SENG 330 - Summary and Exam Review

Final Exam Format

- 2 hours

- plenty of code to read/write (all Java)

- no notes, calculators, phones, etc.

Study suggestions:

- **Design patterns**: the ones we discussed, and also the general format, how they are written in Java
- **OO design principles**: why they help design, relationship to key qualities, their realization in Java code.
- **Project**: familiarity with basic architecture, key components, infrastructure support. General concepts, not specifics of Spring or JavaFX or Android etc.
- **UML**: familiarity reading and sketching class, object, sequence, and state diagrams
- Study the textbook and exercises, as well as my lecture notes. Read code samples.

After the course, students are able to:

- **Name**, using the proper terminology: The important first principles, techniques, and tools of object-oriented software design;
- **Describe and explain**: The important principles, techniques, and tools of object-oriented software design;
- **Apply**: The important principles, techniques, and tools of object-oriented software design to provide effective solutions to realistic design problems;
- **Evaluate**: The quality of design solutions
- **Write**: well-designed, error-free, and easily understandable software.

Overview

1. Encapsulation

2. Types/Polymorphism

3. Object state and equality

4. Testing

5. Composition

6. Inversion of Control

7. Inheritance

8. Design Patterns

9. Concurrency

10. Serialization and Refactoring

11. Other design approaches — functional programming

Core concepts

- What is an OO design, and what mechanisms (in Java) can we use to create one?
  - Polymorphism, scoping modifiers, interfaces vs abstract class, threads, encapsulation
- Design tradeoffs: what are the (dis)advantages to one particular approach?
  - Principles for assessing design quality: coupling and cohesion, SOLID, modularity
- Moving from the abstraction of the problem to the implementation to realize a solution. Testing the solution for how well it does this.
- Well established patterns to common problems:
  - Decorator, Command, Iterator, Template, Composite, Mediator, Observer

Good Luck on the Final

Have a Relaxing Winter Break