Assignment 2: Composite

Time Estimate: 9 Hours
Due: End of Week 12
Course: Design Patterns

Chapters Covered:

- 14.1, 14.3, 15.1, 15.3, 15.9, 15.10, 16.1, 16.2, 16.3, 16.4, and 16.5

INSTRUCTION

This assignment builds on your architectural modeling and pattern design skills, focusing on the **Composite Pattern** and its application in hierarchical systems. You'll also incorporate a relevant **object creation pattern** and demonstrate understanding of pattern-based decision modeling using activity and decision diagrams. This is an **individual assignment** that emphasizes both design accuracy and thoughtful reflection.

TASK

a. Hierarchical System Context (2 hours)

- Choose a domain where hierarchical relationships are inherent (e.g., UI layout, file explorer, organization tree).
- Describe:
  - A feature or module where multiple layers interact (e.g., nested UI panels)
  - The uniform operations needed across levels (e.g., draw(), count(), evaluate())
- Provide a labeled **block diagram** or outline of the hierarchy

b. Design + Modeling (3 hours)

- Apply the **Composite Pattern** in a proposed class-based design:
  - Define Component, Leaf, and Composite roles
  - Use consistent methods across elements (e.g., print(), calculate())
  - Add a supporting **creation pattern** (Factory, Builder, or Template)
- Model logic in two ways:
  - Create a **UML Class Diagram** for your Composite implementation
  - Draw an **Activity Diagram** or **Decision Tree** to show a workflow tied to this component

c. Pattern Specification + Interpreter (2.5 hours)

- Create a **Decision Table** based on one operation (e.g., evaluate() rules, permission checking)
- Briefly outline how this logic could be encoded using the **Interpreter Pattern**
- Add a few example rules as expressions or pseudocode (max $\frac{1}{2}$ page)

d. Reflection (1.5 hours)

- Submit a 1-page article that addresses
  - Why Composite fits your design's structure and complexity
  - The advantage of your chosen creation pattern
  - How pattern-based logic models (Interpreter, Decision Tables) improve maintainability

SUBMISSION DETAILS

- **Submit via:** Canvas LMS
- **File Naming Convention:**

  ```
  StudentID_Assignment2_CS4203Fall2025
  ```

- Submit a PDF or ZIP file that includes:
  - Block diagram
  - UML Class and Activity or Decision Tree diagram
  - Decision Table and Interpreter logic summary
  - 1-page written reflection

Late work subject to the standard course penalty. Peer review is not required for this assignment.