Pre-Midterm Topics

- 33% of exam
  - Java
  - Testing
  - Junit
  - Principles of Design
  - Design Patterns
Post-Midterm Topics

• 67% of exam
  – Concurrency – Threads & Synchronization
  – Concurrency – Design patterns
  – Garbage Collection
  – Java Monitors
  – RMI
  – Nonblocking Algorithms
  – Reflection
  – Generics

Pre-Midterm Topics
Java

• Subtyping
  – Inheritance
  – Interfaces
• Method dispatch
  – Static
  – Dynamic

Testing

• Black box vs. White box testing
• Coverage criteria
  – Stmt
  – Branch
  – Condition
JUnit

- Writing and Understand Junit test cases

Principles of Design

- What are the goals of the design process?
- What are the notional phases of design?
- Be able to define, compare & contrast the following principles:
  - Decomposition, Abstraction, Information Hiding, Modularity, Hierarchy, Separating Policy and Mechanism
- Understand the Uses Hierarchy
Design Patterns

• Be able to describe the design patterns used in your projects:
  – Visitor
  – Observer
  – Factory
  – Decorator
  – Proxy

Post-Midterm Topics
Concurrency – Threads & Sync

- Understand conceptually what a thread is
- Understand scheduling & interleaving
- Java Thread class and methods
- Atomicity, Visibility, Synchronization

Concurrency – Design Patterns

- State-dependent actions
  - Guarding
  - Optimistic retrying
- Producer-Consumer designs
Reading Concurrent Code

• Be able to trace concurrent code & find instances:
  – deadlock
  – livelock
  – data races
  – Improperly synchronized check-then-act sequences

Concurrency – 1.5 Utilities

• Executors
• Locks & Conditions
• Atomic variables
Nonblocking Algorithms

- Definition
- Pros & cons – nonblocking vs. blocking
- Understand nonblocking data structures
  - Stack & Queue

RMI

- Understand RMI Architecture
  - Codebase
  - Remote interfaces
  - Registry
- Server setup
  - Create remote instances & export
  - Registry mgmt methods
  - Security
Reflection

• Be able to recognize simple uses of reflection.
• No detailed questions

Generics

• Recognize intent of generic code
• No detailed questions
Garbage Collection

- General concepts
  - Reachability
- Techniques
  - Reference counting
  - Mark & sweep
  - Copying GC
  - Generational collection
- Pros & Cons, Tradeoffs, etc.