

ASSIGNMENT -2.5

NAME : AKSHITHA

2303A51360

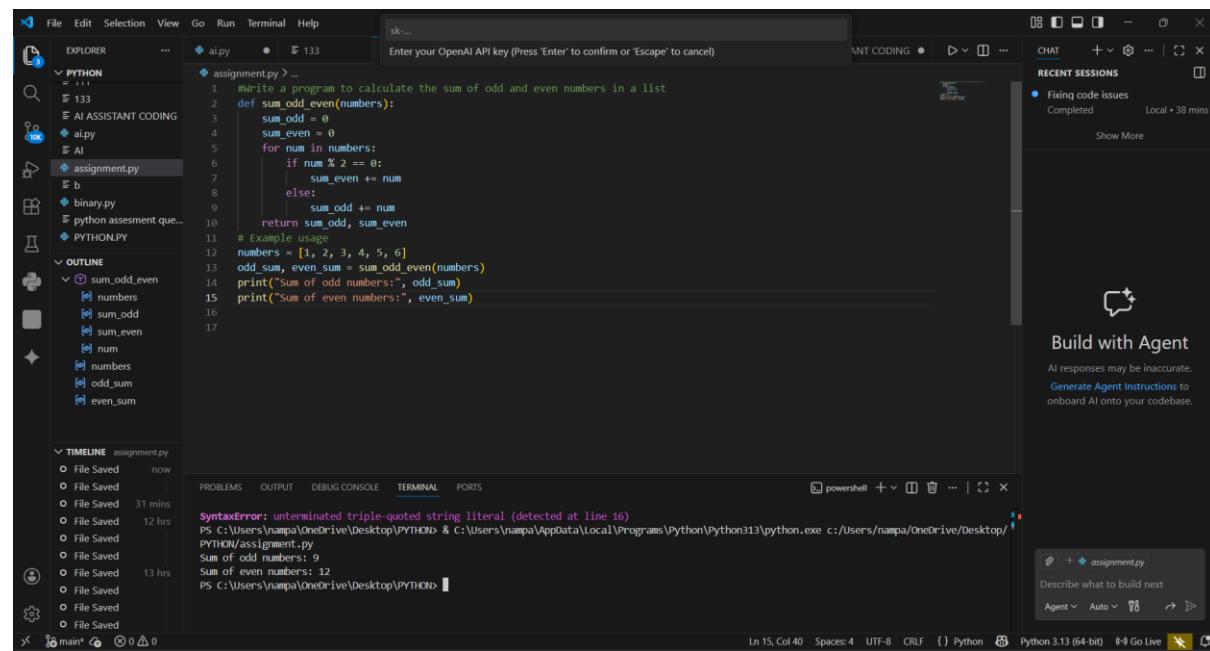
BATCH:29

TASK 1:

PROMPT:

WRITE A PROGRAM TO CALCULATE THE SUM OF ODD AND EVEN NUMBERS IN A LIST

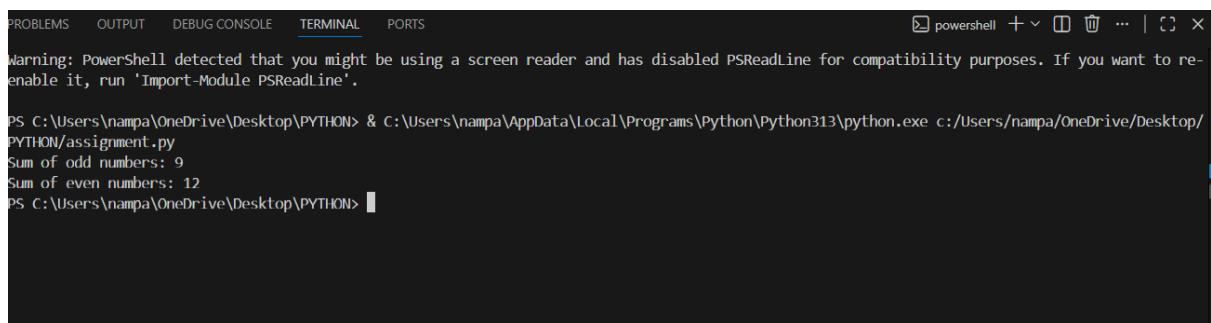
CODE:



The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER:** Shows files in the workspace, including `aipy`, `133`, `AI ASSISTANT CODING`, `ai.py`, `AI`, `assignment.py`, `b`, `binary.py`, `python assment que...`, and `PYTHON.PY`. The `assignment.py` file is currently selected.
- EDITOR:** Displays the Python code for calculating the sum of odd and even numbers in a list. The code uses a function `sum_odd_even` to iterate through a list of numbers, summing odd and even numbers separately.
- OUTPUT:** Shows a terminal window with a `SyntaxError` message indicating an unterminated triple-quoted string literal at line 16. The command run was `python assignment.py`.
- STATUS BAR:** Shows the file path as `C:\Users\nampa\OneDrive\Desktop\PYTHON> & C:\Users\nampa\AppData\Local\Programs\Python\Python313\python.exe c:/Users/nampa/OneDrive/Desktop/PYTHON>`, line 15, col 40, and other system information.

OUTPUT:



A screenshot of a terminal window titled "powershell". The window shows the following command and its output:

```
PS C:\Users\nampa\OneDrive\Desktop\PYTHON> & C:\Users\nampa\AppData\Local\Programs\Python\Python313\python.exe c:/Users/nampa/OneDrive/Desktop/PYTHON/assignment.py
Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purposes. If you want to re-enable it, run 'Import-Module PSReadLine'.
Sum of odd numbers: 9
Sum of even numbers: 12
PS C:\Users\nampa\OneDrive\Desktop\PYTHON>
```

OBSERVATION:

The original code works correctly but is written as a single block, making it harder to reuse and test. The refactored (AI-improved) code separates logic into a function, improving:

- Readability
- Reusability
- Maintainability

Using a function allows the same logic to be reused with different lists without rewriting code.

TASK:2

PROMPT:

WRITE A PROGRAM EXPLAIN A FUNCTION THAT CALCULATES THE AREA OF DIFFERENT SHAPES. THE CODE MUST INCLUDE PROPER COMMENTS FOR EXPLANATION.

CODE:

```

File Edit Selection View Go Run Terminal Help
EXPLORER ... a.py 133
PYTHON
assignment.py > calculate_area
1 # write a program explain a function that calculates the area of different shapes.
2 # The code must include proper comments for explanation.
3
4 def calculate_area(shape, value1, value2=0):
5     """
6         This function calculates the area of different shapes.
7         shape -> type of shape (circle, rectangle, triangle)
8         value1 -> radius (circle) OR length/base
9         value2 -> width/height (default is 0 for circle)
10        """
11
12        # Check if the shape is a circle
13        if shape == "circle":
14            return 3.14 * value1 * value1 # π * r * r
15
16        # Check if the shape is a rectangle
17        elif shape == "rectangle":
18            return value1 * value2 # length * width
19
20        # Check if the shape is a triangle
21        elif shape == "triangle":
22            return 0.5 * value1 * value2 # ½ * base * height
23
24        # If shape is not supported
25        else:
26            return "Invalid shape"
27
28 def main():
29     # Take shape input from the user
30     shape = input("Enter shape (circle/rectangle/triangle): ").lower()
31
32     # If shape is circle, ask only for radius
33     if shape == "circle":
34         radius = float(input("Enter radius: "))
35         area = calculate_area(shape, radius)
36
37
TIMELINE assignment.py
File Saved now
File Saved 2 mins
File Saved 3 mins
File Saved 11 mins
File Saved 42 mins
File Saved 13 hrs
File Saved 36
File Saved 37

```

Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.

OUTPUT:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell + v ⌂ ... | ⌂ x
Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purposes. If you want to re-enable it, run 'Import-Module PSReadLine'.
PS C:\Users\nampa\OneDrive\Desktop\PYTHON> & C:\Users\nampa\AppData\Local\Programs\Python\Python313\python.exe c:/Users/nampa/Desktop/PYTHON/assignment.py
> Enter shape (circle/rectangle/triangle): circle
> Enter radius: 24
> Area: 1808.639999999999
PS C:\Users\nampa\OneDrive\Desktop\PYTHON>

```

OBSERVATION:

This program uses one function to calculate the area of multiple shapes, which avoids code duplication.

The shape parameter decides which formula to apply.

The function uses conditional statements (if /elif) to select the correct formula.

It improves code clarity, making onboarding easier and faster.

TASK:3

PROMPT:

EXPLAIN A FUNCTION THAT CALCULATES THE AREA OF DIFFERENT SHAPES (CURSER USED)

SHAPES. WRITE A PROGRAM TO FIND THE SUM OF EVEN AND ODD NUMBERS IN A LIST

CODE:

The screenshot shows a code editor interface with the following details:

- File Explorer:** Shows files like assignment.py, ai.py, binary.py, and PYTHON.PY.
- Code Editor:** Displays the following Python script:

```
numbers = list(map(int, input("Enter numbers: ").split()))
even_sum = 0
odd_sum = 0

for num in numbers:
    if num % 2 == 0:
        even_sum += num
    else:
        odd_sum += num

print("Even Sum:", even_sum)
print("Odd Sum:", odd_sum)
```
- Terminal:** Shows the command run in PowerShell: `PS C:\Users\nampa\OneDrive\Desktop\PYTHON> & C:\Users\nampa\AppData\Local\Programs\Python\Python313\python.exe c:/Users/nampa/OneDrive/Desktop/PYTHON/assignment.py`. The output is:
 - Enter numbers: 26
 - Even Sum: 26
 - Odd Sum: 0
- Timeline:** Shows a history of file saves and edits.
- RECENT SESSIONS:** Shows a session titled "Fixing code issues" completed 55 mins ago.
- Build with Agent:** A sidebar with a message about AI responses being inaccurate and a button to "Generate Agent Instructions".

OBSERVATION:

The program demonstrates how one function can handle multiple use cases. Comments clearly explain:

What the function does

Why each condition exists

What each parameter represents

Using comments makes the code junior-developer friendly, which is ideal for onboarding.

The main () function separates user interaction from business logic, improving structure.

This style is considered clean, readable, and professional in real-world projects

TASK:4

PROMPT:

BASED ON PRACTICAL USAGE AND EXPERIMENTATION, COMPARE GEMINI, GITHUB COPILOT, AND CURSOR AI IN TERMS OF USABILITY AND CODE QUALITY. OBSERVATION:

GEMINI Is best suited for explanations and learning support. It produces readable, beginner-friendly code and clear step-by-step reasoning, making it ideal for onboarding juniors and understanding concepts.

GitHub Copilot excels in real-time coding assistance inside IDEs. It is fast, context aware, and highly productive for experienced developers, but its code may lack explanations.

Cursor AI stands out for **prompt sensitivity and refactoring quality**. It responds strongly to detailed prompts, generating cleaner, more structured, and optimized code, making it suitable for improving legacy codebases.

usability, Copilot integrates seamlessly into workflows, Gemini is conversational and educational, and Cursor AI offers powerful prompt-driven refactoring.

code quality, Cursor AI and Copilot generally produce more professional, production ready code, while Gemini focuses on clarity over optimization