

# AI-Assisted Coding

## Assignment 2.5

HTNo:2303A52203

Name:C.Vaishnav Reddy

Batch:45

### Task 1: Refactoring Odd/Even Logic (List Version)

❖ Scenario:

You are improving legacy code.

❖ Task:

Write a program to calculate the sum of odd and even numbers in a list, then refactor it using AI.

❖ Expected Output:

❖ improved code

Output:

The screenshot shows a Python development environment with the following details:

- Code Editor:** The main window displays the Python script `assignment_2.5.py`. The code reads a list of integers from user input, calculates the sum of odd and even numbers, and prints the results. It includes error handling for non-integer inputs.
- Terminal:** The terminal window shows the command `cd "d:/AI assited coding"; python assignment_2.5.py <<< "1 2 3 4 5 6 7 8 9 10"` being run, followed by the output of the program which correctly calculates the sums of odd and even numbers.
- Output Window:** The output window shows the program's execution results:

```
PS D:\AI assited coding> & C:/Users/vaish/AppData/Local/Programs/Python/Python314/python.exe "d:/AI assited coding/assignment_2.5.py"
=====
Sum of Odd And Even Numbers calculator
=====

Enter numbers separated by spaces: 1 2 3 4 5 6 7 8
-----
Numbers entered: [1, 2, 3, 4, 5, 6, 7, 8]
-----
Sum of odd numbers: 16
Sum of even numbers: 20
Total sum: 36
-----
```
- Right Panel:** A sidebar provides information about the task, including the title "SUM OF ODD AND EVEN NUMBERS IN LIST", instructions for testing ("Used echo to pipe data instead of here-string syntax"), and a summary of the program's features and test results.

## Task 2: Area Calculation Explanation

#### ❖ Scenario:

## You are onboarding a junior developer.

## ❖ Task:

Ask Gemini to explain a function that calculates the area of different shapes.

#### ❖ Expected Output:

## ➤ Code

## Output:

The screenshot shows a VS Code interface with the following details:

- Title Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Search Bar:** Q AI assisted coding.
- Left Sidebar:** Preview Channel, GEMINI CODE ASSIST: CHAT, a context menu with options like 'Default' and 'Agent', and a status bar showing 'Ln 18, Col 1'.
- Code Editor:** Two tabs are open: 'area\_calculator.py U' and 'assignment\_2.5.py U'. The 'area\_calculator.py' tab contains the following code:

```
❶ area_calculator.py > calculate_area
❷ import math
❸
❹ def calculate_area(shape: str, *args) -> float:
❺     """
❻         Calculates the area of a given shape.
❼     """
⪻     Supported shapes and required arguments:
⪼     - 'circle': radius
⪽     - 'rectangle': length, width
⪾     - 'triangle': base, height
⪿
⪻     Args:
⪼         shape (str): The name of the shape (case-insensitive).
⪽         *args: Variable length argument list for dimensions.
⪾
⪻     Returns:
⪼         float: The calculated area.
⪾
⪻     Raises:
⪼         ValueError: If the shape is unsupported or dimensions are invalid.
⪽         ...
⪾
⪻     shape = shape.lower().strip()
⪽
⪻     if shape == 'circle':
⪼         if len(args) != 1:
⪽             raise ValueError("Circle requires exactly 1 argument: radius.")
⪽         radius = args[0]
⪽         if radius < 0:
⪽             raise ValueError("Radius cannot be negative.")
```
- Bottom Status Bar:** PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, POSTMAN CONSOLE.
- Terminal:** Shows the command line output:

```
PS D:\AI assisted coding> & C:/Users/vaish/AppData/Local/Programs/Python/Python314/python.exe "d:/AI assisted coding/area_calculator.py"
Circle Area (r=5): 78.54
Rectangle Area (l=5, w=4): 40.00
Triangle Area (b=6, h=3): 9.00
PS D:\AI assisted coding>
```

### Task 3: Prompt Sensitivity Experiment

#### ❖ Scenario:

You are testing how AI responds to different prompts.

## ❖ Task:

Use Cursor AI with different prompts for the same problem and observe code changes.

#### ❖ Expected Output:

## ➤ Prompt list

## ➤ Code variations

Prompt:-

=>Calculate the factorial of a number

The screenshot shows the AI assisted coding interface. On the left, there's a file explorer with various files like .idea, assignment\_2.5.py, and test\_results.txt. The main area displays a "Factorial calculation" card with the prompt "calculate the factorial of a number". It shows two code snippets: one using a loop and one using recursion. Below the cards, there's a message asking for preferred language and a command input field.

```
def factorial(n):
    if n < 0:
        raise ValueError("Factorial is not defined for negative numbers")
    result = 1
    for i in range(1, n + 1):
        result *= i
    return result

# Example:
print(factorial(5)) # Output: 120
```

```
def factorial_recursive(n):
    if n < 0:
        raise ValueError("Factorial is not defined for negative numbers")
    if n == 0 or n == 1:
        return 1
    return n * factorial_recursive(n - 1)
```

=> write factorial function in python using recursion

The screenshot shows the AI assisted coding interface. The file explorer on the left is similar to the previous one. The main area displays a "Factorial calculation" card with the prompt "write factorial function in python using recursion". It shows a single recursive code snippet. A tooltip at the bottom left suggests launching the cursor from the command line. A message at the bottom right says "Want to launch Cursor from the command line? You can add the 'code' or 'cursor' commands now, or later with Ctrl+Shift+P and typing 'install code command'." There's also a "Plan, @ for context, / for commands" input field.

```
def factorial(n):
    if n < 0:
        raise ValueError("Factorial is not defined for negative numbers")
    if n == 0 or n == 1:
        return 1
    return n * factorial(n - 1)

# Example usage:
print(factorial(5)) # Output: 120
```