Course Summary

Presentations Thursday
8 min — on laptop
Important Features
Practice/Plan
One partner “drive” while the other person talks
Theme: Design & Communication

⇒ Design for Communication
  modules communicate
  ≈ shared objects
  method calls
  listeners
  streams
  sockets

• abstraction & specification
• orthogonality
• extensibility
• modularity
• clarity/readability
• technical topics: class hierarchies & UML,
  streams
  MVC
  concurrency
  interprocess comm

⇒ Communication about Design
  well-defined interfaces, responsibilities, integration, planning
Labs:

1. Model — what is represented, how, how does it fit together?
   - UML
2. Persistence
3. View — user’s visualization of the model
4. Controllers — user experience (scenarios for design)
   - DIRECT MANIPULATION
5. Interprocess Communication — sockets
   - talked about RMI

Review sheet linked from CSE132 home page

Part I follows labs, Part II — reflects on your experience
Abstraction / Specification
Abstraction Functions

ADTs, later: GUIs

HashTable $t$
$AF(t) = \{ e \mid \exists i, t.a[i] = e \}$

rep. invariants / encapsulation
repOK, testing
Class Hierarchies / Polymorphism / UML

Orthogonality — Comparator (plug-in)

Design Patterns
  - Bridge
  - Composite
  - Factory
  - Flyweight
  - Publish/Subscribe
  - Adapter/Proxy/Wrapper Indirection

Streams / Java IO / File systems
  - Orthogonality
    - Stream types that wrap other streams
  - Human vs. Machine readable formats / Custom vs. standard
Iterator & Visitor Pattern
\/
Composite

User Interface Design & Implementation
- Direct manipulation

Technical issues
View components
Layout managers — orthogonality
Property Change Support
Swing components have models

Controllers
Listener Interfaces
Inner classes — final variables for local and anonymous
EventQueue — handled in a separate thread
Threads & concurrency control

- creating threads
- control interleaving of steps of threads
  - locks \( \Leftarrow \) careful!!!!!
  - synchronized methods & blocks \( \Leftarrow \) careful!
  - deadlock
  - low-level thread memory model
    - volatile

\[ T_1 \xrightarrow{\text{Working Memory}} \xrightarrow{\text{Function}} \xrightarrow{\text{Working Memory}} \xrightarrow{\text{RAM}} T_2 \]

- abstractions for concurrency
  - Bounded Buffer
  - wait-free queue (didn't cover)

In-process Communication

IP/TCP, sockets as support for TCP streams, RMI