Assignment:10.2

Name:Aashutosh Kumar

HallTkt:2403a51316

Task1: AI-Assistant Code review.

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

Ans:-

Prompt:-There is some error in code review and correct it.

```python
def calcFact(n):
    result = 1
    for i in range(1, n + 1):  # Include n in factorial
        result = result * i
    return result

def main():
    num = 5
    FACT = calcFact(num)
    print("The factorial of", num, "is", FACT)

    if FACT > 10:
        print("BIG Number")
    else:
        print("small number")

main()
```

Observations & Corrections:

1. In calcFact(n), the loop should be for i in range(1, n+1): instead of range(1,n) (otherwise factorial is wrong).
2. The unused variable x=0 can be removed.
3. In main(), indentation of else: is wrong → should align with if FACT>10:.
4. Variable t=10 is declared but never used → can be removed.
5. Function call main() is correct, no change needed.

Task2: 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 evalute which suggestions improve readability most one.

Ans:-

Prompt:-Here's my Python code for generating the Fibonacci series. The variable names and formatting look messy. Can you review it, rename variables meaningfully, add inline comments, follow PEP8 style, and rewrite the code in a cleaner and more readable way?.

```python
def fibonacci_series(n):
    """
    Generate Fibonacci series up to n terms.
    """
    first, second = 0, 1
    series = [first, second]  # Starting values of Fibonacci series

    # Generate the series
    while len(series) < n:
        next_val = first + second
        series.append(next_val)
        first, second = second, next_val

    return series


def main():
    terms = 10
    result = fibonacci_series(terms)
    print("Fibonacci series till", terms, ":", result)


main()
```

Observations (One Line Each):

1. Function name f1 is not descriptive → should be renamed to fibonacci_series.
2. Variables a, b, c are unclear → better use first, second, count.
3. Variable Zz should be renamed to something meaningful like series.

4. No inline comments to explain steps → hard to follow for beginners.
5. Function m() is not descriptive → should be renamed to main.
6. Extra variable c is unnecessary → use while len(series) < n.
7. No docstring for functions → readability issue.

Task3:

- 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 functon docstrings.
- Compare the AI-generated docstring with your manually written one.

Ans:-

Prompt:- I've written a small Python calculator script with functions for add, subtract, multiply, and divide. I want you to review it, add Numpy-style docstrings for each function, and also generate a module-level docstring. Please point out any code smells and suggest improvements for readability and maintainability.

```
"""
Calculator Module
-----------------
This module provides basic arithmetic operations including addition,
subtraction, multiplication, and division.

Functions
---------
add(a, b) :
    Returns the sum of two numbers.
subtract(a, b) :
    Returns the difference of two numbers.
multiply(a, b) :
    Returns the product of two numbers.
divide(a, b) :
    Returns the quotient of two numbers if divisor is not zero.
"""


def add(a, b):
    """
    Add two numbers.

    Parameters
    ----------
    a : int or float
        First number.
    b : int or float
        Second number.

    Returns
    -------
    int or float
        The sum of a and b.
    """
    return a + b
```

```python
def subtract(a, b):
    """
    Subtract two numbers.

    Parameters
    ----------
    a : int or float
        First number.
    b : int or float
        Second number.

    Returns
    -------
    int or float
        The result of a - b.
    """
    return a - b


def multiply(a, b):
    """
    Multiply two numbers.

    Parameters
    ----------
    a : int or float
        First number.
    b : int or float
        Second number.

    Returns
    -------
    int or float
        The product of a and b.
    """
    return a * b
```

```python
def divide(a, b):
    """
    Divide two numbers.

    Parameters
    ----------
    a : int or float
        Numerator.
    b : int or float
        Denominator (must not be zero).

    Returns
    -------
    float
        The quotient of a / b.

    Raises
    ------
    ValueError
        If b is zero.
    """
    if b == 0:
        raise ValueError("Division by zero is not allowed.")
    return a / b
```

Observations (One Line Each)

1. Functions were missing **docstrings** → now added in **Numpy-style**.
2. No **module-level docstring** was present → added to explain overall purpose.
3. Function names are good, no poor naming → readable.
4. No code smells like *long function* or *duplicate code*.
5. Error handling for divide by zero was missing → added ValueError.
6. Code follows **PEP8 formatting** now (spaces, naming, docstrings).