

Assignment => 2

Name =>Sai akshith

Batch=>45

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

Code

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

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
even_sum = sum(x for x in numbers if x % 2 == 0)
odd_sum = sum(x for x in numbers if x % 2 != 0)
print("Sum of even numbers:", even_sum)
print("Sum of odd numbers:", odd_sum)

# Refactored code

def calculate_sums(numbers):
    even_sum = sum(x for x in numbers if x % 2 == 0)
    odd_sum = sum(x for x in numbers if x % 2 != 0)
    return even_sum, odd_sum

even_sum, odd_sum = calculate_sums(numbers)
print("Sum of even numbers:", even_sum)
print("Sum of odd numbers:", odd_sum)
```

The screenshot shows a Microsoft Visual Studio Code (VS Code) window with the following details:

- File Explorer:** Shows a folder named "AI-PROGRAMMING" containing files: exp 1.py, exp 2.py, exp 5.py, exp 6.py, and exp4 3.py, along with a log file user_activity.log.
- Code Editor:** Displays the content of "exp 2.py". The code generates a list of numbers from 1 to 10, then calculates and prints the sum of even and odd numbers separately.
- Terminal:** Shows the output of running the script in Python 3.14. The terminal window title is "Python" and it displays the following output:

```
PS C:\Users\SAIAKSHITH\OneDrive\Documents\AI-PROGRAMMING> & C:/Python314/python.exe "C:/Users/SAIAKSHITH/OneDrive/Documents/AI-PROGRAMMING/exp 2.py"
Sum of even numbers: 30
Sum of odd numbers: 25
Sum of even numbers: 30
Sum of odd numbers: 25
PS C:\Users\SAIAKSHITH\OneDrive\Documents\AI-PROGRAMMING>
```
- Bottom Status Bar:** Provides information about the current file (ln 14, Col 38), the programming language (Python), and the date (04-02-2026).

Task 2: Area Calculation Explanation

```
# generate a code to calculate the area of different shapes like circle, square, rectangle
```

```
import math
```

```
def area_of_circle(radius):
```

```
return math.pi * radius ** 2
```

```
def area_of_square(side):
```

return side ** 2

```
def area_of_rectangle(length, width):
```

return length * width

Example usage

```
circle_area = area_of_circle(5)
```

square area = area of square

rectangle area = area of rectangle

```
print("Area of Circle:", circle_area)
```

```
print("Area of Square:", square_area)
```

```
print("Area of Rectangle:" rectangle_area)
```

The screenshot shows a code editor window with a dark theme. In the top left, there's a tab labeled 'exp 2.py'. Below it, a terminal window displays the output of running the script. The terminal output shows the sum of even and odd numbers from 1 to 10, followed by the areas of a circle, square, and rectangle.

```
exp 2.py > area_of_rectangle
16 # generate a code to calculate the area of different shapes like circle, square, rectangle
17 import math
18 def area_of_circle(radius):
19     return math.pi * radius ** 2
20 def area_of_square(side):
21     return side ** 2
22 def area_of_rectangle(length, width):
23     return length * width
24 # Example usage
25 circle_area = area_of_circle(5)
26 square_area = area_of_square(4)
27 rectangle_area = area_of_rectangle(4, 6)
28 print("Area of Circle:", circle_area)
29 print("Area of Square:", square_area)
30 print("Area of Rectangle:", rectangle_area)
31

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\SAIAKSHITH\OneDrive\Documents\AI-PROGRAMMING> & C:/Python314/python.exe "c:/Users/SAIAKSHITH/OneDrive/Documents/AI-PROGRAMMING/exp 2.py"
Sum of even numbers: 30
Sum of odd numbers: 25
Sum of even numbers: 30
Sum of odd numbers: 25
PS C:\Users\SAIAKSHITH\OneDrive\Documents\AI-PROGRAMMING>
```

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

PROMT :

compare Gemini, Copilot, and Cursor AI for usability and code quality.

CODE:

```
32     '''Task 4: Tool Comparison Reflection
33     ♦ Scenario:
34     You must recommend an AI coding tool.
35     ♦ Task:
36     Based on your work in this topic, compare Gemini, Copilot, and Cursor AI
37     for usability and code quality.
38     ♦ Expected Output:
39     Short written reflection'''
40
41     # Reflection:
42     # After using Gemini, Copilot, and Cursor AI for code generation and refactoring tasks,
43     # I found that each tool has its strengths and weaknesses. Copilot excels in usability, providing
44     # seamless integration with popular code editors and offering context-aware suggestions that enhance productivity.
```