Student Management System - Design Patterns Documentation

This project implements a Student Management System in Java 8 using five core object oriented

design patterns: Factory, Singleton, Observer, Strategy, and Decorator. The system is integrated with a simple Java Swing GUI to simulate functionality.

1. Factory Pattern

Used to create instances of Students and Courses without exposing the instantiation logic to the client. Promotes loose coupling and scalability.

Classes involved:

- StudentFactory
- CourseFactory
- Student
- UndergraduateStudent
- GraduateStudent
- PartTimeStudent
- Course
- CoreCourse
- ElectiveCourse
- LabCourse

2. Singleton Pattern

Ensures that only one instance of key manager classes exists in the application. Used for course registration and grade processing to ensure consistency.

Classes involved:

- CourseRegistrationManager
- GradeProcessingManager

3. Observer Pattern

Implements a notification system where students get updates when their grades are released. Helps in decoupling publishers (notifiers) from subscribers (students).

Classes involved:

- Observer

- Observable

- StudentObserver

- GradeNotifier

4. Strategy Pattern

Allows selecting different grading algorithms at runtime. For instance, grading using GPA,

percentage, or pass/fail systems.

Classes involved:

- GradingStrategy

- GpaStrategy

- PercentageStrategy

- PassFailStrategy

- GradeContext

- HonorsStudentDecorator

- ScholarshipStudentDecorator

6. Java Swing GUI

The MainApp.java class provides a Swing-based interface for demonstrating all patterns. Clicking the 'Run Demo' button simulates student creation, course registration, grade notifications, grading format application, and decorated profile display.