**Assignment 13.3**

Name: Harini,

Roll no:2503a51l40,

Batch:20,

Subject: AI Assisted coding.

Lab13: Code refactoring: improving legacy code

Task1:

Remove repetition,

Generate a redundant code and ask it to refactor.

1.prompt:

Refactor a function to calculate the area for different shapes (rectangle, square, circle) without repeating code.

2.code:

A computer screen with text on it

AI-generated content may be incorrect.

3.output:

A screen shot of a computer code

AI-generated content may be incorrect.

4.observation:

The refactored function uses conditional logic to handle each shape efficiently, improving maintainability and reducing code duplication.

Task2:

Error handling in legacy code with proper functions.

1.prompt:

Add error handling to legacy code to prevent crashes.

2.code:

A computer screen shot of a program code

AI-generated content may be incorrect.

3.output:

A screen shot of a computer

AI-generated content may be incorrect.

4.observation:

Using try-except blocks allows the code to handle specific errors (like division by zero) and unexpected exceptions, making the code more robust and user-friendly.

Task3:

Provide this legacy with complex refactoring.

1.prompt:

Refactor a legacy Student class to improve encapsulation, validation, and error handling.

2.code:

A computer screen shot of code

AI-generated content may be incorrect.

3.output:

A black background with white text

AI-generated content may be incorrect.

4.observation:

The refactored class uses private attributes, properties, and input validation, resulting in safer and more maintainable code that prevents invalid data and handles errors gracefully.

Task4:

Inefficient loop refactoring.

1.prompt:

Refactor an inefficient loop to use a more efficient approach, such as list comprehensions.

2.code:

A computer screen with colorful text

AI-generated content may be incorrect.

3.output:

A close up of a computer screen

AI-generated content may be incorrect.

4.observation:

The refactored loop is shorter and faster, leveraging Python’s built-in features for better performance and readability.