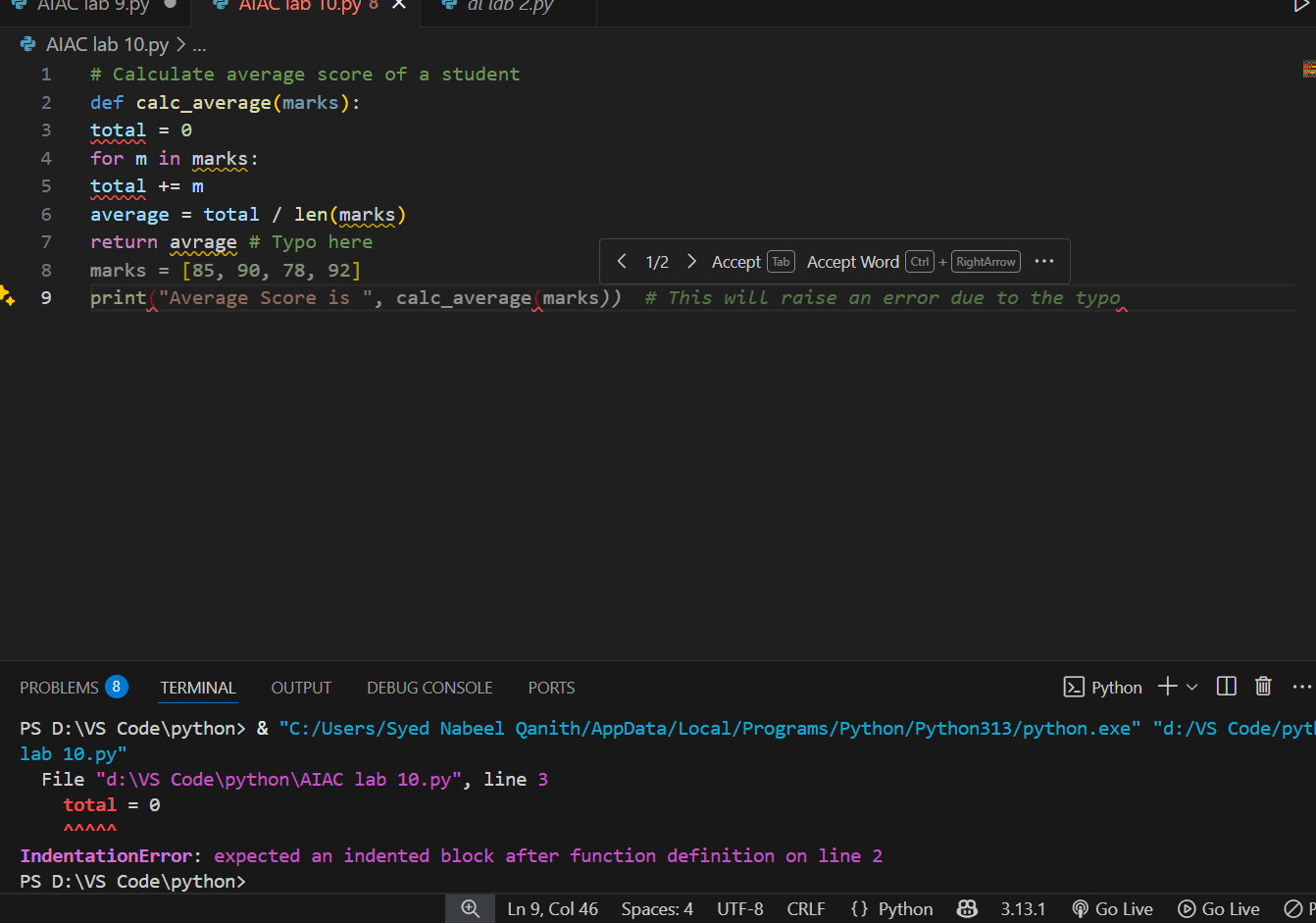
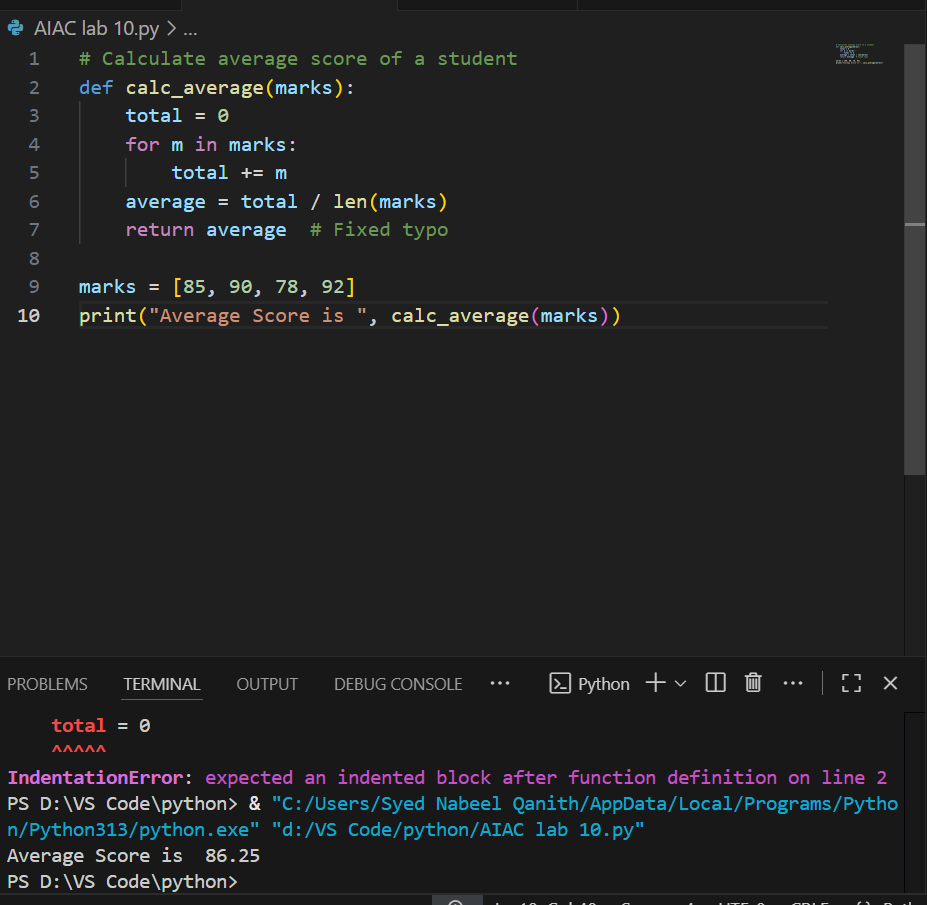
LAB ASSIGNMENT 10

Task Description #1 – Syntax and Logic Errors  
Task: Use AI to identify and fix syntax and logic errors in a faulty  
Python script

Sample code which gives error:



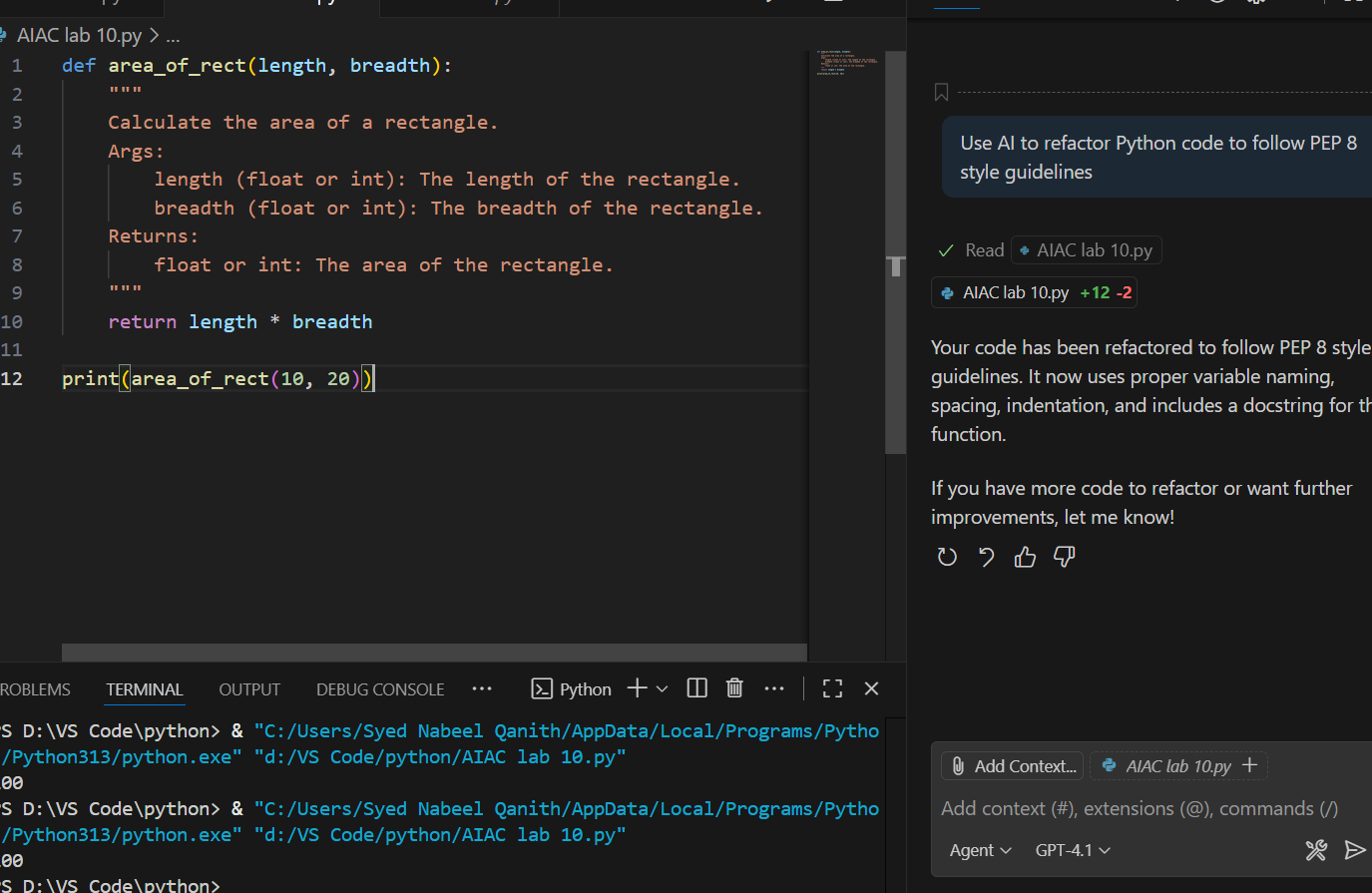
**Prompt:** correct the code and give me appropriate output.



Expected Output:  
• Corrected and runnable Python code with explanations of the  
fixes

Task Description #2 – PEP 8 Compliance  
Task: Use AI to refactor Python code to follow PEP 8 style guidelines.

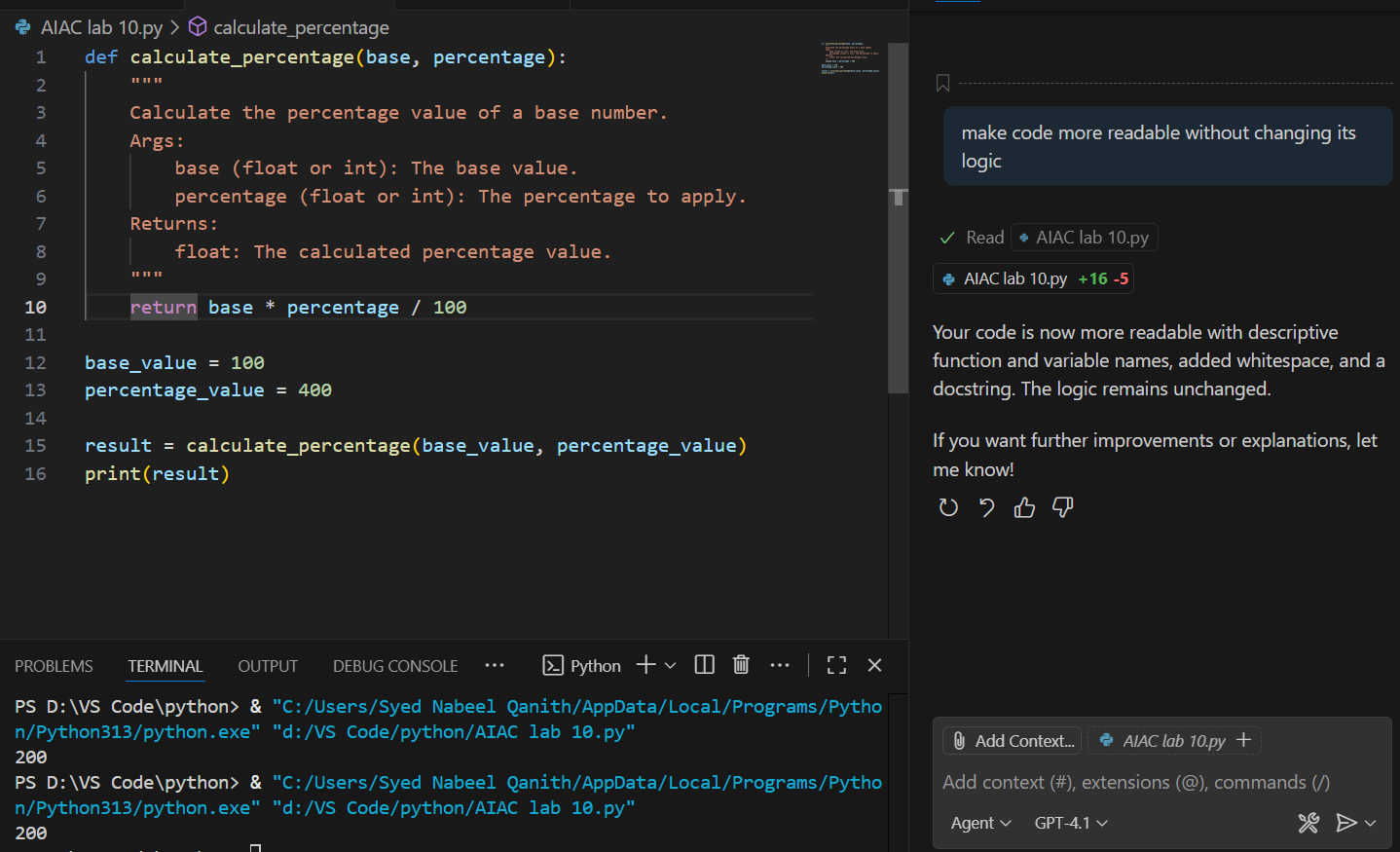
Prompt: Use AI to refactor Python code to follow PEP 8 style guidelines.



Expected Output:  
• Well-formatted PEP 8-compliant Python code.

Task Description #3 – Readability Enhancement  
Task: Use AI to make code more readable without changing its logic.

Prompt:Make Code more readable without changing its logic.

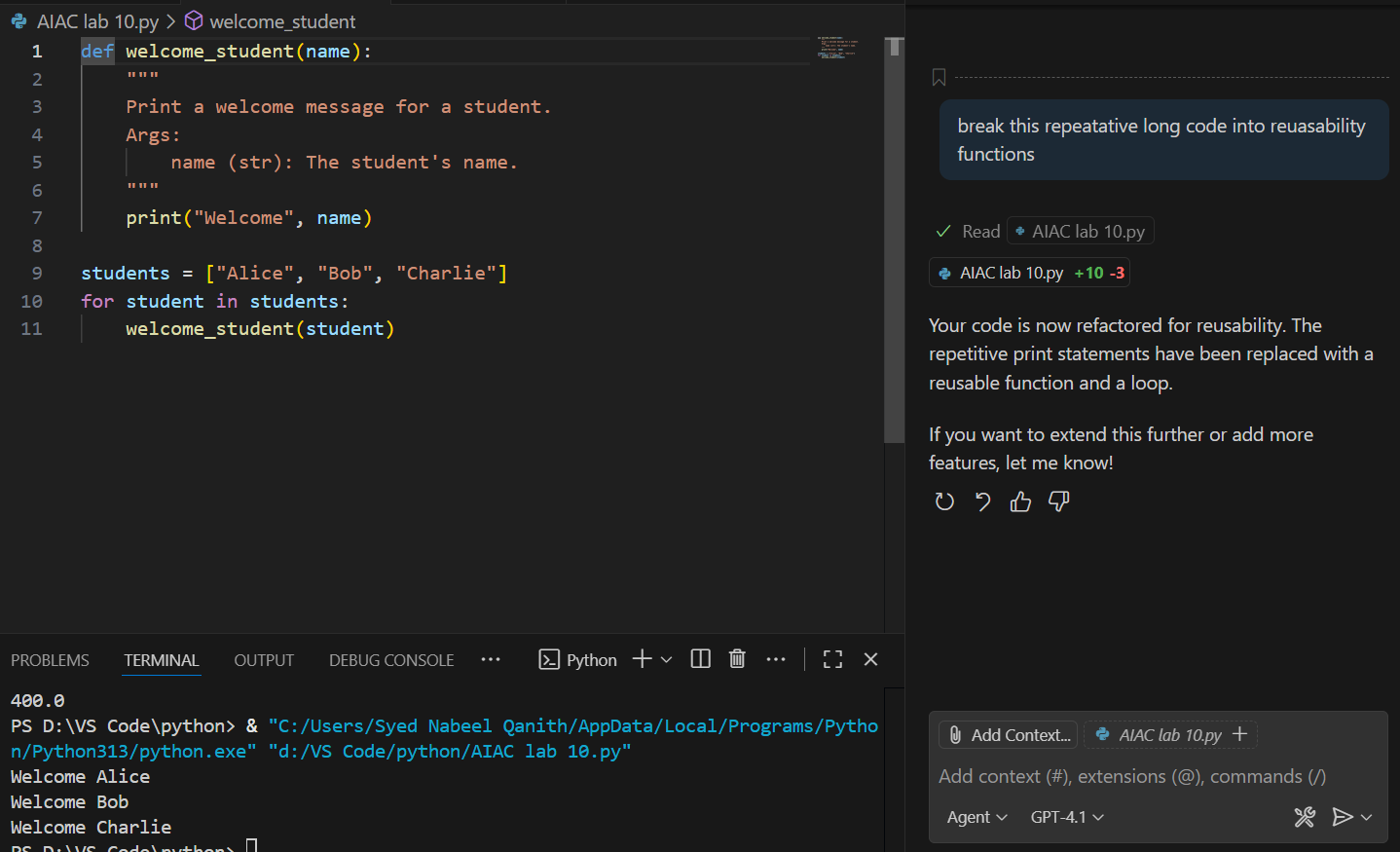


Expected Output:  
• Python code with descriptive variable names, inline comments,

and clear formatting.

Task Description #4 – Refactoring for Maintainability  
Task: Use AI to break repetitive or long code into reusable functions

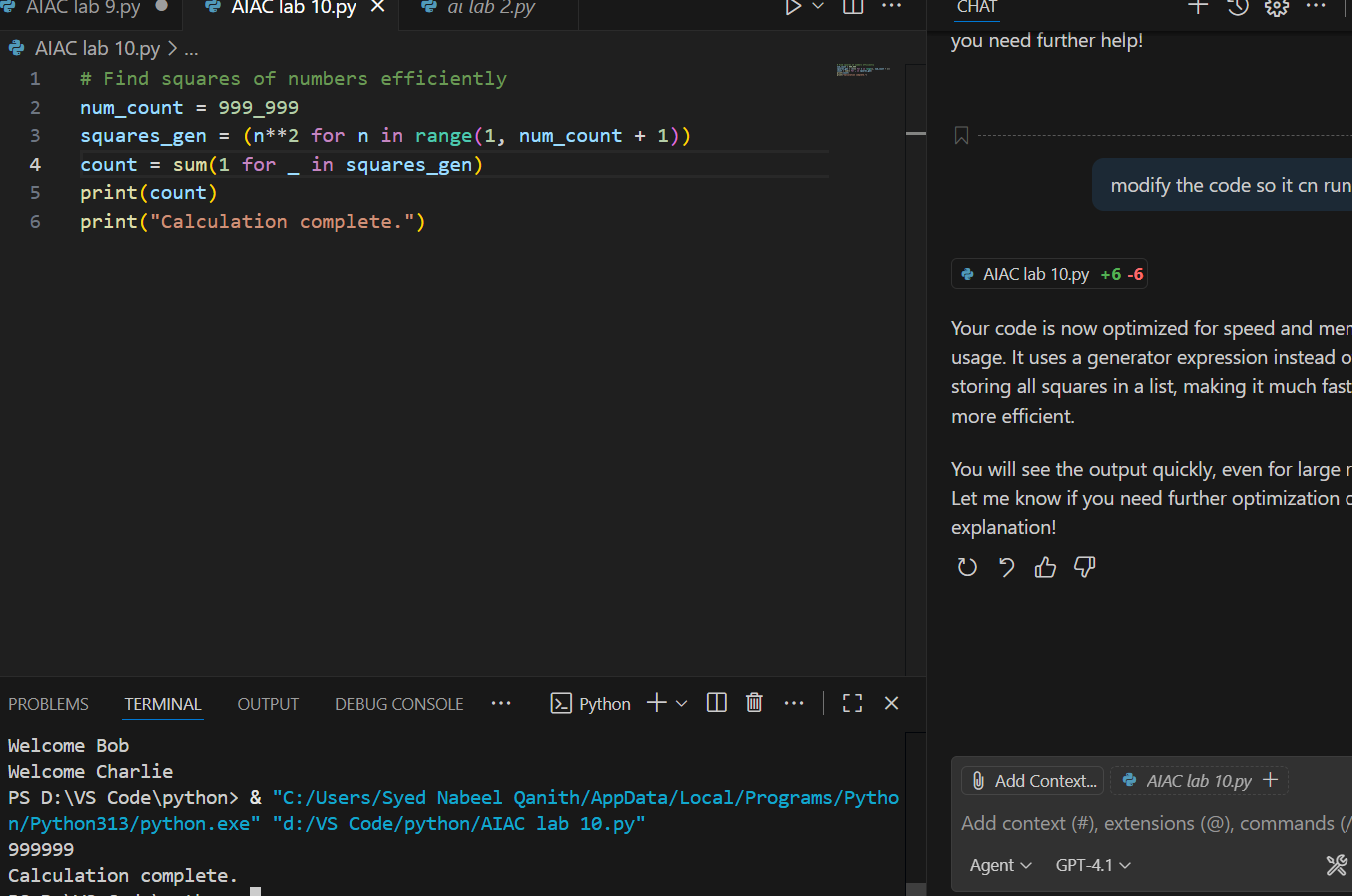
Prompt: break this repeatative long code into reuasability functions



Expected Output:  
• Python code with descriptive variable names, inline comments,

and clear formatting.

Task Description #5 – Performance Optimization  
Task: Use AI to make the code run faster

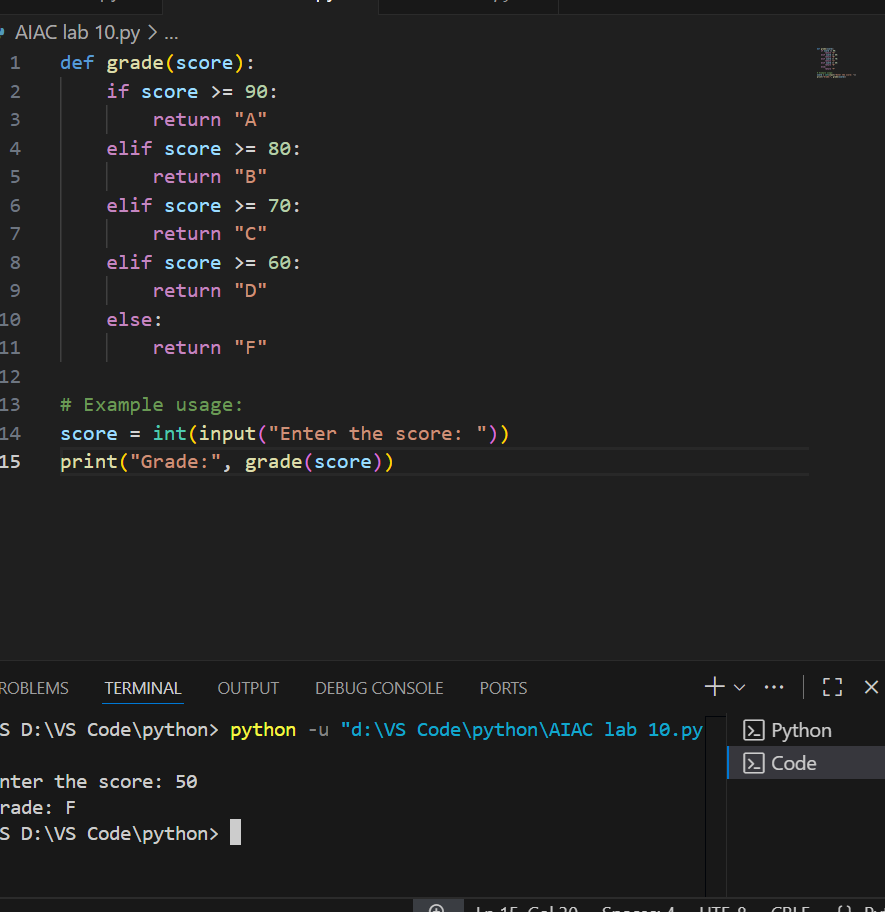


Expected Output:  
• Optimized code using list comprehensions or vectorized  
operations

ask Description #6 – Complexity Reduction  
Task: Use AI to si Expected Output:  
• Optimized code using list comprehensions or vectorized  
operationsmplify overly complex logic.  
Sample Input Code:  
def grade(score):  
if score >= 90:  
return "A"  
else:  
if score >= 80:  
return "B"

else:  
if score >= 70:  
return "C"  
else:  
if score >= 60:  
return "D"  
else:  
return "F"

prompt:Simplify this overly comlex prompt.



Expected Output:  
• Cleaner logic using elif or dictionary mapping