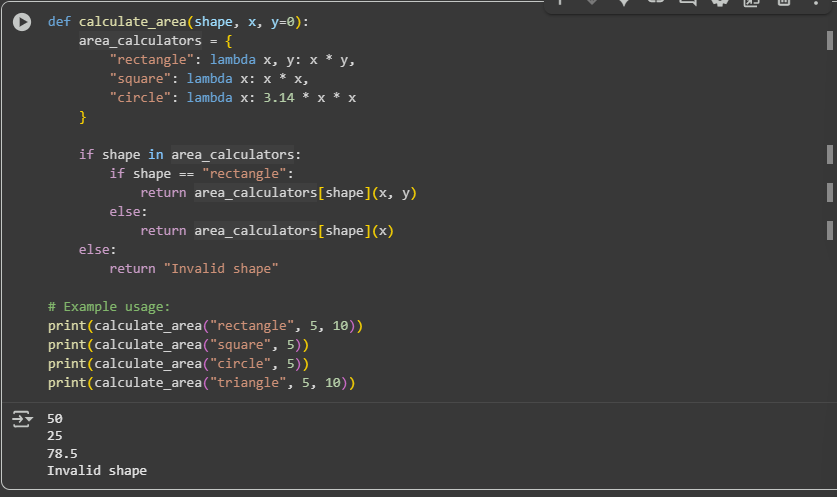
**AI ASSIGNEMNT-13.3**

**Task Description #1 –** Remove Repetition  
Task: Provide AI with the following redundant code and ask it to  
refactor  
Python Code  
def calculate\_area(shape, x, y=0):  
if shape == "rectangle":  
return x \* y  
elif shape == "square":  
return x \* x  
elif shape == "circle":  
return 3.14 \* x \* x  
Expected Output  
• Refactored version with dictionary-based dispatch or separate  
functions.  
• Cleaner and modular design

**PROMPT:** def calculate\_area(shape, x, y=0): if shape == "rectangle": return x \* y elif shape == "square": return x \* x elif shape == "circle": return 3.14 \* x \* x refact the redundant code with dictionary-based dispatch or seperate functions so there should be no repetition

**CODE AND OUTPUT:**



**OBSERVATION:**

A function calculate\_area that calculates the area of different shapes (rectangle, square, circle) using a dictionary of lambda functions. The output shows the calculated areas for the examples provided and also demonstrates the "Invalid shape" output for an unhandled shape ("triangle").

**Task Description #2** – Error Handling in Legacy Code  
Task: Legacy function without proper error handling  
Python Code  
def read\_file(filename):  
f = open(filename, "r")  
data = f.read()

f.close()  
return data  
Expected Output:  
AI refactors with with open() and try-except

**CODE AND OUTPUT:**

A screenshot of a computer program

AI-generated content may be incorrect.

**OBSERVATION:** refactoring a traditional for loop used to calculate the squares of numbers in a list into a more concise and efficient list comprehension. The output shows the list of squared numbers.

**Task Description #3 –** Complex Refactoring  
Task: Provide this legacy class to AI for readability and modularity  
improvements:  
Python Code  
class Student:  
def \_\_init\_\_(self, n, a, m1, m2, m3):  
self.n = n  
self.a = a  
self.m1 = m1  
self.m2 = m2  
self.m3 = m3  
def details(self):  
print("Name:", self.n, "Age:", self.a)  
def total(self):  
return self.m1+self.m2+self.m3  
Expected Output:  
• AI improves naming (name, age, marks).  
• Adds docstrings.  
• Improves print readability.  
• Possibly uses sum(self.marks) if marks stored in a list

**CODE AND OUTPUT:**

A computer screen shot of a program

AI-generated content may be incorrect.

**OBSERVATION:**

**A Student class with methods to display student details and calculate their total score from three subjects. The output shows the student's name and age, followed by their calculated total score.**

**Task Description #4 –** Inefficient Loop Refactoring  
Task: Refactor this inefficient loop with AI help  
Python Code  
nums = [1,2,3,4,5,6,7,8,9,10]  
squares = []  
for i in nums:  
squares.append(i \* i)  
Expected Output: AI suggested a list comprehension

**CODE AND OUPUT:**

A screenshot of a computer program

AI-generated content may be incorrect.

**OBSERVATION:**

A function read\_file\_safe that demonstrates safe file reading using a try...except block to handle FileNotFoundError and other exceptions. It also uses the with statement for proper file handling. The output shows the content of an existing file and an error message for a non-existent file.