CS209 2nd Midterm Prep

Similarities to, differences from Python

Collections
Common interfaces, implementations
Generics
How/when to use

Software Development
Development Models – waterfall, iterative
Prototypes

Testing
Different levels of testing (unit, integration, system, …)
Black-box testing vs. White-box testing
Coverage criteria
JUnit testing framework

- Benefits, limitations
- How to write JUnit test cases

Design Principles
Open-closed principle

- Open to extension, closed to modification
- Liskov substitution principle

DRY (Don't repeat yourself)
Shy code (coupling)
Single Responsibility Principle
Code smells
Appropriate solutions
Defending solutions

Refactoring
Resolving code smells using abstraction

GUI Programming
Common components, containers
Layout Managers
Event-driven programming

- ActionListeners, Handling events, inner classes, anonymous inner classes, adapter classes

Jar Files
Deploying Java applications
How to create, extract, use

What I expect from you on the exam:

- To know Java/OO-programming/design terminology
- To describe a design solution and be able to defend it
- To know the appropriate Linux commands and how to use them, given a typical situation
- To be able to read, understand, and write Java programs, with or without documentation
- To be able to write a program (given an algorithm or creating your own algorithm, given a problem) or class
    - Syntax must be very close to correct (correct keywords, punctuation, special characters, variable naming, operations)
    - Since it’s on paper, there is some leniency

Suggestions on how to prepare:

- Exam is ***terminology heavy***. Make sure you know the terminology (much of it is in the list above).
- Practice reading through programs, tracing through them, and saying what the output should be
    - Review your assignments
- Read through slides for vocabulary, review questions, exercises
- Think about the various designs we have discussed in certain situations and what the tradeoffs are to each design.