Assignment 13.3

AI ASSISTANT CODING

2303A51271 batch -08 25-2-26

Task 1: Refactoring – Removing Code Duplication

Objective

To eliminate repeated logic by extracting reusable functions.

Task Description

Use AI assistance to refactor a legacy Python script that contains repeated

blocks of code calculating the area and perimeter of rectangles.

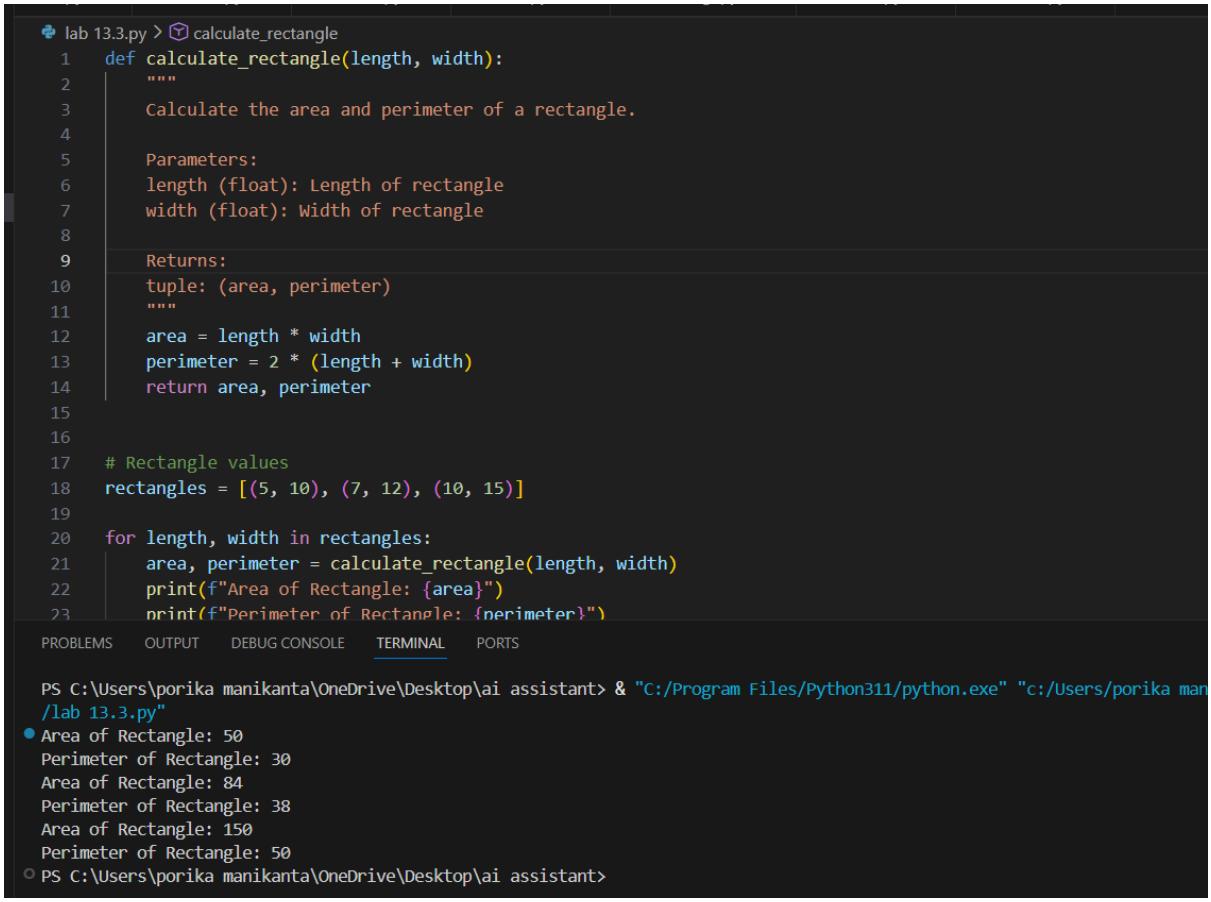
Starter (Legacy) Code

```
# Legacy script with repeated logic
print("Area of Rectangle:", 5 * 10)
print("Perimeter of Rectangle:", 2 * (5 + 10))
print("Area of Rectangle:", 7 * 12)
print("Perimeter of Rectangle:", 2 * (7 + 12))
print("Area of Rectangle:", 10 * 15)
print("Perimeter of Rectangle:", 2 * (10 + 15))
```

Expected Outcome

- A reusable function to calculate area and perimeter

- No duplicated code blocks
- Proper docstrings for all functions



Task 2: Refactoring – Optimizing Loops and Conditionals

Objective

To improve performance by replacing inefficient nested loops with optimized

structures.

Task Description

Use AI to analyze and refactor a script that checks the presence of elements

using nested loops.

Starter (Legacy) Code

```
names = ["Alice", "Bob", "Charlie", "David"]

search_names = ["Charlie", "Eve", "Bob"]

for s in search_names:

found = False

for n in names:

if s == n:

found = True

if found:

print(f"{s} is in the list")

else:

print(f"{s} is not in the list")
```

Expected Outcome

- Optimized solution using set lookups or comprehensions
- Performance comparison before and after refactoring

```python
names = ["Alice", "Bob", "Charlie", "David"]
search_names = ["Charlie", "Eve", "Bob"]

name_set = set(names)  # O(1) lookup time

for name in search_names:
    if name in name_set:
        print(f"{name} is in the list")
    else:
        print(f"{name} is not in the list")
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\porika manikanta\OneDrive\Desktop\ai assistant> & "C:/Program Files/Python311/
/lab 13.3.py"
Charlie is in the list
Eve is not in the list
Bob is in the list
PS C:\Users\porika manikanta\OneDrive\Desktop\ai assistant>
```

Task 3: Refactoring – Extracting Reusable Functions

Objective

To modularize code by extracting calculations into reusable functions.

Task Description

Refactor a legacy script where price and tax calculations are written inline.

Starter (Legacy) Code

price = 250

tax = price * 0.18

total = price + tax

print("Total Price:", total)

price = 500

tax = price * 0.18

total = price + tax

print("Total Price:", total)

Expected Outcome

- A function calculate_total(price)
- Cleaner and reusable code structure
- Proper documentation

```
36  TAX_RATE = 0.18
37
38  def calculate_total(price):
39      """
40      Calculate total price including tax.
41
42      Parameters:
43      price (float): Base price of product
44
45      Returns:
46      float: Total price including tax
47      """
48      tax = price * TAX_RATE
49      total = price + tax
50      return total
51
52
53  prices = [250, 500]
54
55  for price in prices:
56      print("Total Price:", calculate_total(price))
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS C:\Users\porika manikanta\OneDrive\Desktop\ai assistant> & "C:/Program Files/Python311/python.exe" "c:/Users/porika
/lab 13.3.py"
Total Price: 295.0
Total Price: 590.0
PS C:\Users\porika manikanta\OneDrive\Desktop\ai assistant>
```

Task 4: Refactoring – Replacing Hardcoded Values with Constants

Objective

To improve maintainability by replacing magic numbers with named constants.

Task Description

Use AI to identify hardcoded values and replace them with constants.

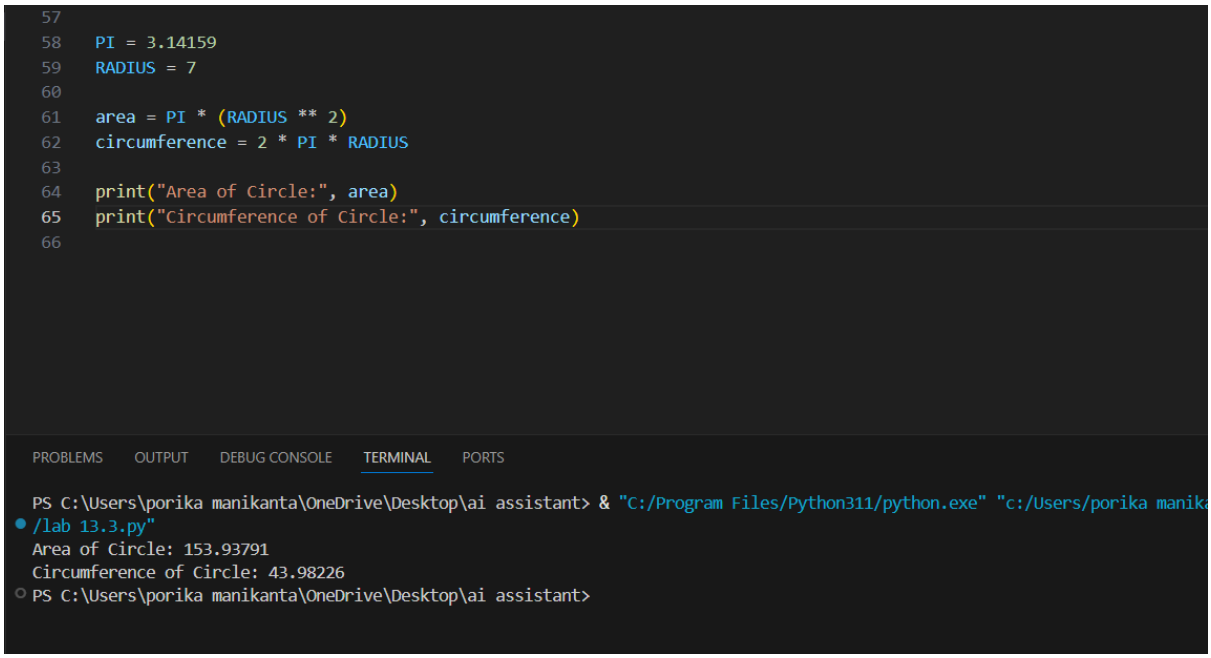
Starter (Legacy) Code

print("Area of Circle:", 3.14159 * (7 ** 2))

print("Circumference of Circle:", 2 * 3.14159 * 7)

Expected Outcome

- Constants such as PI and RADIUS
- Cleaner, maintainable code



Task 5: Refactoring – Improving Variable Naming and Readability

Objective

To enhance readability using descriptive variable names and comments.

Task Description

Refactor a script with unclear variable names.

Starter (Legacy) Code

```
a = 10
b = 20
c = a * b / 2
print(c)
```

Expected Outcome

- Descriptive variable names
- Inline comments explaining logic
- Identical output

Note: Report should be submitted as a word document for all tasks in a single

document with prompts, comments & code explanation, and output and if

required, screenshots.

```
68  base = 10
69  height = 20
70
71  # Calculate area of triangle
72  area_of_triangle = (base * height) / 2
73
74  print(area_of_triangle)
75
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\porika manikanta\OneDrive\Desktop\ai assistant> & "C:/Program Files/Python311/pyth
/lab 13.3.py"
100.0
PS C:\Users\porika manikanta\OneDrive\Desktop\ai assistant>
```