CS5700 – Midterm Study Guide

The midterm will cover patterns, principles, practices, and pitfalls discussed in class since the beginning of the semester, plus material from Chapters 1-5 of the textbook.

Patterns

- What is a software design pattern and what should its definition contain?
- Patterns to know:
    - Strategy
    - Observer
    - Decorator
    - Factory Method
    - Abstract Factory
    - Singleton
    - Flyweight

Principles

- Core concepts and principles to know:
    - Object identity
    - Classification and sub-classification (specialization)
    - Localization of design decisions
    - Encapsulation
    - Abstraction
    - Low Coupling
    - High Cohesion
    - Open for extension / closed for modification
    - Dependency inversion
    - Modularization
- Other concepts to know:
    - Maintainability
    - Extensibility
    - Reuse

Practices

- Conceptual modeling – why and when
- Conceptual modeling with UML
    - UML Class Diagrams
    - UML Interaction Diagrams (Communication or Interaction)
    - UML State Charts
- Other practices to be familiar with

    - Organize software into loosely coupled layers, e.g., GUI, App Logic, Persistence, etc.
    - Prefer aggregation over inheritance
    - Program to an interface or abstraction
    - Use identifiers that improve readability and maintainability
    - Testing with executable unit test cases

Pitfalls

- Pitfalls to be aware of
    - Uncommunitive names
    - Inconsistent Names
    - Types embedded in names
    - Long Methods
    - Duplicate code
    - Class Explosion
    - Long message chains