Exam Format

• Closed book, closed notes.
• However, you may bring one sheet of 8.5 x 11 inch paper with your notes on both sides of the page. You must put your name on the sheet and turn it in with your exam.
• Draft Exam (subject to change):
  1. 8 Multiple choice questions (24 points).
  2. 4 Short answer questions (24 points).
  3. 2 problems related to the Adventure Game design (25 points).
  4. 2 problems related to domain/conceptual modeling of the problem space (27 points).

Exam Topics
Material Since Exam 1
• Software Design & Architecture
  - Notes 4 & text Ch. 5-7, 12
  - Assignment 4: Android Adventure Game design & implementation.
  - Assignment 5: Design Pattern Adventure Game Enhancements.
• User interface design
  - Notes from Geri Georg’s lecture & text Ch. 13.
• Requirements
  - Notes 5, text Ch. 10, 14 (sec. 8.1 sequence diagrams), Ch. 12 Modeling with classes (domain models).
  - Design Studio 6: Domain Modeling.

Topics on Exam: Notes 4, Text Chapter 5, 6 & 9
Architecture & Design for OO Software
• Assignment 4: Android interface.
• Process of software design.
  - Software design: a model of a system.
  - Model derivation at levels of abstraction.
    - Move from informal design to formal program solution.
    - Models expressed in graphical, formal, or other notations.
• Phases: system architecture, architectural design, interface design, component design, algorithm design.

Topics on Exam: Notes 4
• Design Assessment.
  - Design quality Goals: efficient, cheapest, maintainable.
  - Design principles [Davis]:
    • Considers alternatives.
    • Don’t reinvent the wheel.
    • Minimize intellectual distance.
    • Uniformity.
    • Changeability.
    • Degradable.
    • Reviews.

Topics on Exam: Notes 4
• Component assessment
  - Component parts: interface (contract) and secrets.
  - Component design quality.
    • Cohesion: aim for strong cohesion.
    • Coupling: aim for weak coupling.
• Understandability.
  - Good names.
  - Accurate documentation.
  - Avoid complexity!!
Topics on Exam:
Notes 4

- Architectural structures.
  - Hierarchical (tree) structures.
  - Pipeline architectures: compilers.
  - Layered: operating systems.
  - Client/server.
- Control options.
  - Centralized.
  - Broadcast event-driven.
  - Interrupt driven.

Topics on Exam:
Notes 4, Text Chapter 12

Design patterns: design experience.

- Assignment 5: Design patterns and enhancements.
- Pattern structure.
- Patterns:
  - Façade.
  - Model-View-Controller.
  - Abstract Factory.
  - Broker.

Notes 5, Text: Ch. 10, 14 (Sec. 8.1)

OO Requirements Analysis.

- Overview:
  - Informal problem domains.
  - Functional and non-functional requirements.
  - Problem definition and the need for precision.
  - Prototypes for requirements definition.
- Requirements documents.

Notes 5, Ch. 4 Requirements

- Use-case analysis.
  - Actors.
  - Scenarios: Typical and alternative course of events.
  - Use-case diagrams.
  - Identify use cases: actor or event based approach.

Notes 5 OO Requirements
Text Sec. 8.1

- System sequence diagrams.
  - Depict sequence of actions between actors and the system.
  - Generate from use case scenarios.
  - System event: input event generated by an actor.
  - System operation: response to a system event.

Notes 5 OO Requirements
Notes 5, Ch. 11

Design Studio 6: Domain/Conceptual modeling.

- Identify concepts and their attributes.
  - Concepts tend to be nouns; objects pronouns.
  - Attributes: information held by objects.
- Determine associations and generalizations. Look for collaborators.
Notes 5, Text: Ch. 10

- Evaluating requirements.
  - Unambiguous.
  - Verifiable.
  - Consistent.
  - Complete.
  - Should not dictate a particular design.
  - Use cases: end-to-end, named with verb phrase.
  - Domain/conceptual model: a domain concept (class) is named with a noun phrase; needed information made available through attributes or associations.
  - Non-functional requirements must be testable.