PROG-2200 Assignment 3

Title

SOLID Principles Refactoring Project

Value

10%

Learning Outcomes Evaluated

Develop an application by selecting and utilizing appropriate design patterns to support object-oriented design principles.

Instructions

You will be given a Java codebase that is functional but violates several SOLID principles. Your task is to refactor this code to adhere to these principles without altering its core functionality.

Tasks:

- Analysis: Identify the violations of SOLID principles in the provided code.
    - Choose between 1 of the 3 options
- Implementation: Refactor the code according to your plan. Ensure that the functionality remains intact.
- Documentation: Document your changes and explain how your refactoring aligns with each of the SOLID principles.
- Provided Codebase: A simple application, such as a basic inventory management system, a small online booking system, or a simple document editor.

The code will have deliberate flaws that violate SOLID principles, such as classes with multiple responsibilities, rigid design that's hard to extend, inappropriate use of inheritance, etc.

Deliverables

Github URL in a .txt file. Refactored Java code & documentation outlining your refactoring process and justifications.

Evaluation

Learning Outcomes	Components	Points
Develop an application by selecting and utilizing appropriate design patterns to support object-oriented design principles.	Adherence to SOLID Principles: How well your refactoring aligns with each of the SOLID principles.	/5
	Functionality: The refactored code should maintain its original functionality.	/5
	Documentation and Explanation: Clarity of your documentation and explanations.	/5
Outcome Mark Value		/15

Rubric

Adherence to SOLID Principles (5 points)

5 points: The refactored code fully adheres to all five SOLID principles (Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, Dependency Inversion). Each principle is clearly applied and demonstrated in the code.

4 points: The refactored code adheres to most SOLID principles with minor exceptions. One principle might be slightly less evident or fully applied.

3 points: The refactored code adheres to some SOLID principles but misses or misapplies one or two principles.

2 points: The refactored code makes an attempt to adhere to SOLID principles but fails to apply two or more principles correctly.

1 point: The refactored code shows minimal adherence to SOLID principles with significant violations or misapplications.

0 points: The refactored code does not adhere to SOLID principles.

Functionality (5 points)

- 5 points: The refactored code maintains the original functionality perfectly. All features work as intended without any bugs or regressions.
- 4 points: The refactored code maintains the original functionality with minor issues or bugs that do not significantly impact the overall operation.
- 3 points: The refactored code maintains most of the original functionality, but some features are broken or do not work as intended.
- 2 points: The refactored code maintains only some of the original functionality, with several features broken or significantly altered.
- 1 point: The refactored code maintains very little of the original functionality, with most features broken or significantly altered.
- 0 points: The refactored code does not maintain the original functionality at all.

Documentation and Explanation (5 points)

- 5 points: Documentation is clear, thorough, and well-organized. Each change is explained in detail, and the alignment with SOLID principles is explicitly stated and justified.
- 4 points: Documentation is clear and well-organized, with most changes explained in detail. The alignment with SOLID principles is mostly stated and justified.
- 3 points: Documentation is mostly clear but lacks some detail or organization. The alignment with SOLID principles is partially explained but not fully justified.
- 2 points: Documentation is unclear or poorly organized, with significant details missing. The alignment with SOLID principles is minimally explained.
- 1 point: Documentation is minimal and unclear, with little to no explanation of changes or alignment with SOLID principles.
- 0 points: No documentation or explanation provided.

Total: 15 points