**AI LAB ASSISGNMENT 13.3**

NAME:K.ShivaShankar

ROLL NO:2403A510E5

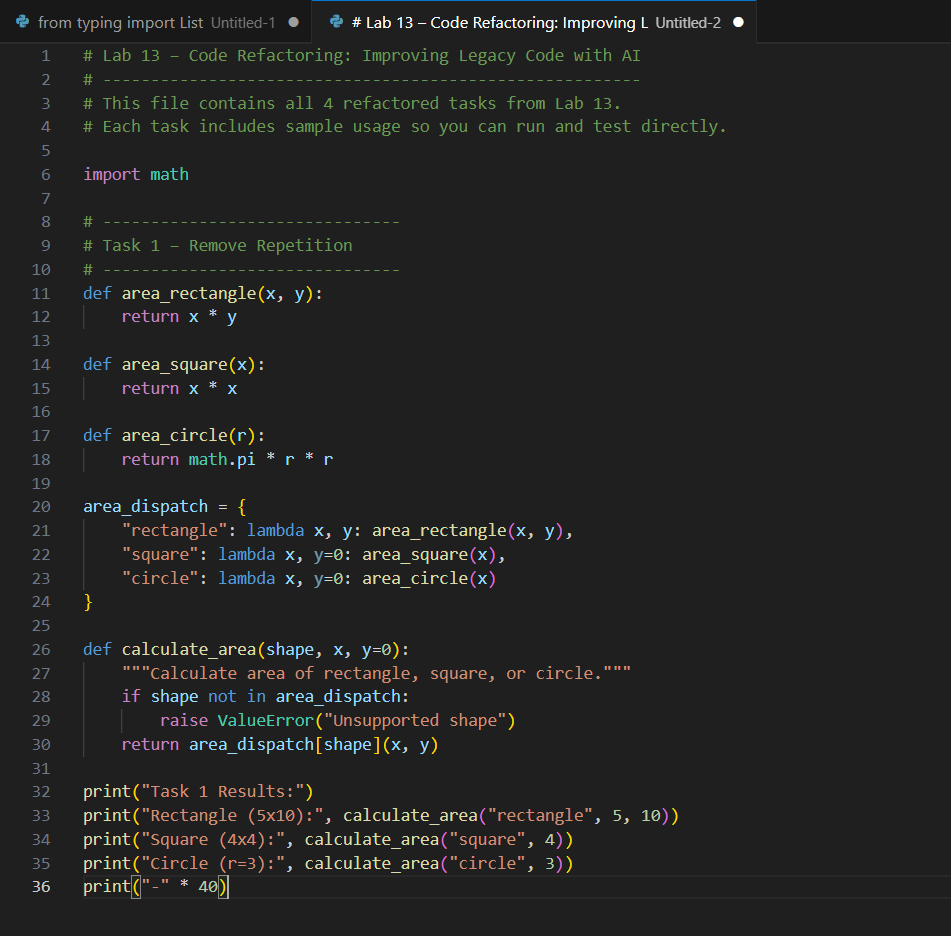
BATCH:05

Task 1 – Remove Repetition

Prompt:

“Refactor a function that calculates the area of different shapes using repetitive if-elif statements. Improve the design by creating separate functions for each shape and using a dictionary-based dispatch to select the correct function, making the code cleaner and easier to maintain.”.

Code



Output :

A screen shot of a computer program

AI-generated content may be incorrect.

Observation:

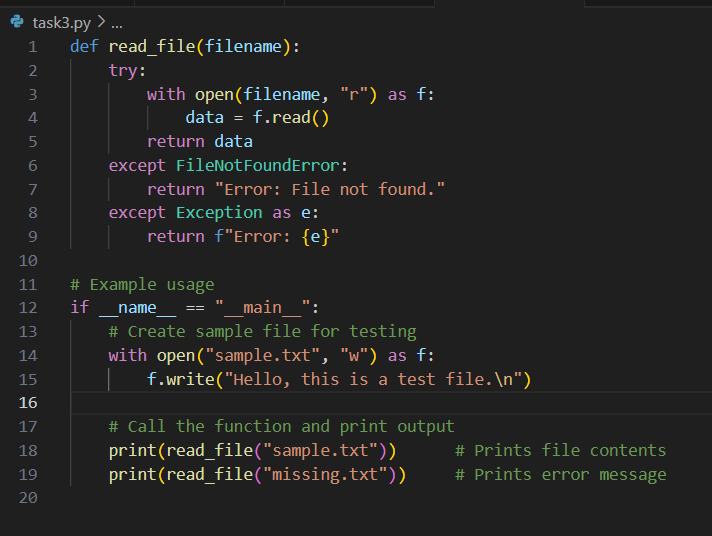
The refactored code eliminates repetition by using helper functions and a dictionary-based dispatch system. It is modular, easier to extend, and more readable.

Task 2 – Error Handling in Legacy Code

Prompt:

“Improve a file reading function that currently crashes when the file does not exist or there is an unexpected error. Use Python’s with statement and exception handling to make it safe and user-friendly.”

Code :



Output Example:

A screen shot of a computer

AI-generated content may be incorrect.

Observation:

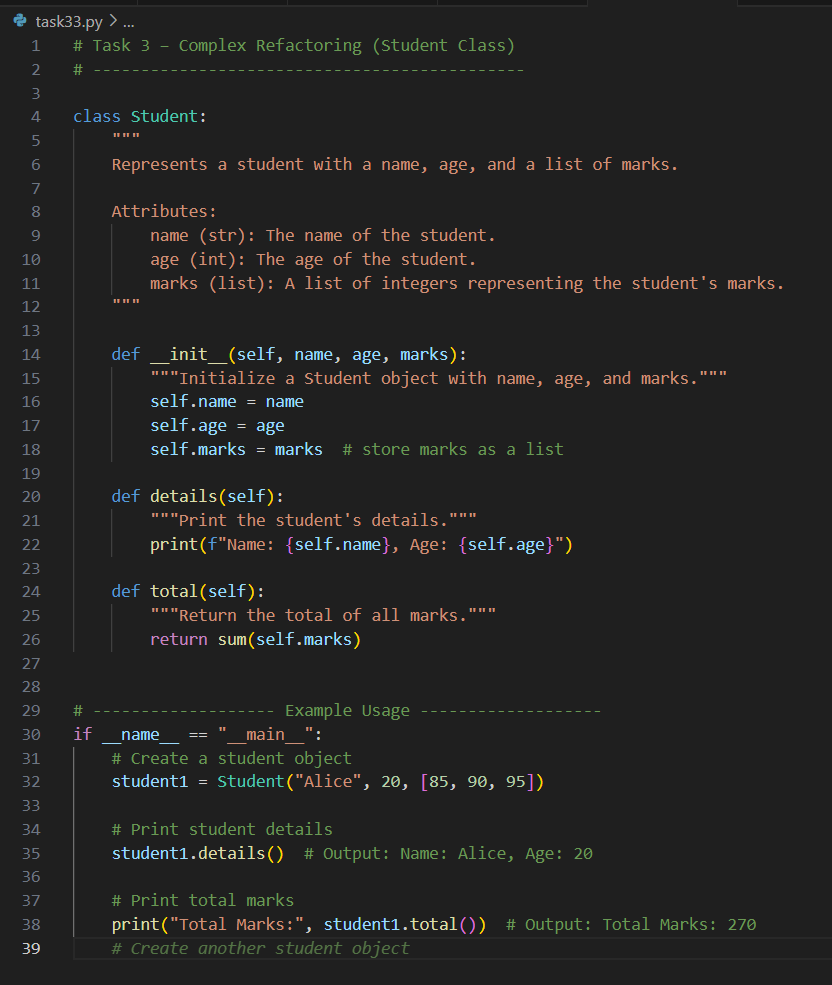
The refactored function ensures the program does not crash when reading files. Using with open() guarantees proper resource management, and try-except handles both missing files and unexpected errors, improving robustness and user experience.”

Task 3 – Complex Refactoring (Class)

Prompt:

“Refactor a student class that stores individual marks as separate variables. Improve code readability, maintainability, and scalability by using a list for marks, better variable names, and adding docstrings.”

Code :



Output Example:

A black screen with white text

AI-generated content may be incorrect.

Observation:

The refactored class is more modular and readable. Storing marks in a list allows easy extension to more subjects. Method names and docstrings improve code understanding, and using sum(self.marks) simplifies total calculation.”

Task 4 – Inefficient Loop Refactoring

Prompt:

“Optimize a loop that calculates the squares of numbers using a more Pythonic approach. Refactor it into a concise, readable format while maintaining the same output.”

Code :

A screenshot of a computer program

AI-generated content may be incorrect.Output Example:

A screenshot of a computer

AI-generated content may be incorrect.

Observation:

“Using a list comprehension reduces code length and improves readability. The functionality remains identical, but performance is slightly better and the code is more idiomatic to Python standards.”