

# AI ASSISTED CODING

## ASSIGNMENT-10.4

E.RAVALI

2503A51L18

### TASK-1: AI-Assisted Code Review (Basic Errors)

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

```
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()
```

### PROMPT:

- Review the above code and suggest corrections. rewrite the correct code with comments.

```
# first10 > calc_fact
1 # Function to calculate factorial
2 def calc_fact(n):
3     result = 1
4     for i in range(1, n + 1): # loop should include 'n'
5         result *= i           # fixed syntax error
6     return result
7 # Main function
8 def main():
9     num = 5
10    fact = calc_fact(num) # fixed naming and function call
11
12    print("The factorial of", num, "is", fact)
13
14    if fact > 10: # changed FACT to fact
15        print("BIG Number")
16    else:
17        print("small number")
18 # Call the main function
19 main()

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS + ... ×
isted/first10"
The factorial of 5 is 120
BIG Number
PS C:\Users\DELL\OneDrive\SAHASRA SRU\AI assisted>
```

### TASK-2 :Automatic Inline Comments

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

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

```

def f1(x):
    a=0
    b=1
    c=2
    Zz=[a,b]
    while c<=x:
        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:

- USE AI to improve variable names, add comments, and apply PEP8 formatting (cleaned up) for the above python code.

```

second10.py > ...
1  # Function to generate Fibonacci series up to 'n' terms
2  def generate_fibonacci_series(n_terms):
3      first = 0
4      second = 1
5      count = 2 # Starts from 2 since 0 and 1 are already added
6      series = [first, second]
7      # Generate the rest of the series
8      while count < n_terms:
9          next_number = first + second
10         series.append(next_number)
11         first = second
12         second = next_number
13         count += 1
14     return series
15  # Main function
16  def main():
17      number_of_terms = 10
18      fibonacci_series = generate_fibonacci_series(number_of_terms)
19      print("Fibonacci series till", number_of_terms, "terms:", fibonacci_series)
20  # Execute main
21  main()
22

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS + ... | X

BIG Number
PS C:\Users\DELL\OneDrive\SAHASRA SRU\AI assisted> & C:/Users/DELL/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/DELL/OneDrive/SAHASRA SRU/AI assisted/second10.py"
Fibonacci series till 10 terms: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
PS C:\Users\DELL\OneDrive\SAHASRA SRU\AI assisted> []

```

## 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

## PROMPT:

- Use AI assistance to generate a module-level docstring + individual function docstrings.
- Compare the AI-generated docstring with your manually written one.

```
◆ third10man.py > ...
1  def add(a, b):
2      """
3          Add two numbers.
4          args
5          a : The first number.
6          b : The second number.
7          Returns The sum of a and b.
8          """
9      return a + b
10     def subtract(a, b):
11         """
12             Subtract one number from another.
13             args
14             a : The number to subtract from.
15             b : The number to subtract.
16             Returns The result of a - b.
17             """
18         return a - b
19     def multiply(a, b):
20         """
21             Multiply two numbers.
22             args
23             a : The first number.
24             b : The second number.
25             Returns The product of a and b.
26             """
27         return a * b
28     def divide(a, b):
29         """
30             Divide one number by another.
```

```
◆ third10mpy > ...
1  """
2  basic_calculator.py
3  A simple calculator module that provides basic arithmetic operations:
4  addition, subtraction, multiplication, and division.
5  This module is intended for demonstration or educational use,
6  showcasing the use of modular functions and NumPy-style docstrings.
7  Functions
8  -----
9  - add(a, b)
10 - subtract(a, b)
11 - multiply(a, b)
12 - divide(a, b)
13 """
14 def add(a, b):
15     """
16         Add two numbers.
17         Parameters
18         -----
19         a : int or float
20             | The first number.
21         b : int or float
22             | The second number.
23         Returns
24         -----
25         int or float
26             | The sum of a and b.
27         """
28     return a + b
29 def subtract(a, b):
30     """
31         Subtract one number from another.
32         Parameters
33         -----
34         a : int or float
35             | The number to subtract from.
36         b : int or float
37             | The number to subtract.
```

```
31     Subtract one number from another.
32
33     Parameters
34     -----
35     a : int or float
36         The number to subtract from.
37     b : int or float
38         The number to subtract.
39
40     Returns
41     -----
42     int or float
43         The result of a - b.
44
45     def multiply(a, b):
46         """
47             Multiply two numbers.
48             Parameters
49             -----
50             a : int or float
51                 The first number.
52             b : int or float
53                 The second number.
54
55             Returns
56             -----
57             int or float
58                 The product of a and b.
59
60             return a * b
61
62     def divide(a, b):
63         """
64             Divide one number by another.
65             Parameters
66             -----
67             a : int or float
68                 The numerator.
69             b : int or float
70                 The denominator.
71
72             Returns
73             -----
74             float
75                 The result of a / b.
76
77             Raises
78             -----
79             ValueError
80                 If b is zero.
```

## OBSERVATION:

### AI-assisted docstrings offer:

- Professional structure
- Better readability
- Tool compatibility
- Reusability in teams, APIs, and documentation websites