

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)
```

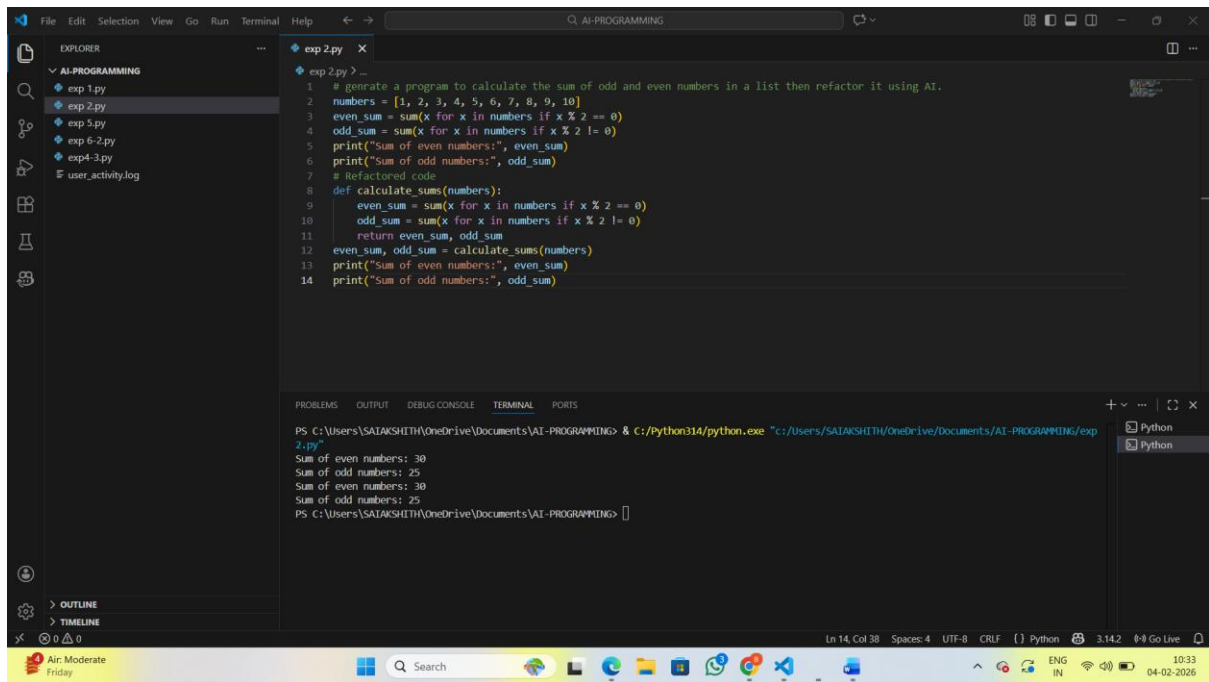
```
    odd_sum = sum(x for x in numbers if x % 2 != 0)
```

```
    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)
```



Task 2: Area Calculation Explanation

genrate 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)

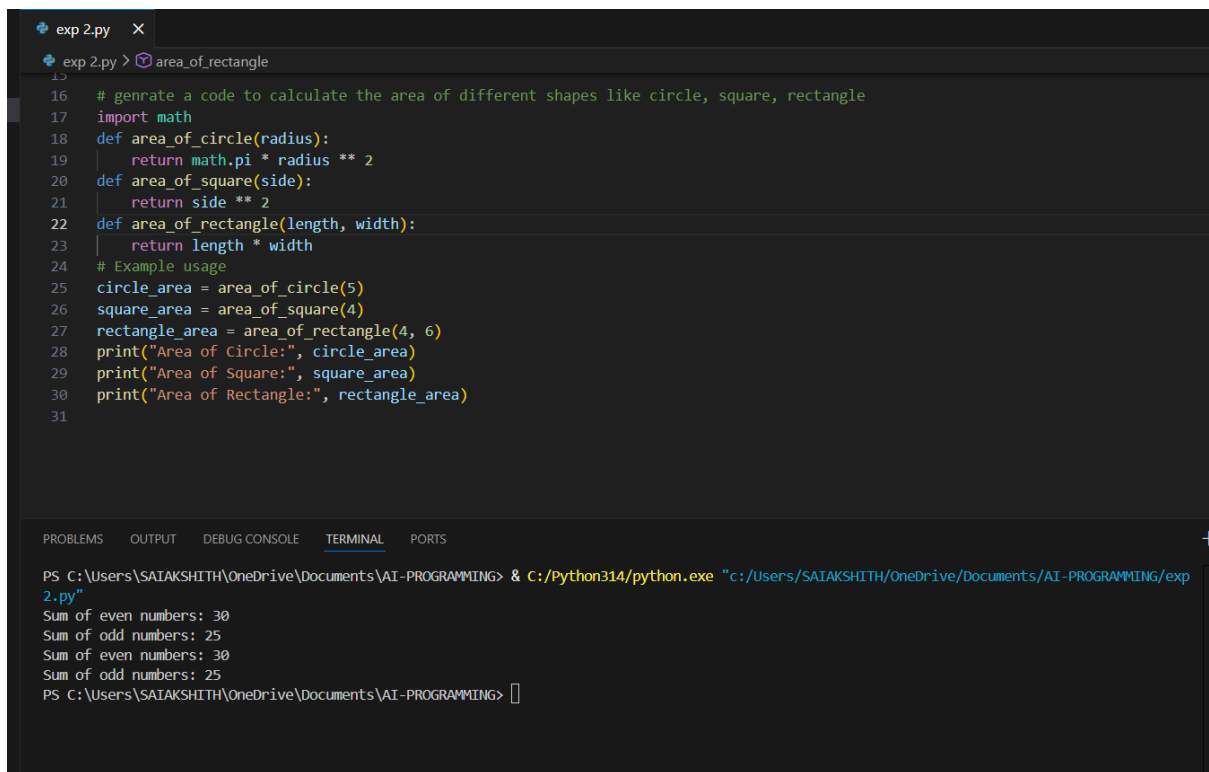
square_area = area_of_square(4)

rectangle_area = area_of_rectangle(4, 6)

print("Area of Circle:", circle_area)

print("Area of Square:", square_area)

print("Area of Rectangle:", rectangle_area)



The screenshot shows a code editor with a file named `exp 2.py`. The code defines three functions: `area_of_circle`, `area_of_square`, and `area_of_rectangle`. It then uses these functions to calculate the areas of a circle with radius 5, a square with side 4, and a rectangle with length 4 and width 6. The terminal output shows the execution of the script, displaying the calculated areas.

```
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
```

Terminal 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>
```

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

PROMPT :

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

CODE:

```
32
33 '''Task 4: Tool Comparison Reflection
34 ❖ Scenario:
35 You must recommend an AI coding tool.
36 ❖ Task:
37 Based on your work in this topic, compare Gemini, Copilot, and Cursor AI
38 for usability and code quality.
39 ❖ Expected Output:
40 Short written reflection'''
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.
```