

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

```

lab 13.3.py > calculate_rectangle
1 def calculate_rectangle(length, width):
2     """
3         Calculate the area and perimeter of a rectangle.
4
5     Parameters:
6         length (float): Length of rectangle
7         width (float): Width of rectangle
8
9     Returns:
10        tuple: (area, perimeter)
11        """
12        area = length * width
13        perimeter = 2 * (length + width)
14        return area, perimeter
15
16
17 # Rectangle values
18 rectangles = [(5, 10), (7, 12), (10, 15)]
19
20 for length, width in rectangles:
21     area, perimeter = calculate_rectangle(length, width)
22     print(f"Area of Rectangle: {area}")
23     print(f"Perimeter of Rectangle: {perimeter}")

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```

PS C:\Users\porika manikanta\OneDrive\Desktop\ai assistant> & "C:/Program Files/Python311/python.exe" "c:/Users/porika manikanta\OneDrive\Desktop\ai assistant> /lab 13.3.py"
● Area of Rectangle: 50
Perimeter of Rectangle: 30
Area of Rectangle: 84
Perimeter of Rectangle: 38
Area of Rectangle: 150
Perimeter of Rectangle: 50
○ PS C:\Users\porika manikanta\OneDrive\Desktop\ai assistant>

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

```
24
25     names = ["Alice", "Bob", "Charlie", "David"]
26     search_names = ["Charlie", "Eve", "Bob"]
27
28     name_set = set(names) # O(1) lookup time
29
30     for name in search_names:
31         if name in name_set:
32             print(f"{name} is in the list")
33         else:
34             print(f"{name} is not in the list")
35
```

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

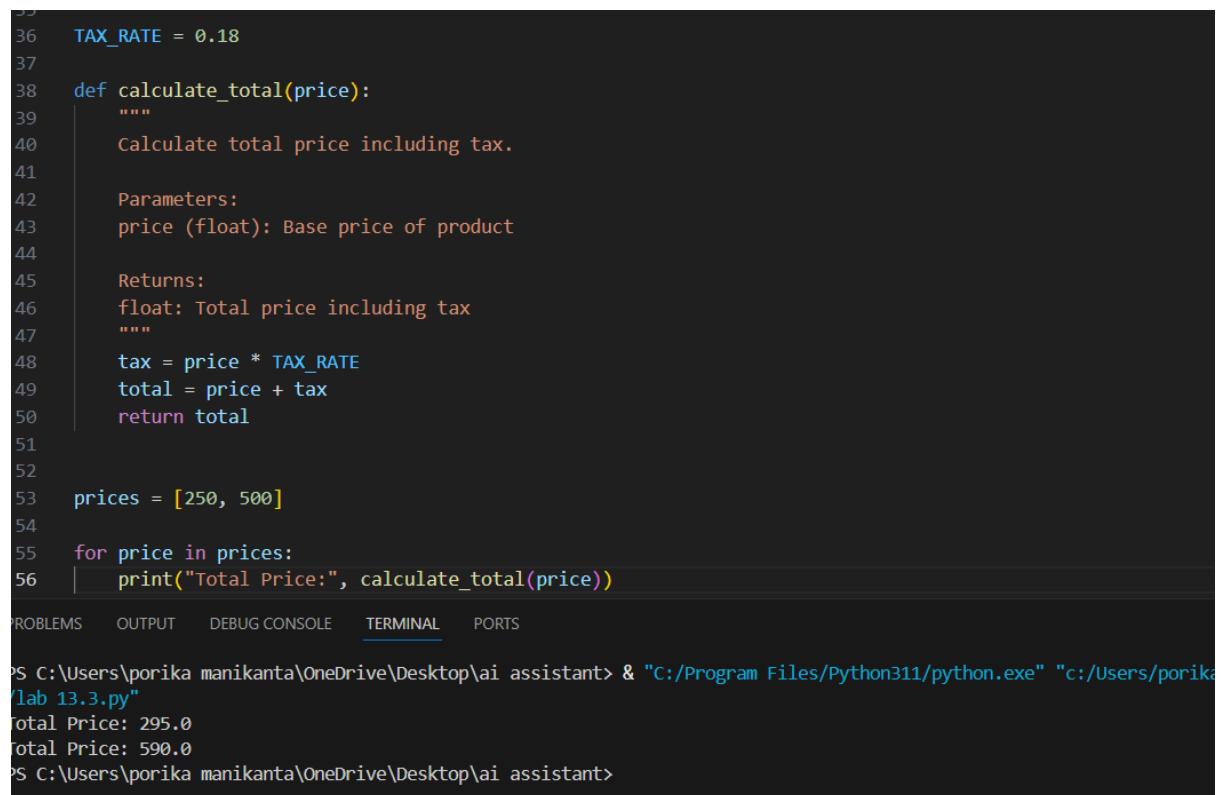
```

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



```

35
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

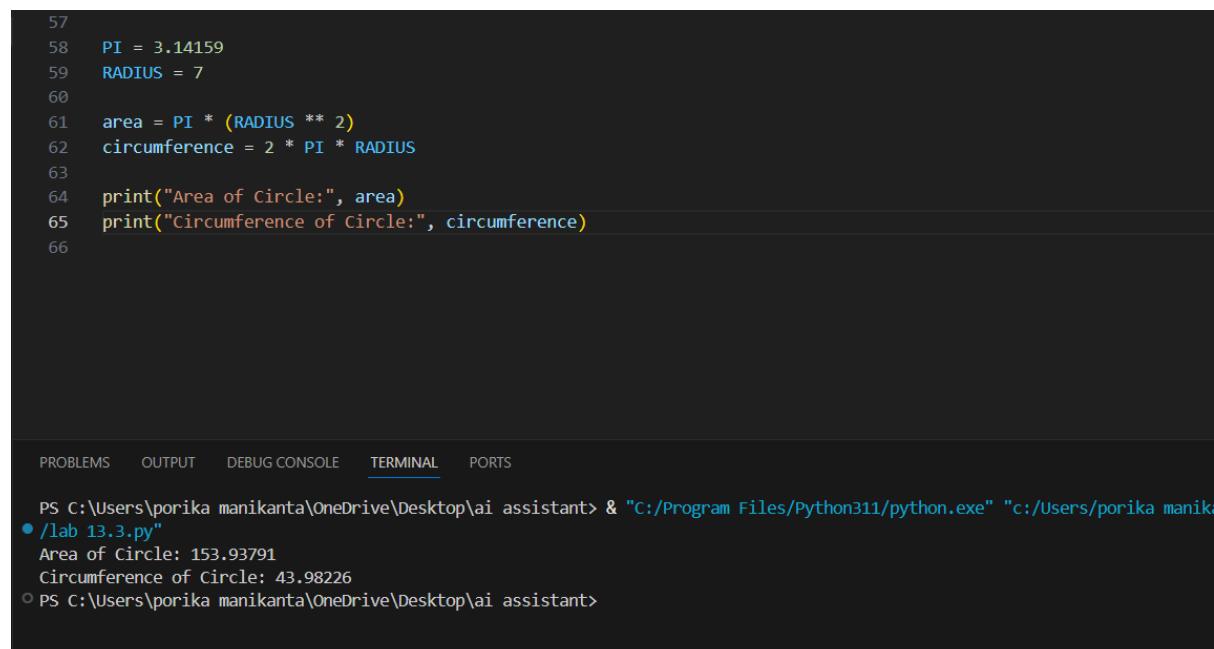
Use AI to identify hardcoded values and replace them with constants.

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



The screenshot shows a terminal window with the following content:

```
57
58     PI = 3.14159
59     RADIUS = 7
60
61     area = PI * (RADIUS ** 2)
62     circumference = 2 * PI * RADIUS
63
64     print("Area of Circle:", area)
65     print("Circumference of Circle:", circumference)
66
```

Below the code, the terminal shows the output of running the script:

```
PS C:\Users\porika.manikanta\Desktop\ai assistant> & "C:/Program Files/Python311/python.exe" "c:/Users/porika.manika
● /Lab 13.3.py"
Area of Circle: 153.93791
Circumference of Circle: 43.98226
○ PS C:\Users\porika.manikanta\Desktop\ai assistant>
```

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.

```
6/  
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/python /lab 13.3.py"  
100.0  
PS C:\Users\porika manikanta\OneDrive\Desktop\ai assistant>
```