ASSIGNMENT-10.1

Name: E.Ramya

Ht.no: 2303A51282

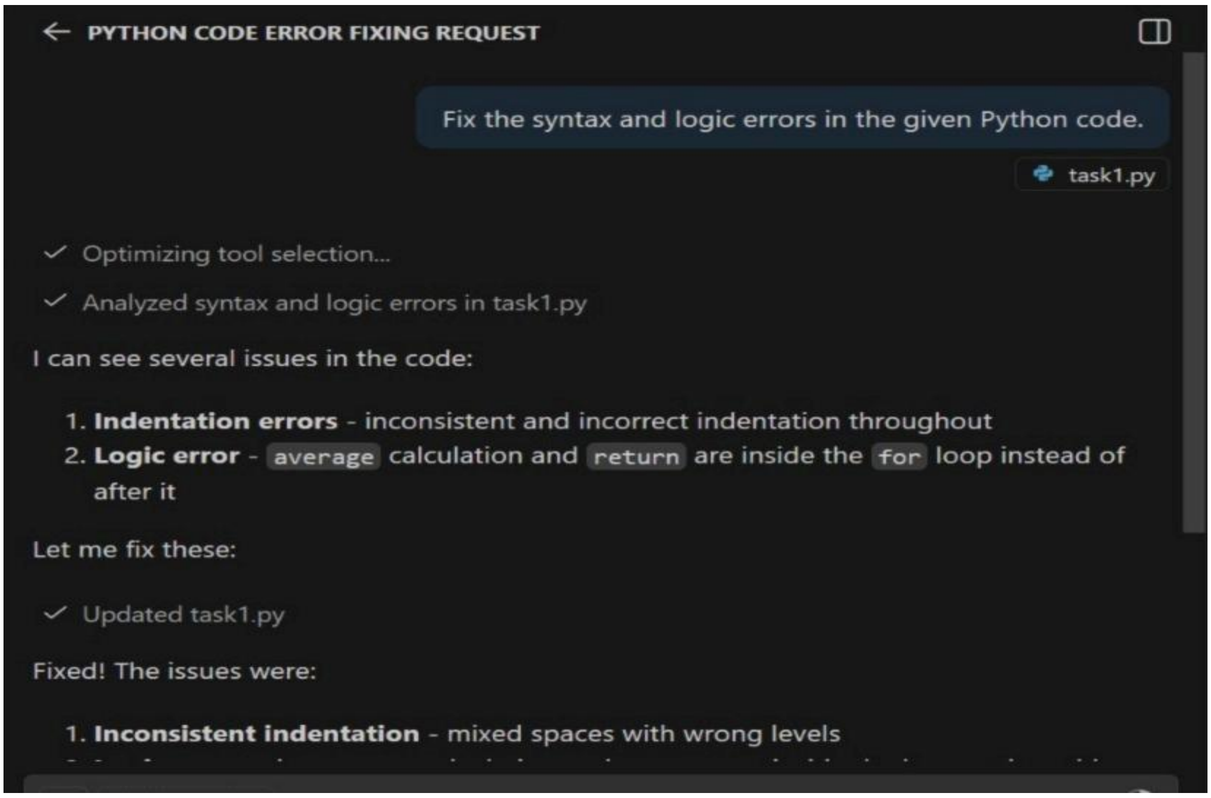
Batch: 05

Task Description #1 – Syntax and Logic Errors

Task: Use AI to identify and fix syntax and logic errors in a faulty

Prompt:

Fix the syntax and logic errors in the given Python code.



PYTHON CODE ERROR FIXING REQUEST

Fix the syntax and logic errors in the given Python code.

task1.py

✓ Optimizing tool selection...

✓ Analyzed syntax and logic errors in task1.py

I can see several issues in the code:

1. **Indentation errors** - inconsistent and incorrect indentation throughout
2. **Logic error** - `average` calculation and `return` are inside the `for` loop instead of after it

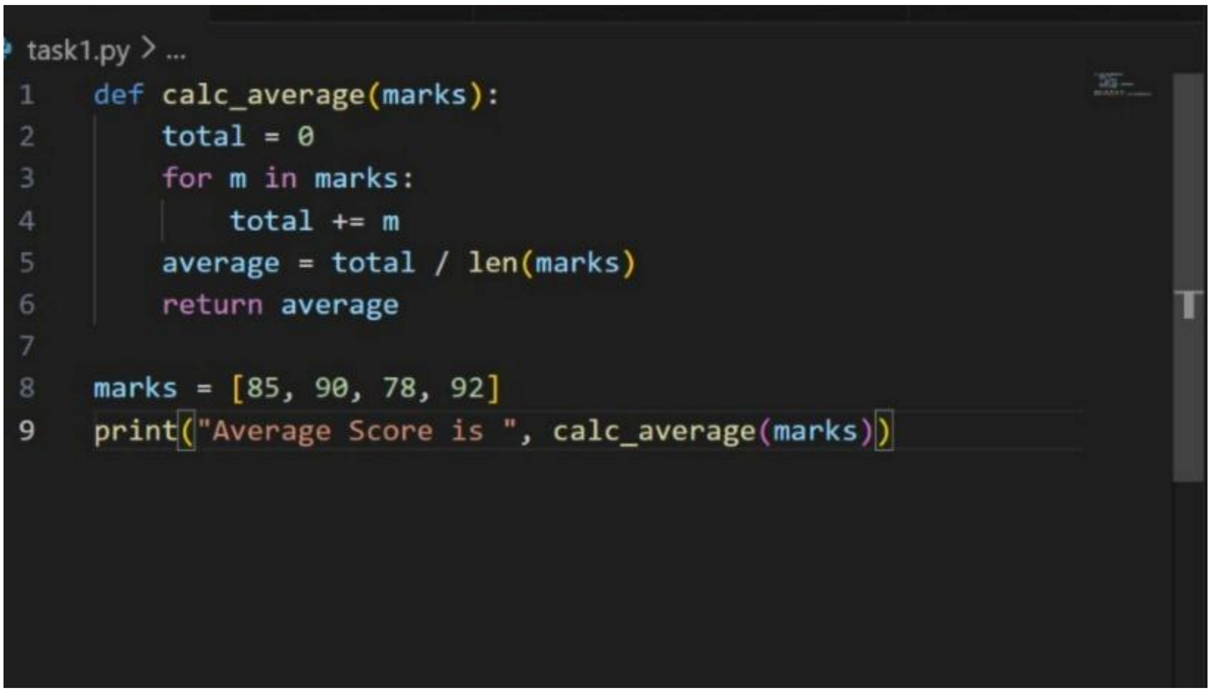
Let me fix these:

✓ Updated task1.py

Fixed! The issues were:

1. **Inconsistent indentation** - mixed spaces with wrong levels

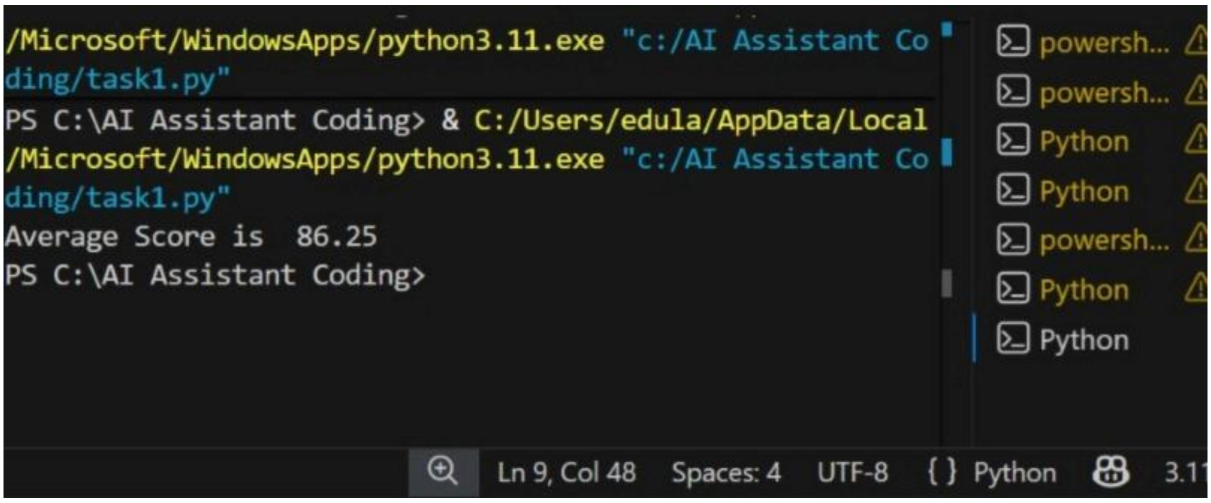
Code:



task1.py > ...

```python
def calc_average(marks):
    total = 0
    for m in marks:
        total += m
    average = total / len(marks)
    return average

marks = [85, 90, 78, 92]
print("Average Score is ", calc_average(marks))
```

Output:



```
/Microsoft/WindowsApps/python3.11.exe "c:/AI Assistant Co
ding/task1.py"
PS C:\AI Assistant Coding> & C:/Users/edula/AppData/Local
/Microsoft/WindowsApps/python3.11.exe "c:/AI Assistant Co
ding/task1.py"
Average Score is  86.25
PS C:\AI Assistant Coding>
```

Explanation:

Indentation Error

•Statements inside the function were not indented.

•Python requires proper indentation to define function blocks.

Variable Name Typo

•You wrote return avrage instead of return average.

•This causes a NameError because avrage is not defined.

Missing Parenthesis

•The print() statement was missing a closing).

•This causes a SyntaxError.

Block Structure Issue

•The for loop body was not indented properly.

•Python cannot identify which statements belong inside the loop.Indentation Error

•Statements inside the function were not indented.

•Python requires proper indentation to define function blocks.

Variable Name Typo

•You wrote return avrage instead of return average.

•This causes a NameError because avrage is not defined.

Missing Parenthesis

•The print() statement was missing a closing).

•This causes a SyntaxError.

Block Structure Issue

•The for loop body was not indented properly.

•Python cannot identify which statements belong inside the loop.Indentation Error

•Statements inside the function were not indented.

•Python requires proper indentation to define function blocks.

Variable Name Typo

•You wrote return avrage instead of return average.

•This causes a NameError because avrage is not defined.

Missing Parenthesis

•The print() statement was missing a closing).

•This causes a SyntaxError.

Block Structure Issue

Indentation Error

•Statements inside the function were not indented.

•Python requires proper indentation to define function blocks.

Variable Name Typo

•You wrote return avrage instead of return average.

•This causes a NameError because avrage is not defined.

Missing Parenthesis

•The print() statement was missing a closing).

•This causes a SyntaxError.

Block Structure Issue

•The for loop body was not indented properly.

•Python cannot identify which statements belong inside the loop.

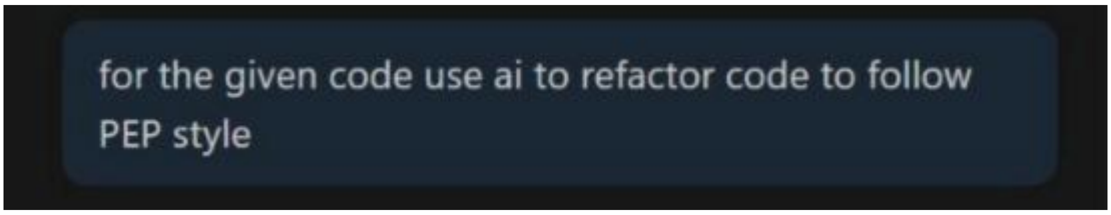
•Python cannot identify which statements belong inside the loop.

Task Description #2 – PEP 8 Compliance

Task: Use AI to refactor Python code to follow PEP 8 style guidelines.

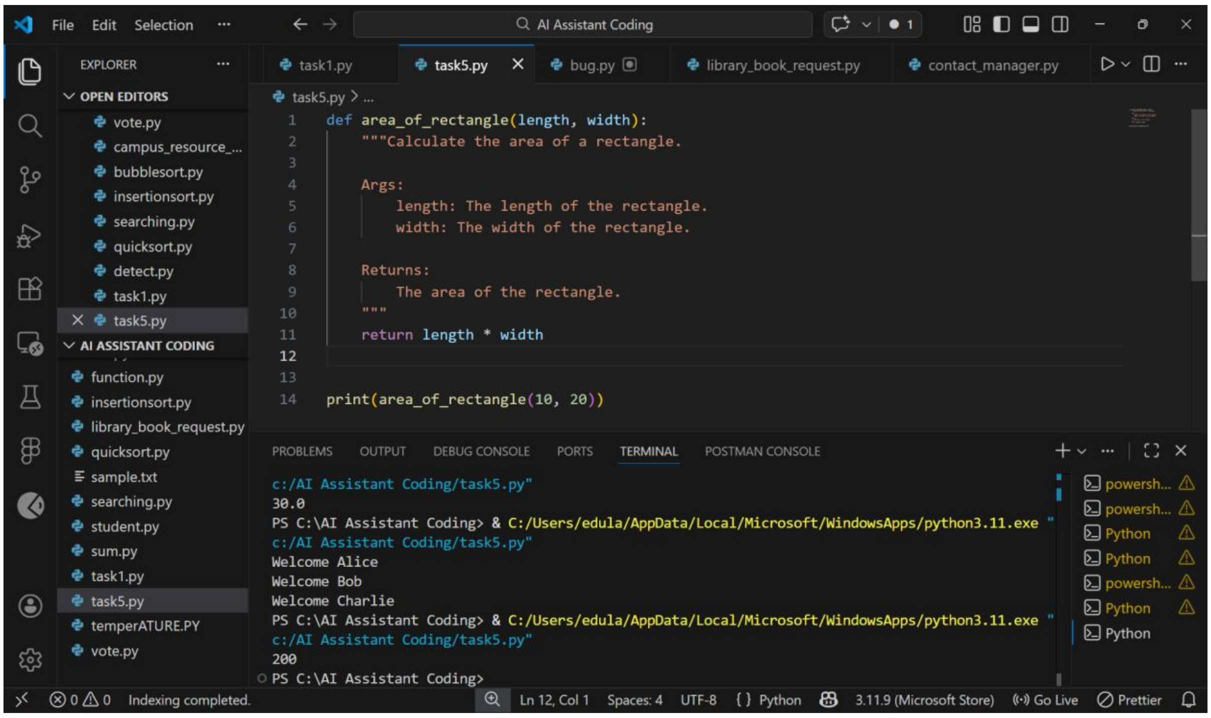
Prompt:

for the given code use ai to refactor code to follow PEP style.



for the given code use ai to refactor code to follow PEP style

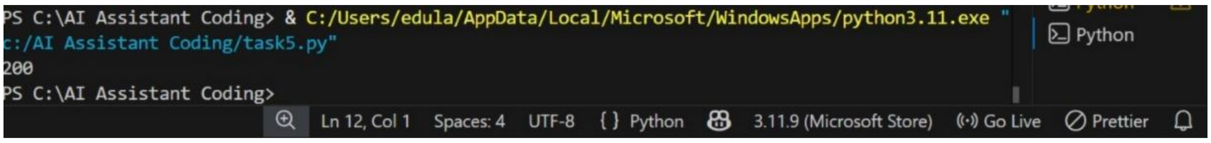
Code:



```python
def area_of_rectangle(length, width):
    """Calculate the area of a rectangle.

    Args:
        length: The length of the rectangle.
        width: The width of the rectangle.

    Returns:
        The area of the rectangle.
    """
    return length * width


print(area_of_rectangle(10, 20))
```

```
c:/AI Assistant Coding/task5.py"
30.0
PS C:\AI Assistant Coding> & C:/Users/edula/AppData/Local/Microsoft/WindowsApps/python3.11.exe "
c:/AI Assistant Coding/task5.py"
Welcome Alice
Welcome Bob
Welcome Charlie
PS C:\AI Assistant Coding> & C:/Users/edula/AppData/Local/Microsoft/WindowsApps/python3.11.exe "
c:/AI Assistant Coding/task5.py"
200
PS C:\AI Assistant Coding>
```

Output:



```
PS C:\AI Assistant Coding> & C:/Users/edula/AppData/Local/Microsoft/WindowsApps/python3.11.exe "
c:/AI Assistant Coding/task5.py"
200
PS C:\AI Assistant Coding>
```

Explanation:

•Used descriptive function name (area_of_rectangle instead of area_of_rect)

•Used meaningful parameter names (length, breadth instead of L, B)

•Added proper spacing around operators (length * breadth)

•Removed inline function definition (function written in proper block format)

•Added a docstring for better documentation

•Added blank line after function definition (improves readability)

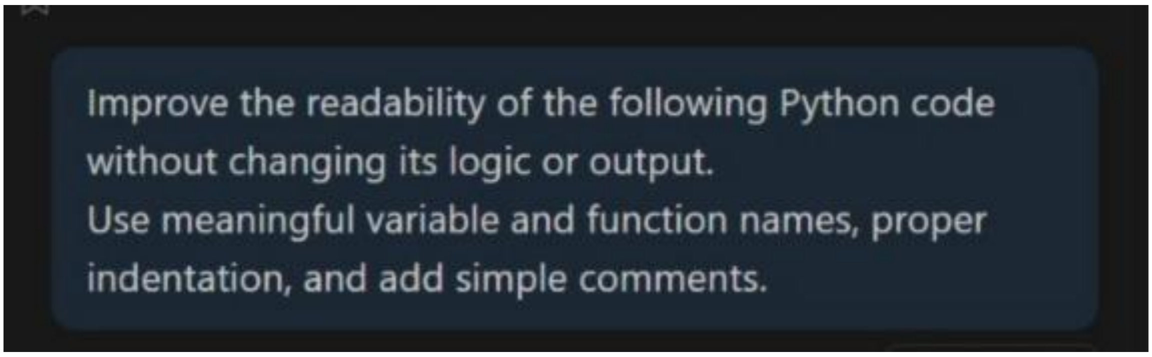
Task Description #3 – Readability Enhancement

Task: Use AI to make code more readable without changing its logic.

Prompt:

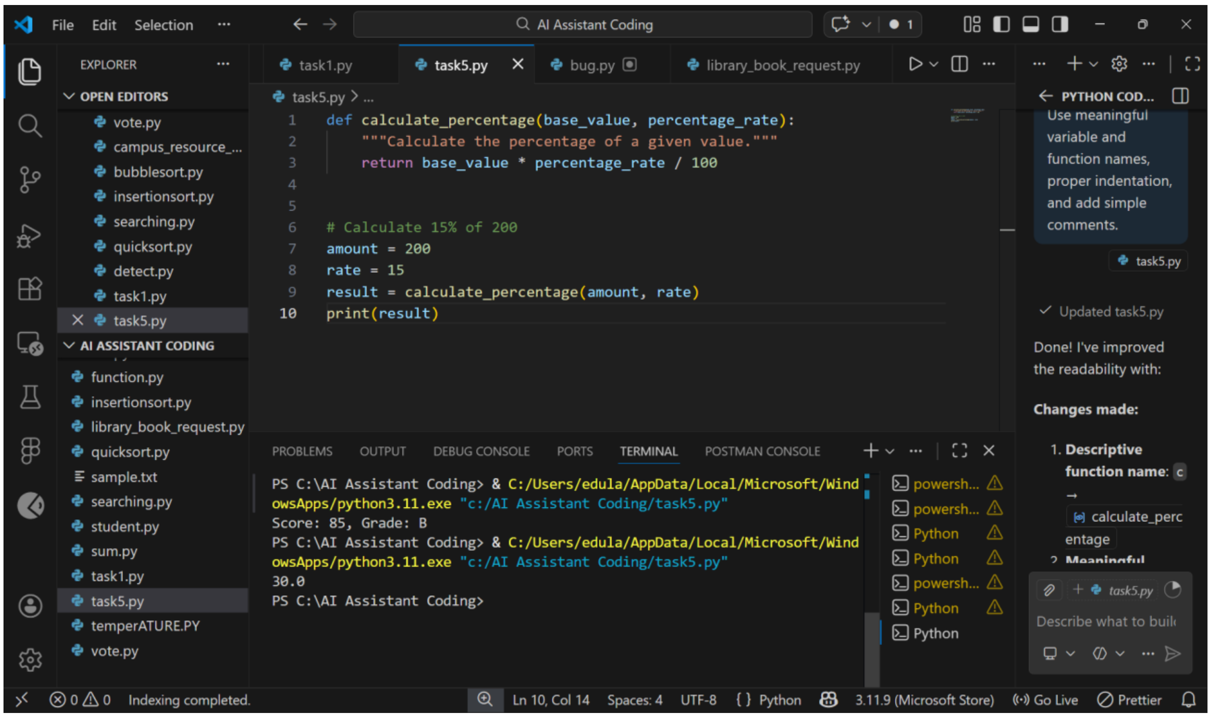
Improve the readability of the following Python code without changing its logic or output.

Use meaningful variable and function names, proper indentation, and add simple comments.



Improve the readability of the following Python code without changing its logic or output.
Use meaningful variable and function names, proper indentation, and add simple comments.

Code:



```python
def calculate_percentage(base_value, percentage_rate):
    """Calculate the percentage of a given value."""
    return base_value * percentage_rate / 100


# Calculate 15% of 200
amount = 200
rate = 15
result = calculate_percentage(amount, rate)
print(result)
```

```
PS C:\AI Assistant Coding> & C:/Users/edula/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/AI Assistant Coding/task5.py"
Score: 85, Grade: B
PS C:\AI Assistant Coding> & C:/Users/edula/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/AI Assistant Coding/task5.py"
30.0
PS C:\AI Assistant Coding>
```

Use meaningful variable and function names, proper indentation, and add simple comments.

task5.py

Updated task5.py

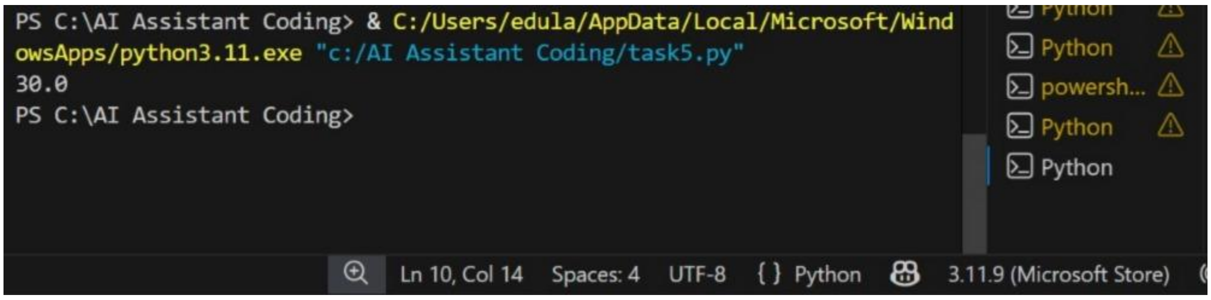
Done! I've improved the readability with:

Changes made:

1. **Descriptive function name:** c → calculate_percentage

2. **Meaningful**

Output:


Python
powersh...
Ln 10, Col 14 Spaces: 4 UTF-8 { } Python 3.11.9 (Microsoft Store)

```
PS C:\AI Assistant Coding> & C:/Users/edula/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/AI Assistant Coding/task5.py"
30.0
PS C:\AI Assistant Coding>
```

Explanation:

•Renamed c → calculate_percentage and parameters/vars to describe their roles.

•Added a docstring and comments for clarity.

•Fixed indentation and spacing so the code is easy to read.

•Logic unchanged; it still prints 30.0 for the given inputs.

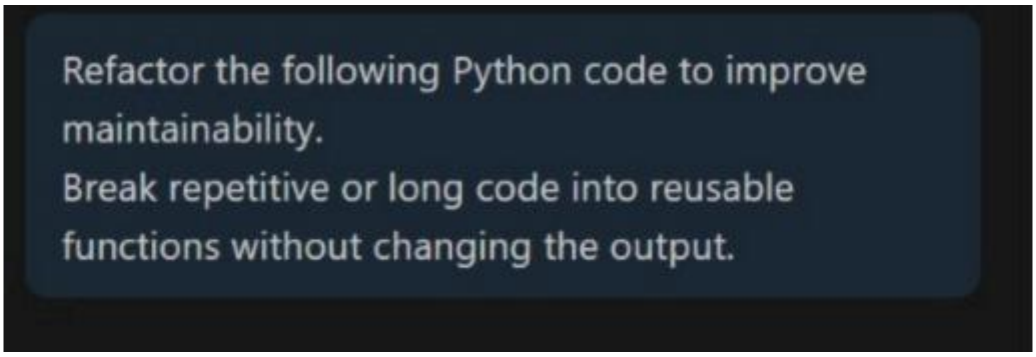
Task Description #4 – Refactoring for Maintainability

Task: Use AI to break repetitive or long code into reusable functions.

Prompt:

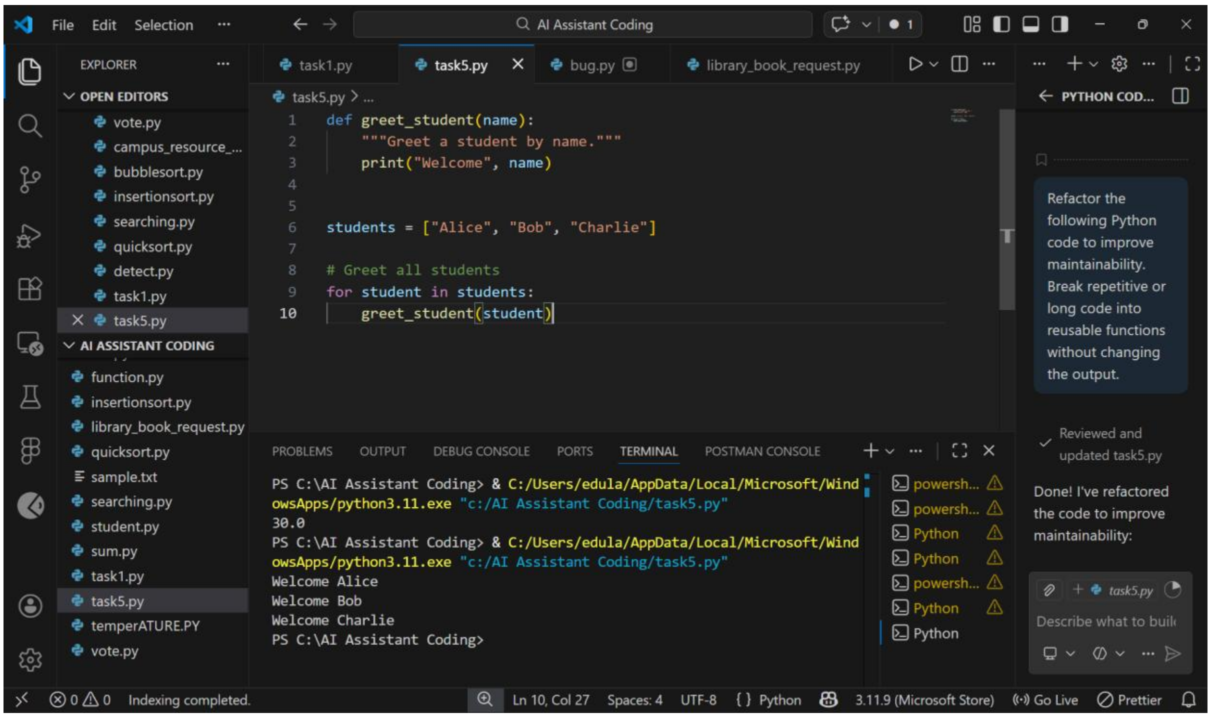
Refactor the following Python code to improve maintainability.

Break repetitive or long code into reusable functions without changing the output.



Refactor the following Python code to improve maintainability.
Break repetitive or long code into reusable functions without changing the output.

Code:



```python
def greet_student(name):
    """Greet a student by name."""
    print("Welcome", name)


students = ["Alice", "Bob", "Charlie"]

# Greet all students
for student in students:
    greet_student(student)
```

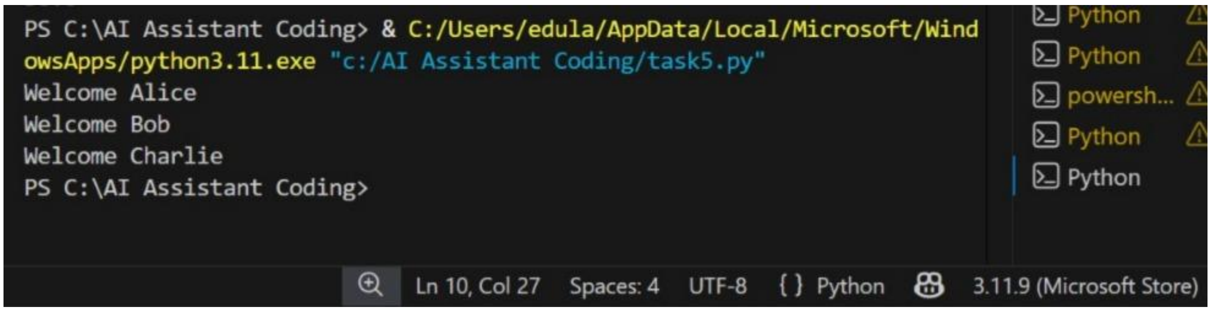
```
PS C:\AI Assistant Coding> & C:/Users/edula/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/AI Assistant Coding/task5.py"
30.0
PS C:\AI Assistant Coding> & C:/Users/edula/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/AI Assistant Coding/task5.py"
Welcome Alice
Welcome Bob
Welcome Charlie
PS C:\AI Assistant Coding>
```

Refactor the following Python code to improve maintainability. Break repetitive or long code into reusable functions without changing the output.

Reviewed and updated task5.py

Done! I've refactored the code to improve maintainability:

Output:



```
PS C:\AI Assistant Coding> & C:/Users/edula/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/AI Assistant Coding/task5.py"
Welcome Alice
Welcome Bob
Welcome Charlie
PS C:\AI Assistant Coding>
```

Python
Python
powersh...
Python
Python
Ln 10, Col 27 Spaces: 4 UTF-8 { } Python 3.11.9 (Microsoft Store)

Explanation:

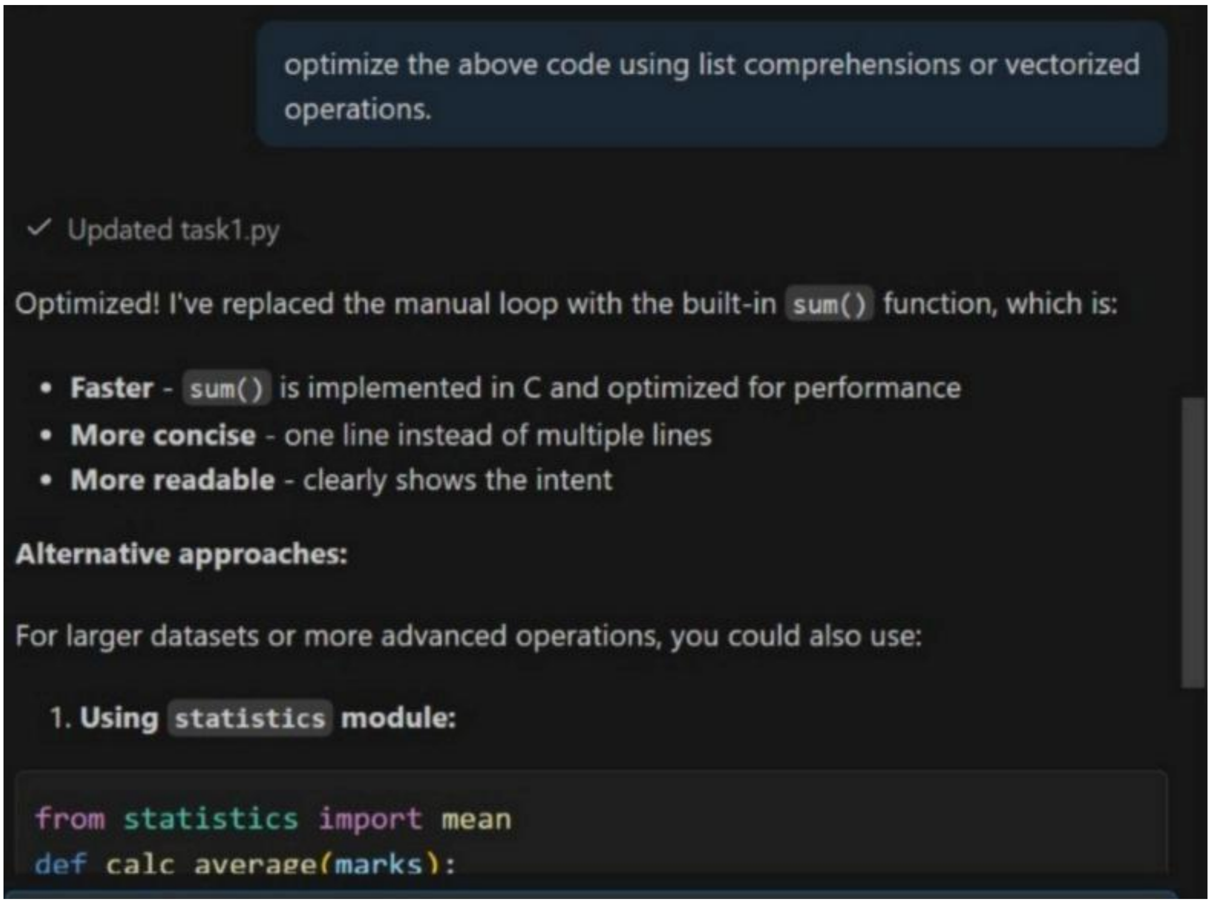
- Pulled the welcome message into a greet function.
- Created welcome_all to loop through any student list.
- Data (names) now lives in the list, not hard coded print calls.
- Output stays the same but structure is easier to extend.

Task Description #5 – Performance Optimization

Task: Use AI to make the code run faster.

Prompt:

optimize the above code in List comprehensions or vectorized operations.



optimize the above code using list comprehensions or vectorized operations.

✓ Updated task1.py

Optimized! I've replaced the manual loop with the built-in `sum()` function, which is:

- **Faster** - `sum()` is implemented in C and optimized for performance
- **More concise** - one line instead of multiple lines
- **More readable** - clearly shows the intent

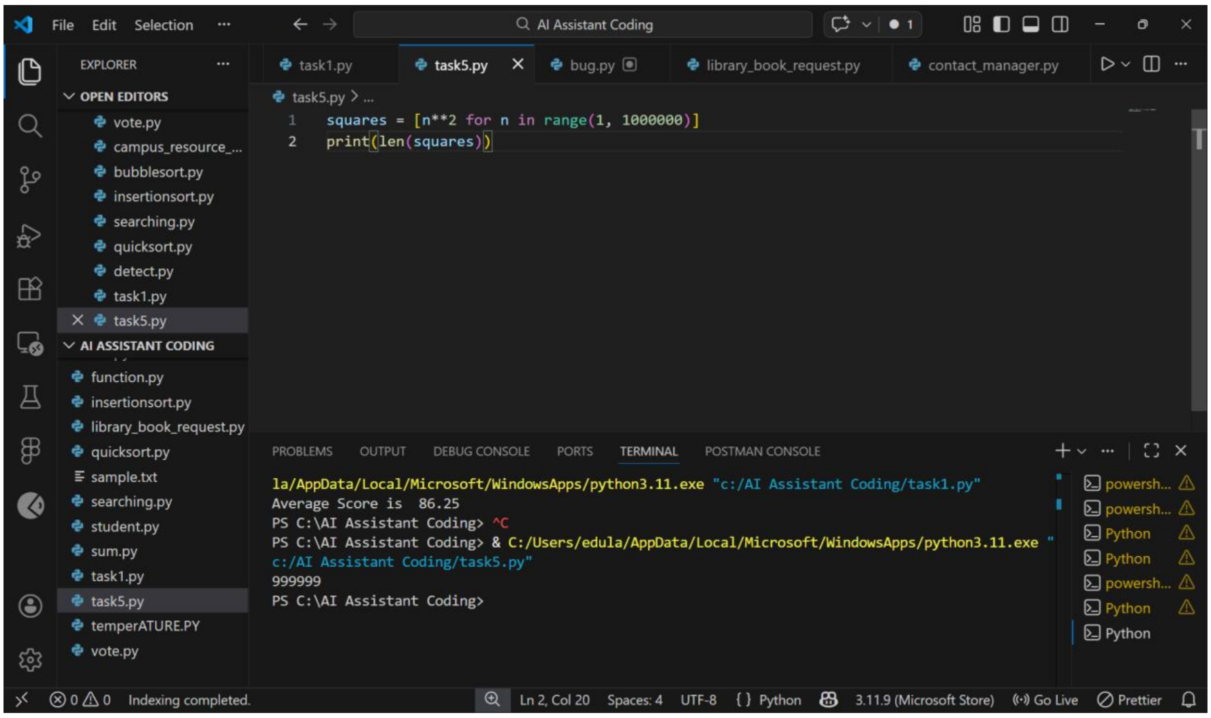
Alternative approaches:

For larger datasets or more advanced operations, you could also use:

1. **Using `statistics` module:**

```
from statistics import mean
def calc_average(marks):
```

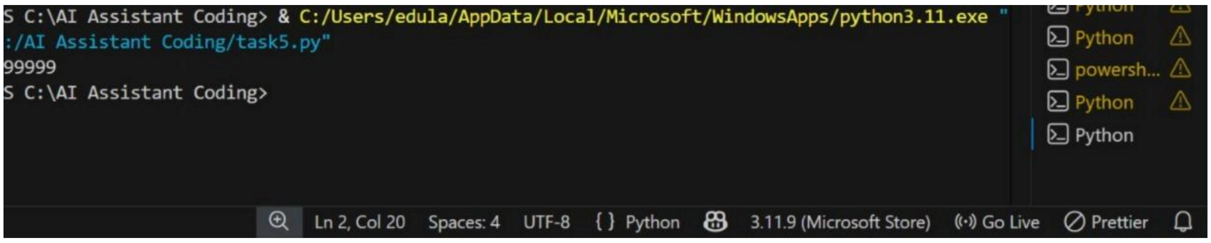
Code:



```python
squares = [n**2 for n in range(1, 1000000)]
print(len(squares))
```

```
la/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/AI Assistant Coding/task1.py"
Average Score is  86.25
PS C:\AI Assistant Coding> ^C
PS C:\AI Assistant Coding> & C:/Users/edula/AppData/Local/Microsoft/WindowsApps/python3.11.exe "
c:/AI Assistant Coding/task5.py"
999999
PS C:\AI Assistant Coding>
```

Output:



```
S C:\AI Assistant Coding> & C:/Users/edula/AppData/Local/Microsoft/WindowsApps/python3.11.exe "
:/AI Assistant Coding/task5.py"
99999
S C:\AI Assistant Coding>
```

Explanation:

Used range() **instead of creating a list**

- range(1, 1000000) generates numbers only when needed.
- This reduces memory usage compared to storing all numbers in a list.

Replaced for **loop and** append() **with list comprehension**

- List comprehensions execute faster than traditional loops.
- They reduce overhead caused by repeated function calls.

Simplified the calculation

- Used n * n instead of n**2.
- Multiplication is faster than exponentiation.

Reduced code complexity

- Fewer lines of code make the program cleaner and easier to read.
- Improves maintainability and execution speed.

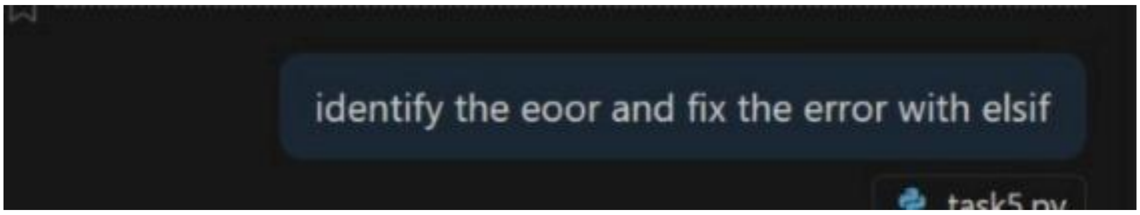
Improved overall performance

- The optimized code runs faster and uses less memory.
- Suitable for handling large datasets efficiently.

Task Description #6 – Complexity Reduction
Task: Use AI to simplify overly complex logic.

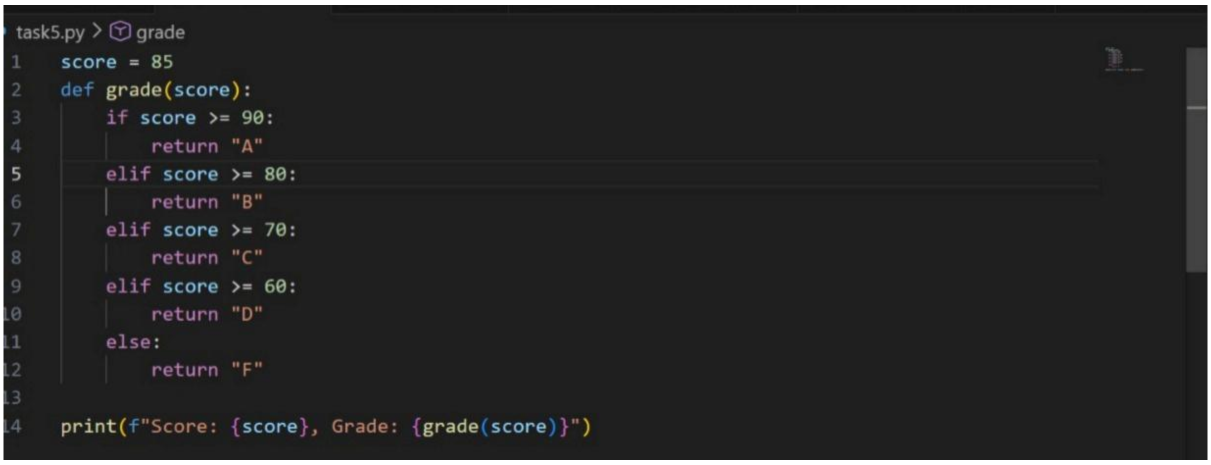
Prompt:

identify the error and fix the error with elif.



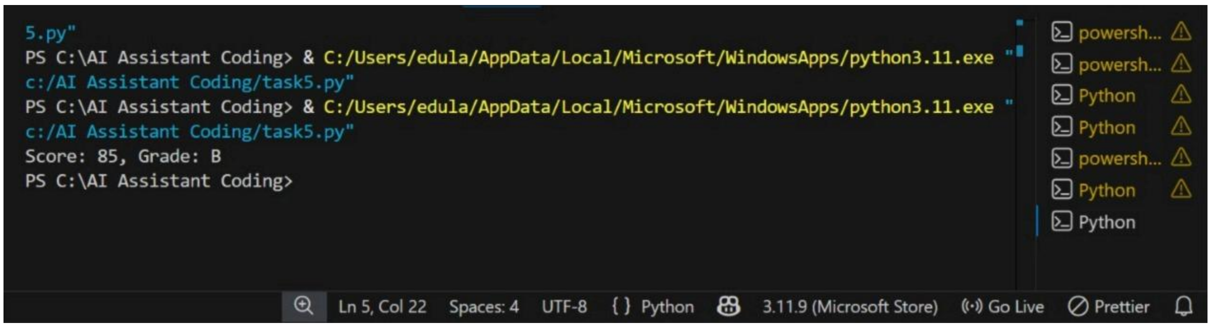
identify the eoor and fix the error with elsif

Code:



```
task5.py > grade
score = 85
def grade(score):
    if score >= 90:
        return "A"
    elif score >= 80:
        return "B"
    elif score >= 70:
        return "C"
    elif score >= 60:
        return "D"
    else:
        return "F"

print(f"Score: {score}, Grade: {grade(score)}")
```

Output:



```
5.py"
PS C:\AI Assistant Coding> & C:/Users/edula/AppData/Local/Microsoft/WindowsApps/python3.11.exe "
c:/AI Assistant Coding/task5.py"
PS C:\AI Assistant Coding> & C:/Users/edula/AppData/Local/Microsoft/WindowsApps/python3.11.exe "
c:/AI Assistant Coding/task5.py"
Score: 85, Grade: B
PS C:\AI Assistant Coding>
```

Explanation:

Replaced nested if-else **blocks with** elif

- Reduces unnecessary nesting.
- Makes the logic easier to understand.

Improved code readability

- The grading conditions are now clearly ordered.

- Each condition is checked only when the previous one fails.

Reduced logical complexity

- Eliminates deep indentation levels.
- Easier to debug and maintain.

Same functionality with fewer lines

- Output remains unchanged.
- Code is more concise and clean.

Better performance and maintainability

- Less branching improves logical flow.
- Suitable for future updates or modifications.