Com S 430
Topics for midterm exam

Concurrency fundamentals

Be able to recognize correct/incorrect use of synchronization locks for mutual exclusion and when/where locking is needed (shared + mutable state variables)
Fully synchronized objects (aka the "monitor" pattern)
Understand/apply the fundamental rule of locking (e.g. see Goetz, p. 28)
Class-level locks for static data, locking instance locks from inner classes
The problem of memory visibility, reordering of memory operations, fundamental rule of visibility (see Goetz, p. 37), using synchronization to ensure visibility
Semantics of wait/notify/notifyAll, the fundamental rules of condition waiting (e.g. see p. 301)
Client-side locking for traversals, why it is needed
Immutability (particularly as defined for Java, and the initialization guarantees for final fields)
Proper construction, safe publication guidelines (e.g. see p. 52)
Confinement concepts (method, instance, thread)

Using threads and concurrency utilities

Using an auxiliary thread to execute a task asynchronously
Alternatives to starting threads explicitly – using ad hoc thread pools
Confinement rules for UIs
Know how to use an auxiliary thread in a UI and provide safe updates using invokeLater
Be able to implement a simple promise-style Future, know how to use a Future as in the AST cache example, know how to use FutureTask and the Callable interface
Thread-safe queues and their relevance for confinement
Asynchronous messages and correlation

References

Relevant sections of Goetz' book - although we covered these topics in class, the sections below are generally concise, clear, and accurate explanations.
2.1 - 2.4 - safety and locking
3.1, 3.2 - visibility
3.2.1 - safe construction
3.3 - confinement
3.4 - immutability
3.5 - safe publication
4.1, 4.2 - full synchronization (instance confinement)
4.4.1 - client-side locking
5.1, 5.2 - synchronized collections
5.6 - a result cache example (similar to the AST example in homework)
6.1, 6.2 - Executor interface and simple thread pools
9.1, 9.2, 9.3 - threads in UI frameworks
14.1, 14.2 - using wait/notify
16.1 - background on the Java Memory Model
16.3 - initialization guarantees for final fields