

AI ASSISTANT CODING

ASSIGNMENT-02

Name: Kashaboina.Archana

HT.No: 2303A51329

Batch: 20

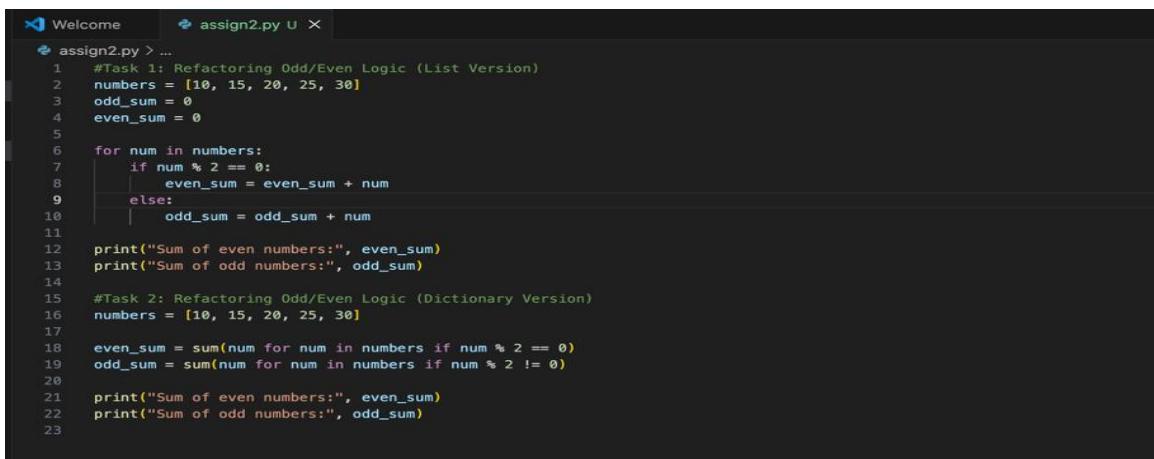
Task 1: Task 1: Refactoring Odd/Even Logic (List Version) Write a program to calculate the sum of odd and even numbers in a list, then refactor it using AI.

Scenario

You are improving legacy code.

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

Code:



```
Welcome assign2.py U X
assign2.py > ...
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.

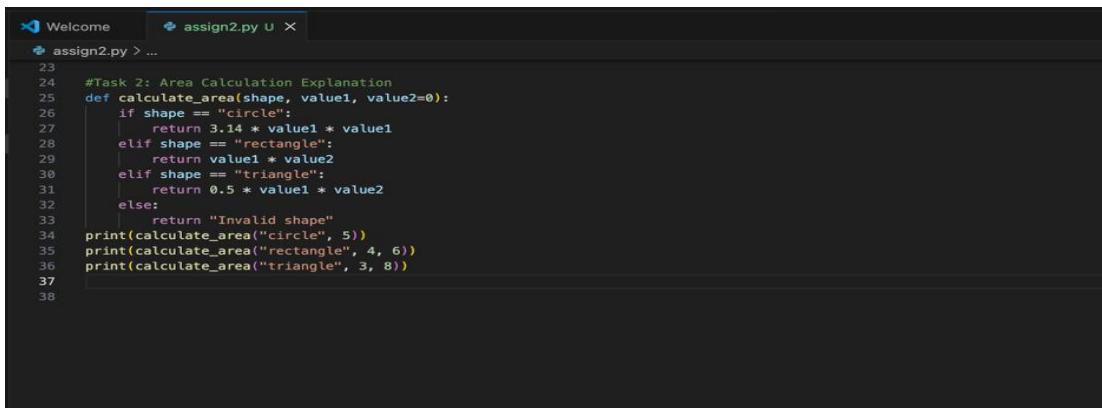
Task 2: Area Calculation Explanation. Ask Gemini to explain a function that calculates the area of different shapes.

Scenario

You are onboarding a junior developer.

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

Code:



```
23
24     #Task 2: Area Calculation Explanation
25     def calculate_area(shape, value1, value2=0):
26         if shape == "circle":
27             return 3.14 * value1 * value1
28         elif shape == "rectangle":
29             return value1 * value2
30         elif shape == "triangle":
31             return 0.5 * value1 * value2
32         else:
33             return "Invalid shape"
34     print(calculate_area("circle", 5))
35     print(calculate_area("rectangle", 4, 6))
36     print(calculate_area("triangle", 3, 8))
37
38
```

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
```

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

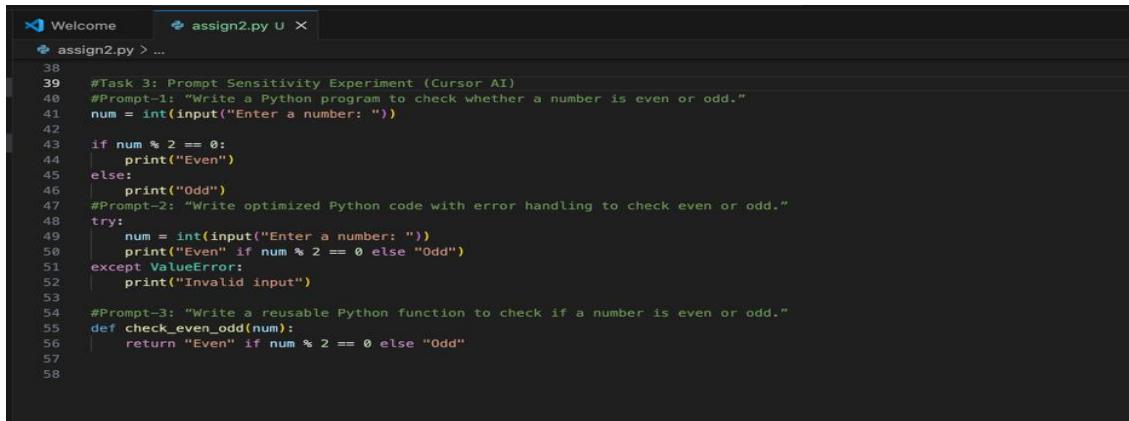
You are testing how AI responds to different prompts.

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:



```
 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
24.0
Enter a number: 3
Odd
Enter a number: 2
Even
```

