

CPSC 210 Software Construction
Mid-term #2 Study Guide

2009-2010 Winter Term 2

1. At the start of each reading, 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 GUIs lists as an action:
 - Trace code that uses Swing event handlers.Make sure you can do this kind of tracing of a given code that uses the Swing library.
2. Go through the posted exercises.
3. Here is a list of concepts that may be covered on the midterm.
 - Given a problem statement in English, produce an object-oriented design using the responsibility-driven design approach. In particular, you should be able to:
 - i. Identify likely classes that will be in an object-oriented design for the problem.
 - ii. Identify likely responsibilities for those classes.
 - iii. Produce a UML class diagram that describes the relationships between the classes/objects in the design.
 - Given a UML class diagram, produce Java code that represents the design presented in the diagram. In particular, you should be able to:
 - i. Correctly implement associations, compositions and aggregations.
 - ii. Select an appropriate data structure from the Java Collection Framework (JCF) to represent an association, composition or aggregation based on the UML diagram and the problem statement for the design.
 - iii. Appropriately override the equals and hashCode methods and be able to explain why these methods require overriding.
 - Extract a containment hierarchy for code for an existing Swing GUI and explain the difference between an inheritance and containment hierarchy.
 - Trace Java code that uses Swing event handlers.
 - Recognize and use delegation appropriately when programming.
4. The following concepts will not be covered:
 - Memory management in Java and Java garbage collection
 - Models in Swing programming
 - Producing a GUI with Swing (which would be difficult in the scope of an exam!)
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)