

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 displays a VS Code editor with a Python file named `assignment_2.5.py`. The code defines a `main` function that takes user input, splits it into a list of integers, and calculates the sum of odd and even numbers using a `calculate_odd_even_sums` function. The code also includes a `ValueError` exception handler for invalid input.

The AI chat interface on the right provides suggestions for improving the code, such as using `echo` to pipe data instead of here-string syntax. It also lists key features of the program, including the `calculate_odd_even_sums` function, input validation, clear output, and well-documented code.

The terminal window at the bottom shows the program's output for the input `1 2 3 4 5 6 7 8 9`. The output displays the sum of odd numbers (16), the sum of even numbers (20), and the total sum (36).

```
def main():
    # Convert input to list of integers
    numbers = [int(x) for x in user_input.split()]

    # Calculate sums
    sum_odd, sum_even = calculate_odd_even_sums(numbers)

    # Display results
    print("\n" + "-" * 50)
    print(f"Numbers entered: {numbers}")
    print("-" * 50)
    print(f"Sum of odd numbers: {sum_odd}")
    print(f"Sum of even numbers: {sum_even}")
    print(f"Total sum: {sum_odd + sum_even}")
    print("-" * 50)

except ValueError:
    print("Error: Please enter valid integers only.")
```

Sum of Odd and Even Numbers Calculator

Enter numbers separated by spaces: 1 2 3 4 5 6 7 8 9

Numbers entered: [1, 2, 3, 4, 5, 6, 7, 8, 9]

Sum of odd numbers: 16
Sum of even numbers: 20
Total sum: 36

Task 2: Area Calculation Explanation

❖ Scenario:

You are onboarding a junior developer.

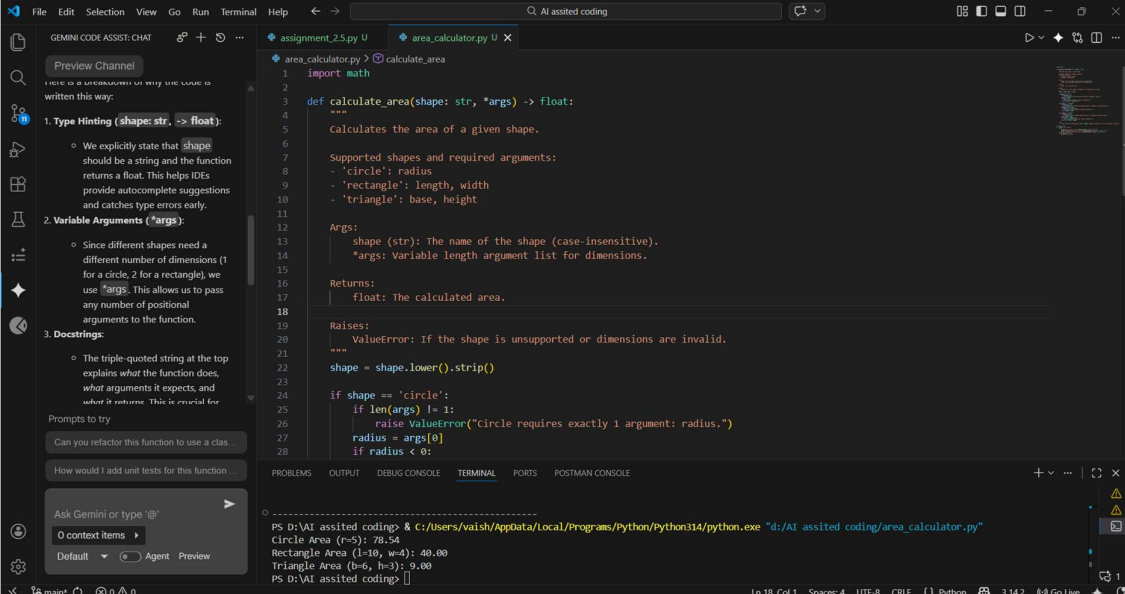
❖ Task:

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

❖ Expected Output:

➤ Code

Output:



The screenshot shows the VS Code interface with a Python file named `area_calculator.py` and the Gemini AI chat assistant. The code defines a function `calculate_area` that takes a shape name and dimensions as arguments and returns the area. The chat assistant provides a detailed explanation of the function's parameters, arguments, and return value.

```
1 import math
2
3 def calculate_area(shape: str, *args) -> float:
4     """
5     Calculates the area of a given shape.
6
7     Supported shapes and required arguments:
8     - 'circle': radius
9     - 'rectangle': length, width
10    - 'triangle': base, height
11
12    Args:
13        shape (str): The name of the shape (case-insensitive).
14        *args: Variable length argument list for dimensions.
15
16    Returns:
17        float: The calculated area.
18
19    Raises:
20        ValueError: If the shape is unsupported or dimensions are invalid.
21    """
22    shape = shape.lower().strip()
23
24    if shape == 'circle':
25        if len(args) != 1:
26            raise ValueError("Circle requires exactly 1 argument: radius.")
27        radius = args[0]
28        if radius < 0:
```

Preview Channel

I have to be a second hand to help the user to write this way:

- 1. Type Hinting (shape: str, -> float):**
 - We explicitly state that `shape` should be a string and the function returns a float. This helps IDEs provide autocomplete suggestions and catches type errors early.
- 2. Variable Arguments (*args):**
 - Since different shapes need a different number of dimensions (1 for a circle, 2 for a rectangle), we use `*args`. This allows us to pass any number of positional arguments to the function.
- 3. Docstrings:**
 - The triple-quoted string at the top explains *what* the function does, *what* arguments it expects, and *what* it returns. This is crucial for

Prompts to try

Can you refactor this function to use a class

How would I add unit tests for this function

Ask Gemini or type '@'

0 context items

Default Agent Preview

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=10, 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.

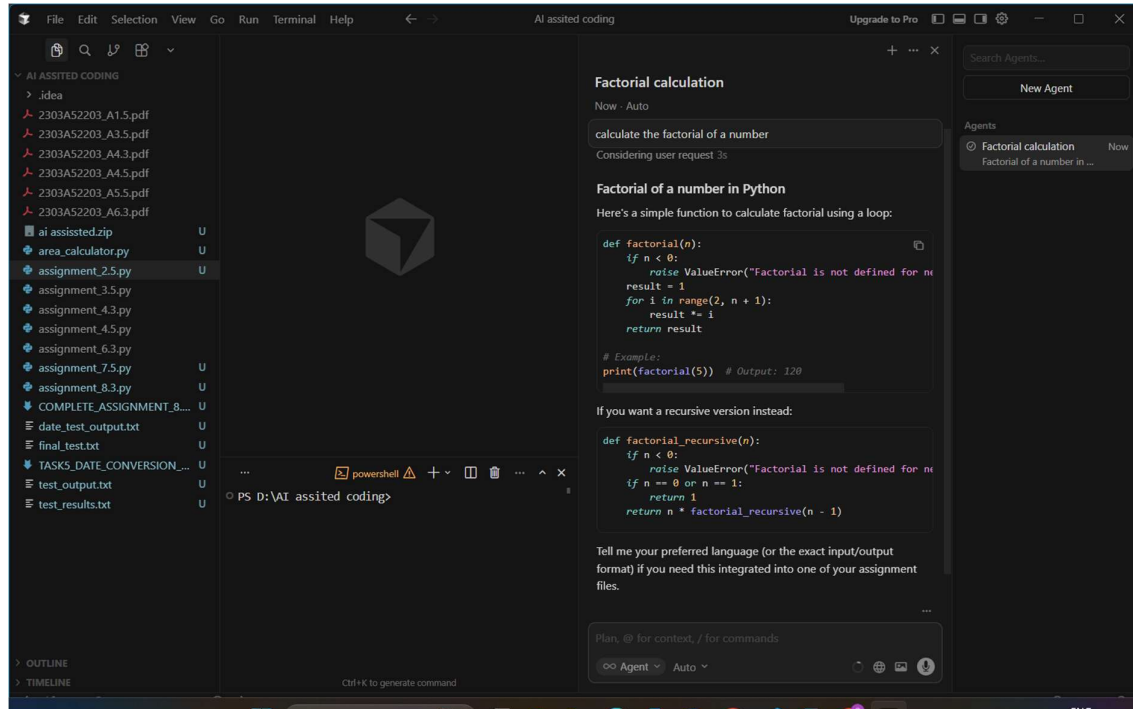
❖ Expected Output:

➤ Prompt list

➤ Code variations

Prompt:-

=> Calculate the factorial of a number



=> write factorial function in python using recursion

