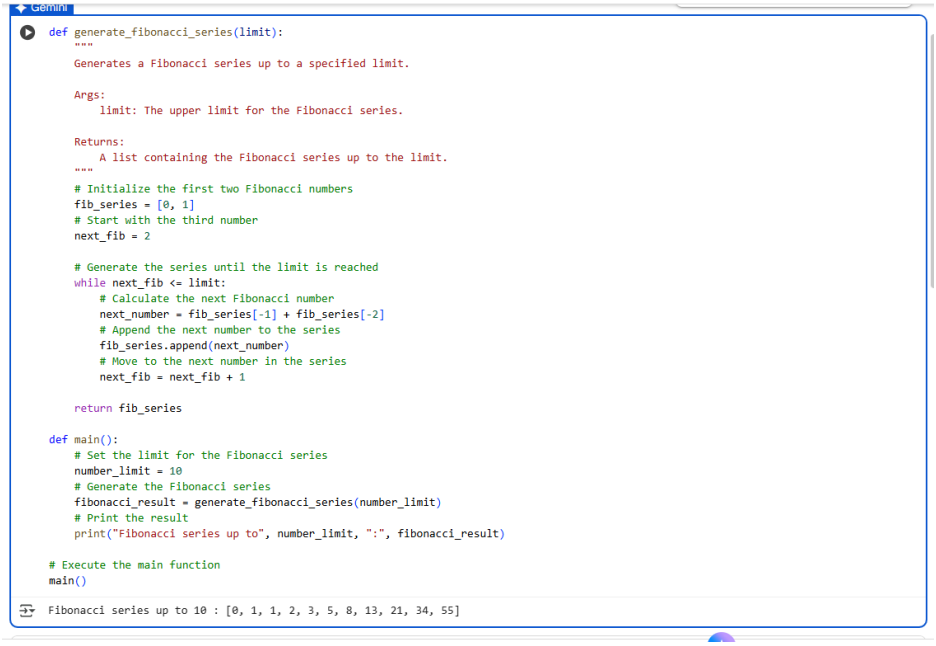
```
def f1(xX):
    a=0
    b=1
    c=2
    Zz=[a,b]
    while c<=xX:
        d=a+b
        Zz.append(d)
        a=b
        b=d
        c=c+1
    return Zz

def m():
      NN=10
      ans=f1(NN)
      print("fib series till",NN,":",ans)

m()
```

PROMPT:-correct the given code and use proper comments.

CORRECTED CODE WITH OUTPUT:-



OBSERVATION: The **given code** is incomplete, and the names are not given properly, which leads to wrong output. And the **AI corrected** and completed the full code with comments to get proper output.

TASK DESCRIPTION:3

• Write a Python script with 3–4 functions (e.g., calculator: add, subtract, multiply,
divide).
• Incorporate manual docstring in code with NumPy Style
• Use AI assistance to generate a module-level docstring + individual function
docstrings.
• Compare the AI-generated docstring with your manually written one

CODE:

```
def add(x, y):
  """ Adds two numbers and returns the result."""
  return x + y


def subtract(x, y):
```

```
    """Subtracts the second number from the first and returns the
result."""
    return x - y

def multiply(x, y):
    """Multiplies two numbers and returns the result."""
    return x * y

def divide(x, y):
    """Divides the first number by the second and returns the result.
    Handles division by zero by returning an error message."""
    if y == 0:
        return "Error: Division by zero!"
    return x / y

# Example usage:
num1 = 10
num2 = 5

print(f"{num1} + {num2} = {add(num1, num2)}")
print(f"{num1} - {num2} = {subtract(num1, num2)}")
print(f"{num1} * {num2} = {multiply(num1, num2)}")
print(f"{num1} / {num2} = {divide(num1, num2)}")
print(f"{num1} / 0 = {divide(num1, 0)}")
```

OUTPUT:

10 + 5 = 15

10 - 5 = 5

10 * 5 = 50

10 / 5 = 2.0

10 / 0 = Error: Division by zero!

MANUAL CODE WITH DOCSTRINGS:-

```
def add(x, y):
  """
  Adds two numbers.

    x: float or int
       The first number.
  y: float or int
       The second number.

  float or int
       The sum of the two numbers.
  """
  return x + y

def subtract(x, y):
```

```
    """
    Subtracts the second number from the first.

    x : float or int
        The first number.
    y : float or int
        The second number.

     float or int
        The difference between the two numbers.
    """
    return x - y

def multiply(x, y):
    """
    Multiplies two numbers.

     x : float or int
        The first number.
    y : float or int
        The second number.

    Returns
    -------
```

```
  float or int
      The product of the two numbers.
  """
  return x * y


def divide(x, y):
  """
  Divides the first number by the second.
  x : float or int
      The numerator.
  y : float or int
      The denominator.




  float or int or str
      The quotient of the two numbers, or an error message if division
by zero.
  """
  if y == 0:
    return "Error: Division by zero!"
  return x / y


# Example usage:
```

```
num1 = 10
num2 = 5

print(f"{num1} + {num2} = {add(num1, num2)}")
print(f"{num1} - {num2} = {subtract(num1, num2)}")
print(f"{num1} * {num2} = {multiply(num1, num2)}")
print(f"{num1} / {num2} = {divide(num1, num2)}")
print(f"{num1} / 0 = {divide(num1, 0)}")
```

AI to generate a module-level docstring + individual function docstrings:-

```
"""
This module provides basic arithmetic operations: addition, subtraction,
multiplication, and division.
"""

def add(x, y):
  """
  Adds two numbers.

  Parameters
  ----------
  x : float or int
      The first number.
  y : float or int
      The second number.

  Returns
  -------
  float or int
      The sum of the two numbers.
  """
  return x + y

def subtract(x, y):
  """
  Subtracts the second number from the first.

  Parameters
  ----------
  x : float or int
      The first number.
  y : float or int
      The second number.

  Returns
  -------
  float or int
      The difference between the two numbers.
```

```
      The second number.

  Returns
  -------
  float or int
      The difference between the two numbers.
  """
  return x - y

def multiply(x, y):
  """
  Multiplies two numbers.

  Parameters
  ----------
  x : float or int
      The first number.
  y : float or int
      The second number.

  Returns
  -------
  float or int
      The product of the two numbers.
  """
  return x * y

def divide(x, y):
  """
  Divides the first number by the second.

  Parameters
  ----------
  x : float or int
      The numerator.
  y : float or int
      The denominator.

  Returns
  -------
  float or int or str
      The quotient of the two numbers, or an error message if division by zero.
  """
```

```
    Returns
    -------
    float or int or str
        The quotient of the two numbers, or an error message if division by zero.
    """
    if y == 0:
      return "Error: Division by zero!"
    return x / y

# Example usage:
num1 = 10
num2 = 5

print(f"{num1} + {num2} = {add(num1, num2)}")
print(f"{num1} - {num2} = {subtract(num1, num2)}")
print(f"{num1} * {num2} = {multiply(num1, num2)}")
print(f"{num1} / {num2} = {divide(num1, num2)}")
print(f"{num1} / 0 = {divide(num1, 0)}")
```

```
10 + 5 = 15
10 - 5 = 5
10 * 5 = 50
10 / 5 = 2.0
10 / 0 = Error: Division by zero!
```

OBSERVATION:-

I found that the Doctstrings which is generated by AI with neat and clear.

Doctstrings is written in the type comments, not as detailed written there compared to the AI-generated Docstring.

THANK YOU