

2303A52315

BATCH 45

ASS 2.5

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

❖ Scenario:

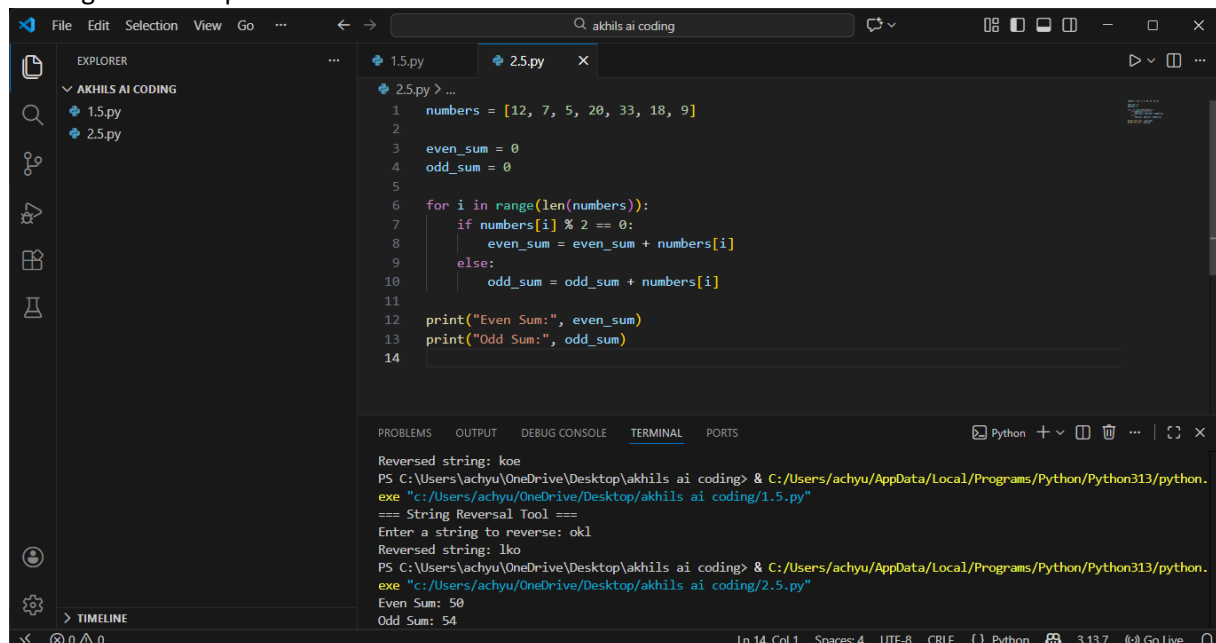
You are improving legacy code.

❖ Task:

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

❖ Expected Output:

❖ Original and improved code



The screenshot shows a Visual Studio Code editor window with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'AKHILS AI CODING' with two files: '1.5.py' and '2.5.py'. The code editor is open to '2.5.py', which contains the following Python code:

```
1 numbers = [12, 7, 5, 20, 33, 18, 9]
2
3 even_sum = 0
4 odd_sum = 0
5
6 for i in range(len(numbers)):
7     if numbers[i] % 2 == 0:
8         even_sum = even_sum + numbers[i]
9     else:
10        odd_sum = odd_sum + numbers[i]
11
12 print("Even Sum:", even_sum)
13 print("Odd Sum:", odd_sum)
14
```

The terminal at the bottom shows the output of running the script:

```
Reversed string: koe
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> & C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/achyu/OneDrive/Desktop/akhils ai coding/1.5.py"
=== String Reversal Tool ===
Enter a string to reverse: okl
Reversed string: lko
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> & C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/achyu/OneDrive/Desktop/akhils ai coding/2.5.py"
Even Sum: 50
Odd Sum: 54
```

Task 2: Area Calculation Explanation

❖ Scenario:

You are onboarding a junior developer.

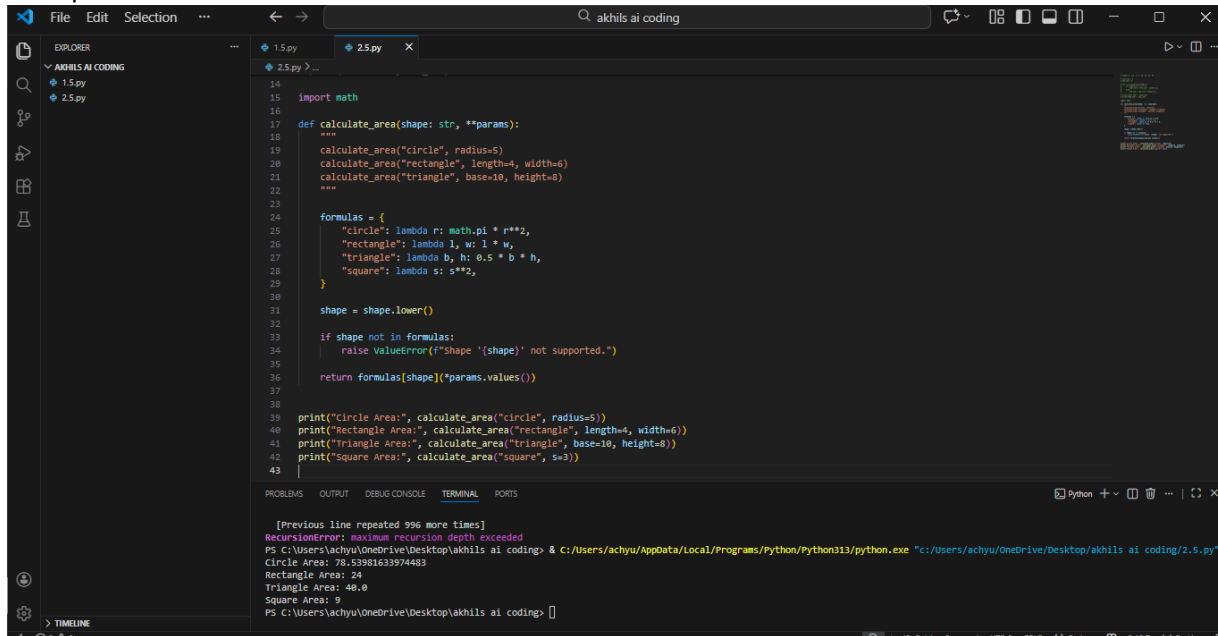
❖ Task:

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

❖ Expected Output:

➤ Code

➤ Explanation



The screenshot shows a code editor with a file named `2.5.py`. The code defines a recursive function `calculate_area` that calculates the area of a circle, rectangle, or triangle. The function uses a dictionary of formulas and recursive calls. The execution output in the terminal shows a `RecursionError: maximum recursion depth exceeded` followed by the calculated areas for each shape.

```
14
15 import math
16
17 def calculate_area(shape: str, **params):
18     """
19     calculate_area("circle", radius=5)
20     calculate_area("rectangle", length=4, width=6)
21     calculate_area("triangle", base=10, height=8)
22     """
23
24     formulas = {
25         "circle": lambda r: math.pi * r**2,
26         "rectangle": lambda l, w: l * w,
27         "triangle": lambda b, h: 0.5 * b * h,
28         "square": lambda s: s**2,
29     }
30
31     shape = shape.lower()
32
33     if shape not in formulas:
34         raise ValueError(f"Shape '{shape}' not supported.")
35
36     return formulas[shape](**params.values())
37
38 print("Circle Area:", calculate_area("circle", radius=5))
39 print("Rectangle Area:", calculate_area("rectangle", length=4, width=6))
40 print("Triangle Area:", calculate_area("triangle", base=10, height=8))
41 print("Square Area:", calculate_area("square", s=3))
42
43
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

[Previous line repeated 996 more times]
RecursionError: maximum recursion depth exceeded
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> & C:\Users\achyu\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/achyu/OneDrive/Desktop/akhils ai coding/2.5.py"
Circle Area: 78.53981633974483
Rectangle Area: 24
Triangle Area: 40.0
Square Area: 9
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding>

Task 3: Prompt Sensitivity Experiment

❖ Scenario:

You are testing how AI responds to different prompts.

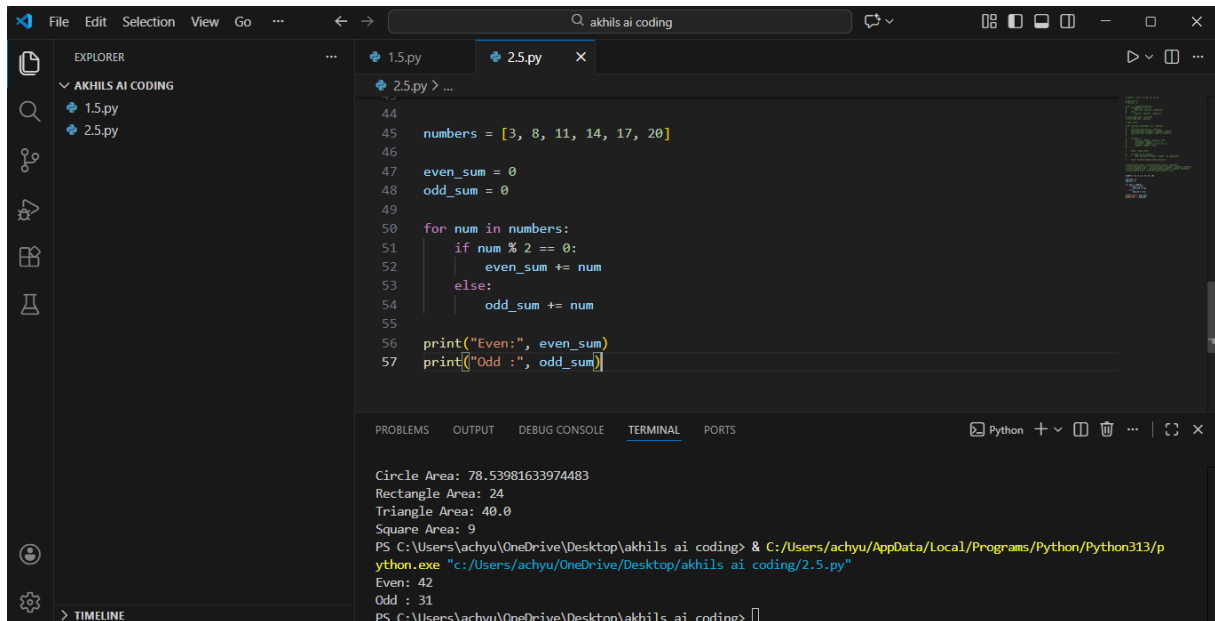
❖ Task:

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

❖ Expected Output:

➤ Prompt list

➤ Code variations



The screenshot shows a Visual Studio Code editor window with a dark theme. The Explorer sidebar on the left shows a project named 'AKHILS AI CODING' containing two files: '1.5.py' and '2.5.py'. The main editor area displays the contents of '2.5.py', which is a Python script. The script defines a list of numbers, calculates the sum of even and odd numbers, and prints the results. The output of the script is visible in the integrated terminal at the bottom, which shows the execution of the script and the resulting even and odd sums.

```
44
45 numbers = [3, 8, 11, 14, 17, 20]
46
47 even_sum = 0
48 odd_sum = 0
49
50 for num in numbers:
51     if num % 2 == 0:
52         even_sum += num
53     else:
54         odd_sum += num
55
56 print("Even:", even_sum)
57 print("Odd :", odd_sum)
```

Circle Area: 78.53981633974483
Rectangle Area: 24
Triangle Area: 40.0
Square Area: 9
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> & C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/achyu/OneDrive/Desktop/akhils ai coding/2.5.py"
Even: 42
Odd : 31
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> |

Task 4: Tool Comparison Reflection

❖ Scenario:

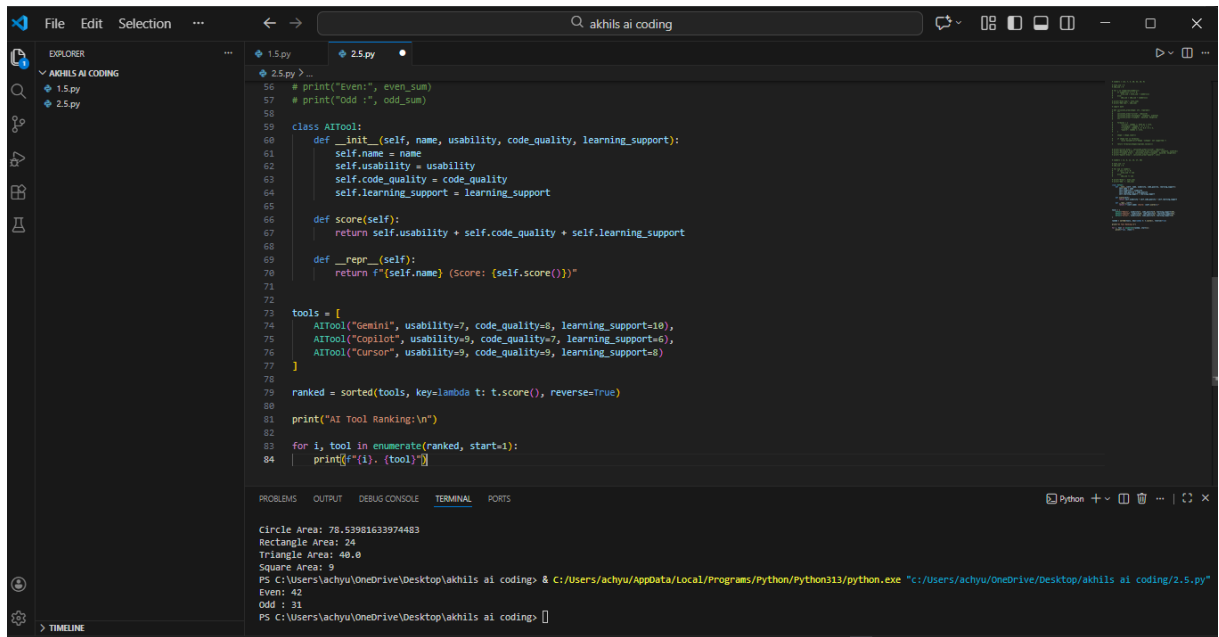
You must recommend an AI coding tool.

❖ Task:

Based on your work in this topic, compare Gemini, Copilot, and Cursor AI for usability and code quality.

❖ Expected Output:

Short written reflection



```
File Edit Selection ...  ← →  Q akhils ai coding  ↻  🔍  📄  📄  📄  -  □  ×

EXPLORER  ...  2.5.py  2.5.py
AKHILS AI CODING
1.5.py
2.5.py

2.5.py
56 # print("Even:", even_sum)
57 # print("Odd :", odd_sum)
58
59 class AITool:
60     def __init__(self, name, usability, code_quality, learning_support):
61         self.name = name
62         self.usability = usability
63         self.code_quality = code_quality
64         self.learning_support = learning_support
65
66     def score(self):
67         return self.usability + self.code_quality + self.learning_support
68
69     def __repr__(self):
70         return f"{self.name} (Score: {self.score()})"
71
72
73 tools = [
74     AITool("Gemini", usability=7, code_quality=8, learning_support=10),
75     AITool("Copilot", usability=9, code_quality=7, learning_support=6),
76     AITool("Cursor", usability=9, code_quality=9, learning_support=8)
77 ]
78
79 ranked = sorted(tools, key=lambda t: t.score(), reverse=True)
80
81 print("AI Tool Ranking:\n")
82
83 for i, tool in enumerate(ranked, start=1):
84     print(f"{i}. {tool}")

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
Python +  -  🗑  ...  |  🔍  ×

Circle Area: 78.53981633974483
Rectangle Area: 24
Triangle Area: 40.8
Square Area: 9
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> & C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/achyu/OneDrive/Desktop/akhils ai coding/2.5.py"
Even: 42
Odd : 31
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> 
```