

**CPSC 210 Software Construction**  
**Final Study Guide**

**2009-2010 Winter Term 2**

The final will cover material from the beginning of the term. The weighting of the exam will be roughly:

- 45% on material after midterm #2
- 55% on material from the beginning of term until midterm #2

This study guide covers only the material after midterm #2. Please refer to the study guides for midterm #1 and midterm #2 for guides on studying the other material.

Come to office hours, post to the google group or email questions!

1. At the start of some of the readings, there is a checkmark list about the major concepts covered by the reading. Review these points and make sure you can do the actions listed. For example, the reading about object-oriented design II lists:
  - Perform a basic assessment of a design using the design principles of coupling and cohesion.

Make sure you can do this kind of assessment.

2. Go through the posted exercises.

3. Here is a list of concepts that may be covered on final (again from midterm#2 to the final).
  - Given a problem description (and possibly a partial design), apply the composite, decorator or factory/factory method design pattern and produce an object-oriented design to solve the problem
  - Recognize the composite, decorator or factory/factory method design pattern in a given design or given source code
  - Critique whether a given object-oriented design has high or low coupling
  - Assess whether a given object-oriented design adheres to the open-closed principle and/or the dependency inversion principle
  - Given a specification for a data structure (i.e., like a queue or a doubly-linked list), produce a linked data structure implementation
  - Trace through code for a linked data structure (e.g., lists or trees) and explain what happens for the execution of particular operations given particular data. The execution may involve recursion.
  - Write particular forms of tree traversals

4. The following concepts will not be covered:

- N/A

5. You should have solid working knowledge of the following Java concepts:

- class
- field
- method
- object
- for-each loop (e.g., for (Player p: players) )
- if-else statement
- try-catch statement
- assert
- that a Collection class can aggregate objects of a particular type (e.g., List<Channel>)
- properties of different Java Collections Framework classes (i.e., a list has positional access whereas a set ensures no duplicates...)
- anonymous inner classes (when used with Swing events)
- recursion
- use of object references to produce linked data structures