Assignment 10.1

# Lab 10 – Code Review and Quality: Using AI to Improve Code Quality and Readability

* Name – G.Sanjansah ,
* Date – 15-09-2025
* Subject – AI Assisted Coding
* Hall Ticket Number – 2503a52l20
* Student Mail Id – [2503a52l20@sru.edu.in](mailto:2503a52l20@sru.edu.in)

Task Description 1 – Syntax and Logic Errors

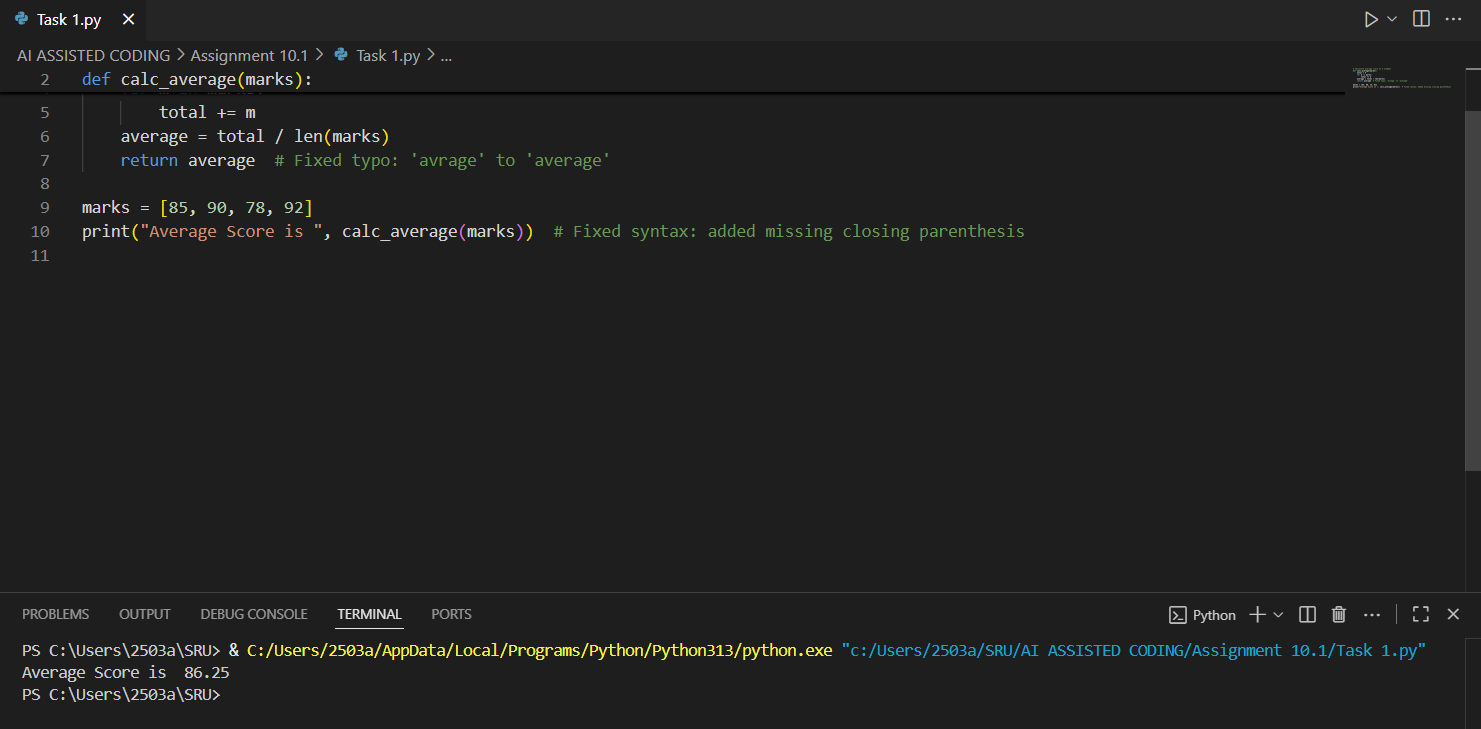
Task: Use AI to identify and fix syntax and logic errors in a faulty Python script.

Sample Input Code:

# Calculate average score of a student  
def calc\_average(marks):  
total = 0  
for m in marks:  
total += m  
average = total / len(marks)  
return avrage # Typo here  
marks = [85, 90, 78, 92]  
print("Average Score is ", calc\_average(marks)

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

A screenshot of a computer program

AI-generated content may be incorrect. 

Task Description 2 – PEP 8 Compliance

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

Sample Input Code:

def area\_of\_rect(L,B):return L\*B  
print(area\_of\_rect(10,20))

Expected Output:

• Well-formatted PEP 8-compliant Python code

Screenshots :

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Task Description 3 – Readability Enhancement

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

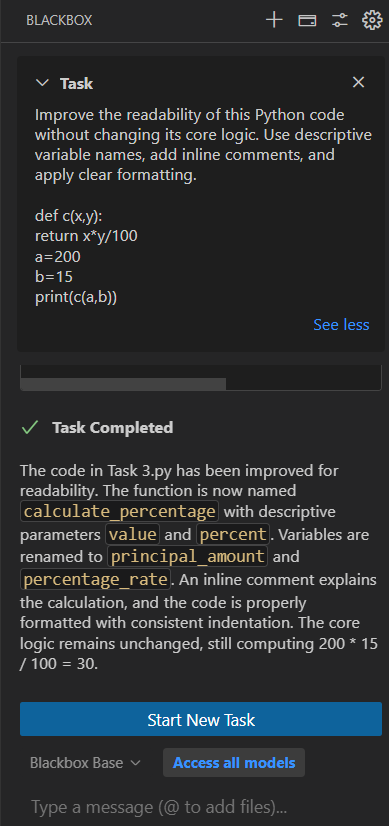
Sample Input Code:

def c(x,y):  
return x\*y/100  
a=200  
b=15  
print(c(a,b))

Expected Output:

• Python code with descriptive variable names, inline comments, and clear formatting

Screenshots :



A screenshot of a computer

AI-generated content may be incorrect.

Task Description 4 – Refactoring for Maintainability

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

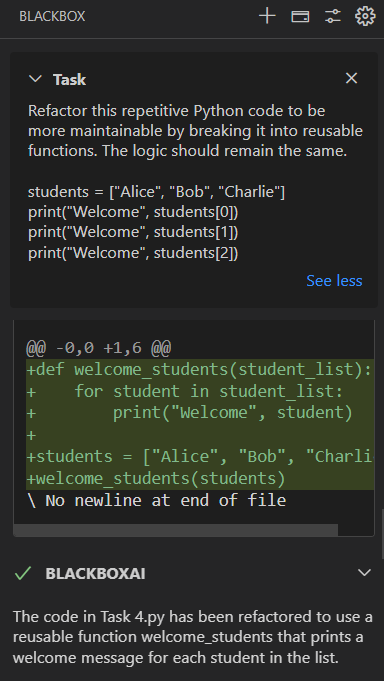
Sample Input Code:

students = ["Alice", "Bob", "Charlie"]  
print("Welcome", students[0])  
print("Welcome", students[1])  
print("Welcome", students[2])

Expected Output:

• Modular code with reusable functions.

Screenshots :



A screenshot of a computer program

AI-generated content may be incorrect.

Task Description 5 – Performance Optimization

Task: Use AI to make the code run faster.

Sample Input Code:

# Find squares of numbers  
nums = [i for i in range(1,1000000)]  
squares = []  
for n in nums:  
squares.append(n\*\*2)  
print(len(squares))

Expected Output:

• Optimized code using list comprehensions or vectorized operations.

Screenshots :

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Task Description 6 – Complexity Reduction

Task: Use AI to simplify 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"

Expected Output:

• Cleaner logic using elif or dictionary mapping.

Screenshots :

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.