

# AI ASSISTANT CODING

## ASSIGNMENT-02

**Name:** Kashaboina.Archana

**HT.No:** 2303A51329

**Batch:** 20

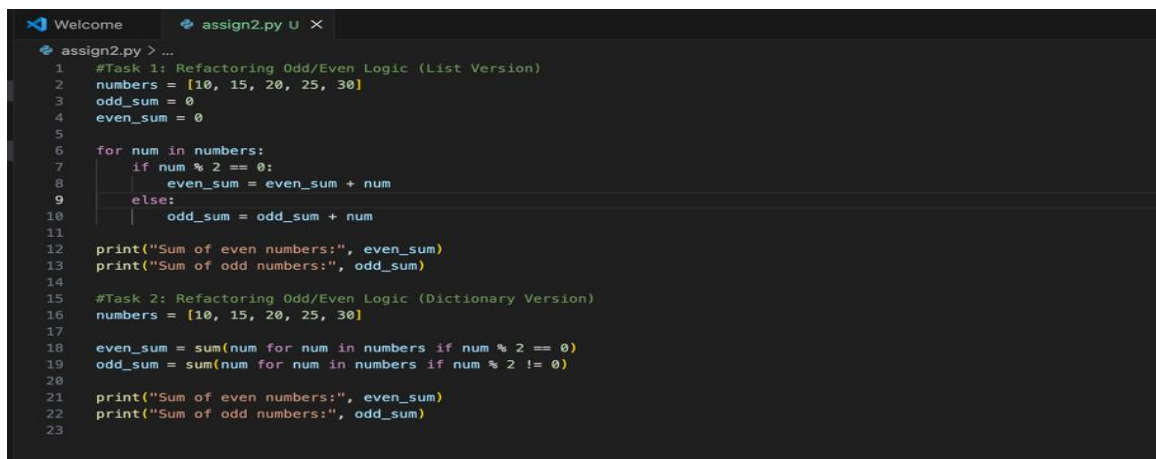
**Task 1:** Task 1: Refactoring Odd/Even Logic (List Version) Write a program to calculate the sum of odd and even numbers in a list, then refactor it using AI.

### Scenario

You are improving legacy code.

**Prompt:** calculate the sum of odd and even numbers and refactor it using AI.

### Code:



```
1 #Task 1: Refactoring Odd/Even Logic (List Version)
2 numbers = [10, 15, 20, 25, 30]
3 odd_sum = 0
4 even_sum = 0
5
6 for num in numbers:
7     if num % 2 == 0:
8         even_sum = even_sum + num
9     else:
10        odd_sum = odd_sum + num
11
12 print("Sum of even numbers:", even_sum)
13 print("Sum of odd numbers:", odd_sum)
14
15 #Task 2: Refactoring Odd/Even Logic (Dictionary Version)
16 numbers = [10, 15, 20, 25, 30]
17
18 even_sum = sum(num for num in numbers if num % 2 == 0)
19 odd_sum = sum(num for num in numbers if num % 2 != 0)
20
21 print("Sum of even numbers:", even_sum)
22 print("Sum of odd numbers:", odd_sum)
23
```

### Result:



```
Sum of even numbers: 60
Sum of odd numbers: 40
Sum of even numbers: 60
Sum of odd numbers: 40
```

### Observation:

The refactored version removes manual looping and conditional accumulation, making the code shorter, more readable, and efficient while producing the same output.

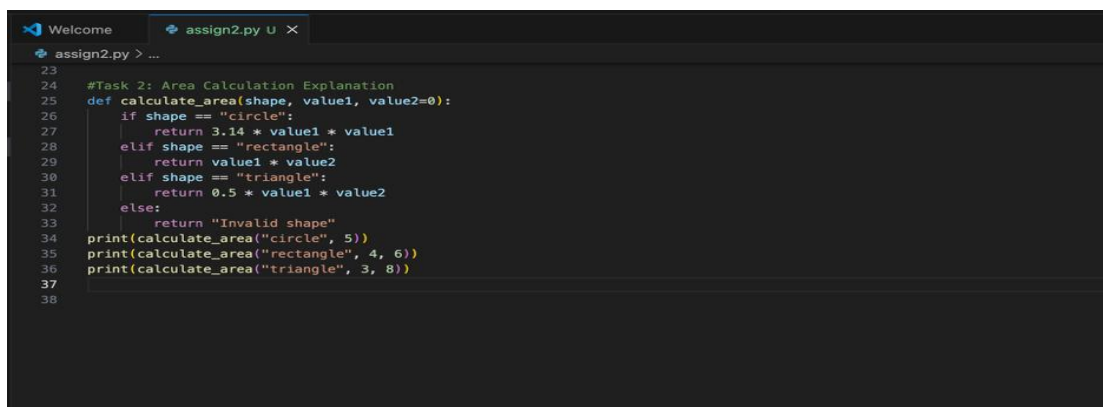
**Task 2:** Area Calculation Explanation. Ask Gemini to explain a function that calculates the area of different shapes.

### Scenario

You are onboarding a junior developer.

**Prompt:** give a function that calculates the area of different shapes

**Code:**

A screenshot of a code editor with a dark theme. The editor has two tabs: 'Welcome' and 'assign2.py'. The 'assign2.py' tab is active, showing a Python script. The script starts with a comment '#Task 2: Area Calculation Explanation' on line 24. It defines a function 'calculate\_area(shape, value1, value2=0)' on line 25. The function uses an if-elif-else structure to calculate the area for 'circle', 'rectangle', and 'triangle' shapes. For 'circle', it returns 3.14 \* value1 \* value1. For 'rectangle', it returns value1 \* value2. For 'triangle', it returns 0.5 \* value1 \* value2. For any other shape, it returns 'Invalid shape'. Below the function definition, there are three print statements: 'print(calculate\_area("circle", 5))', 'print(calculate\_area("rectangle", 4, 6))', and 'print(calculate\_area("triangle", 3, 8))'. The line numbers 23 through 38 are visible on the left side of the editor.

**Result:**

A screenshot of a terminal window with a dark background. It displays the output of the Python script. The output consists of five lines: 'Sum of even numbers: 60', 'Sum of odd numbers: 40', 'Sum of even numbers: 60', 'Sum of odd numbers: 40', and '78.5'. The line numbers 24 and 12.0 are visible on the left side of the terminal.

**Observation:**

Gemini effectively explains both the logic and mathematical reasoning in a clear and structured way, making it suitable for junior developers and beginners.

**Task 3:** Prompt Sensitivity Experiment Use Cursor AI with different prompts for the same problem and observe code changes.

## Scenario

You are testing how AI responds to different prompts.

**Prompt-1:** Write a Python program to check whether a number is even or odd

**Prompt-2:** Write optimized Python code with error handling to check even or odd

**Prompt-3:** Write a reusable Python function to check if a number is even or odd

**Code:**

```
Welcome  assign2.py U x
assign2.py > ...
38
39 #Task 3: Prompt Sensitivity Experiment (Cursor AI)
40 #Prompt-1: "Write a Python program to check whether a number is even or odd."
41 num = int(input("Enter a number: "))
42
43 if num % 2 == 0:
44     print("Even")
45 else:
46     print("Odd")
47 #Prompt-2: "Write optimized Python code with error handling to check even or odd."
48 try:
49     num = int(input("Enter a number: "))
50     print("Even" if num % 2 == 0 else "Odd")
51 except ValueError:
52     print("Invalid input")
53
54 #Prompt-3: "Write a reusable Python function to check if a number is even or odd."
55 def check_even_odd(num):
56     return "Even" if num % 2 == 0 else "Odd"
57
58
```

**Result:**

```
Sum of even numbers: 60
Sum of odd numbers: 40
Sum of even numbers: 60
Sum of odd numbers: 40
78.5
24
12.0
Enter a number: 3
Odd
Enter a number: 2
Even
```

