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.

I. Design
   A. Code
   B. User interface
   C. Testing
II. Distribution
   A. Concurrency
   B. Documentation
III. Optimization
   A. Types
   B. Methods
IV. Security
   A. Coding
   B. Defense against the future
   C. Best practices

I. Design

A. Code
   Modularity
   Duplicate code
   Function size
   Patterns
B. User interface

 - Model-View-Controller

C. Testing

 - Debugging
   - Watchpoints
   - Breakpoints
   - Multi-threaded
   - Multi-process
 - Unit/system
   - Coverage
   - Boundary conditions
 - Mock
   - Black/clear box

II. Distribution

A. Concurrency

 - Thread safety
 - Liveness hazards
 - Deadlock
 - Livelock
 - Synchronization
Locks
- Blocking
- Atomicity

Deployment
- Virtual Tuesday
- Version management

Project management
- Peer review
- Version control
- Ticketing
- Continuous integration
  - Development and production

B. Documentation

III. Optimization

A. Types

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

B. Methods

- Premature
- Profiling

IV. Security

A. Coding

- Defensive programming
  - Input validation
B. Defense against the future

- Agile programming
- Advances in cryptography

C. Best practices

- Backend vs. frontend verification
- User-Interface security
- Fortify taxonomy