

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

```
File Edit Selection View Go ... 1.5.py 2.5.py ...
EXPLORER AKHILS AI CODING 1.5.py 2.5.py ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
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
Ln 14, Col 1  Spaces: 4  UTF-8  CRLF  {} Python  3.13.7  (•) Go Live
```

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 VS Code interface with the following details:

- File Explorer:** Shows two files: 1.5.py and 2.5.py.
- Code Editor:** The active file is 2.5.py, containing Python code for calculating the area of various shapes using lambda functions. The code includes imports for math, defines a calculate_area function, and uses a dictionary of lambda functions to calculate the area based on the shape name.
- Terminal:** Shows the output of running the script, which prints the area for a circle, rectangle, triangle, and square.
- Status Bar:** Shows the Python version (Python 3.11.2) and other system information.

```
14 import math
15
16 def calculate_area(shape: str, **params):
17     """
18         calculate_area("circle", radius=5)
19         calculate_area("rectangle", length=4, width=6)
20         calculate_area("triangle", base=10, height=8)
21         calculate_area("square", s=3)
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)
37
38
39 print("Circle Area:", calculate_area("circle", radius=5))
40 print("Rectangle Area:", calculate_area("rectangle", length=4, width=6))
41 print("Triangle Area:", calculate_area("triangle", base=10, height=8))
42 print("Square Area:", calculate_area("square", s=3))
43
```

```
[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/Python311/python.exe "C:/Users/achyu/OneDrive/Desktop/akhils ai coding/2.5.py"
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> 48.0
Rectangle Area: 24
Triangle Area: 48.0
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 code editor interface with two tabs open: '1.5.py' and '2.5.py'. The '2.5.py' tab is active, displaying the following Python code:

```
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)
```

The terminal below the editor shows the execution of the script and its output:

```
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

A screenshot of the Visual Studio Code (VS Code) interface. The left sidebar shows a folder named 'AKHILS AI CODING' containing two files: '1.5.py' and '2.5.py'. The right pane displays the content of '2.5.py'. The code defines a class 'AITool' with methods for calculating a score based on usability, code quality, and learning support, and for printing the tool's name and score. It also lists three tools: Gemini, Copilot, and Cursor, and prints their scores. The terminal at the bottom shows the execution of the script and its output.

```
File Edit Selection ... ← → Q akhils ai coding
EXPLORER AKHILS AI CODING 1.5.py 2.5.py ...
1.5.py
2.5.py
1.5.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}")
85
86
87 Circle Area: 78.53981633974483
88 Rectangle Area: 24
89 Triangle Area: 40.0
90 Square Area: 9
91 PS C:\Users\achyu\Desktop\akhils ai coding> & C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/achyu/Desktop/akhils ai coding/2.5.py"
92 Even: 42
93 Odd: 31
94 PS C:\Users\achyu\Desktop\akhils ai coding> []

```