Phase 1: Planning

Requirements Gathering: Identify the requirements of the code, which appears to be a simple calculator program with additional classes for unstructured code, a calculator, and shapes (Circle and Square).

Define Project Scope: Determine the scope of the project, which includes creating a calculator program with the specified classes and methods.

Phase 2: Analysis

Functional Requirements: Identify the functional requirements of the code, which include:

The Unstructured Code class should have a method xyzzy that takes two arguments and prints their sum.

The Calculator class should have methods add and multiply that take two arguments and print their sum and product, respectively.

The legacy function should take two arguments and print their sum.

The Inflexible Shape class should have a method calculate area that is implemented by its subclasses Circle and Square.

Non-Functional Requirements: Identify any non-functional requirements, such as performance, security, or usability constraints.

Phase 3 Design

Class Diagram: Create a class diagram to visualize the relationships between the classes:

Phase 4: Implementation

Write the Code: Implement the code according to the design, which is already provided.

Phase 5: Testing

Unit Testing: Write unit tests for each class and method to ensure they function as expected.

Integration Testing: Write integration tests to ensure the classes work together correctly.

Phase 6: Maintenance

Monitor and Fix Issues: Monitor the code for any issues or bugs and fix them as needed.

Refactor and Improve: Refactor and improve the code to make it more efficient, readable, and maintainable.