Tentative Syllabus

The following is a tentative logically-ordered syllabus for EC500. The actual material covered may be a subset or superset of this syllabus, depending on class progress and makeup, and will likely follow a different order. Please see the CourseSchedule topic for an updated schedule of course topics.

0. General
   A. Languages
   B. Project Management
I. Design
   A. Goals
   B. Code
   C. Testing
   D. Patterns
II. Distribution
   A. General
   B. Concurrency
   C. Best practices
III. Optimization
   A. Types
   B. Databases
   C. Streams
IV. Security
   A. Coding
   B. Defensive programming
   C. Android framework examples
   D. Side-channels

0. General
A. Languages

- Core
  - C
  - C++
  - Java
  - JavaScript
- Possible additions
  - PHP
  - Perl
  - Ruby
  - Kotlin
  - Rust
  - Assembly

B. Project Management

- Teams
  - Roles
  - Meetings
  - Inter-personal
- Development
  - Peer review
  - Version control
  - Ticketing
  - Continuous integration
  - Development and production
  - Process management

I. Design

A. Goals

- Fault tolerance
- Transparency
- Flexibility
- Scalability

B. Code

- Documentation
- Modularity
C. Testing

- Debugging
  - Watchpoints/ breakpoints
  - Exceptions, assertions
  - Multi-threaded /multi-process

- Types
  - Unit
  - System

- Properties
  - Coverage
  - Flakiness
  - Boundary conditions
  - Mock
  - Black/clear box

D. Patterns

- Creation
  - Factory
    - Simple, Classic, Abstract
  - Singleton
  - Prototype
  - Builder

- Structural
  - Adapter
  - Bridge
  - Composite
  - Decorator
  - Facade
  - Flyweight
  - Proxy
II. Distribution

A. General

- Distributed vs. parallel
- Process vs. thread
- Java memory model

B. Concurrency

- Threads
- Liveness
  - Livelock
  - Deadlock
  - Starvation
- Synchronization
  - Locks
  - Blocking
  - Atomicity
  - Race conditions
- Deployment
  - Containers
  - Docker
  - Kubernetes
  - OpenShift
  - REST
  - Virtual Tuesday

C. Best practices
III. Optimization

A. Types

- Space
  - Memory
  - Local storage
  - Remote storage
- Computation
  - Data structures
  - Standard Libraries (e.g. STL)
- Communication
  - Caching
  - Parallelization

B. Databases

- Relational algebra
- Entity-relationship modeling
- Normal forms
- Optimizations
  - Queries and views
  - Federation and distribution
  - Triggers

C. Streams

- Mapping
- Reductions
- Parallel
  - Fork/Join
  - Spliterators

IV. Security

A. Coding

- Fortify kingdoms
  - Input validation and representation*
B. Defensive programming

- Input validation
- Exception management
  - Checked/unchecked
- Pointer management
- Assertions

C. Android framework examples

- UI redressing
- Interfaces
- Parsing
- Deserialization
- Verification
- Flash/hardware
- SIM
- AGPS
- USB

D. Side-channels

- Timing
- Power
- Cache