



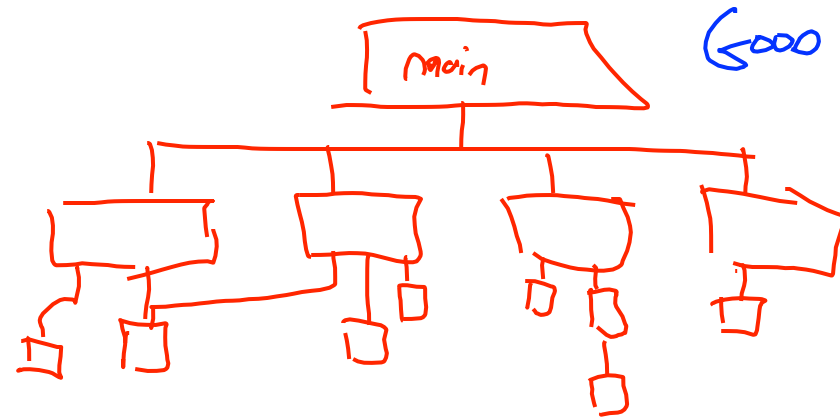
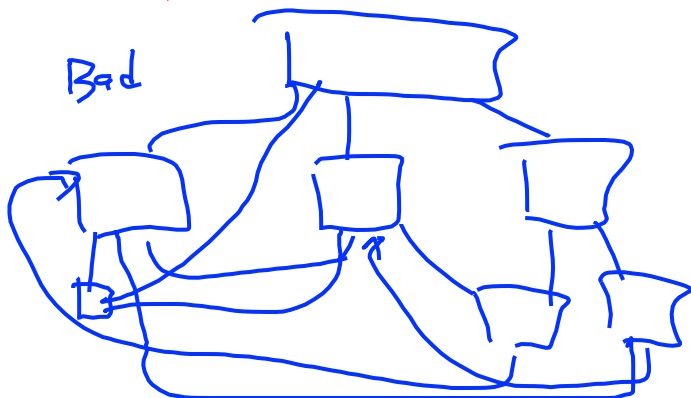
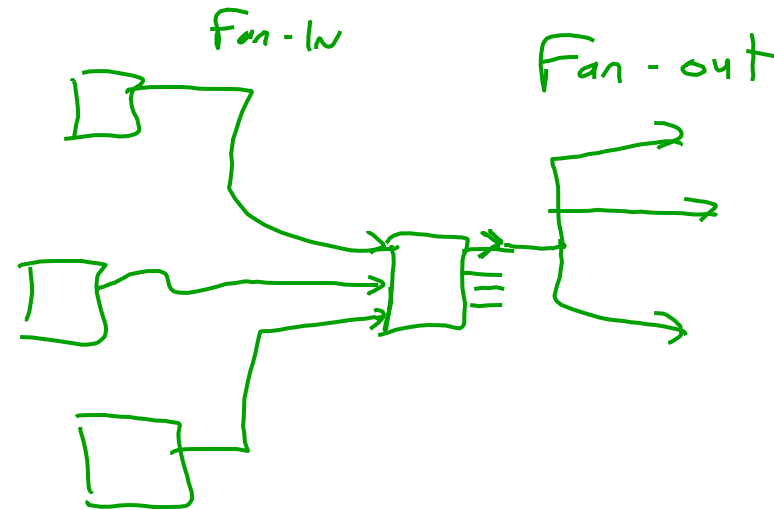
Object/Class Design

Design is hard

- Design is an art, not a science
- Large/infinite design space, not enumerable
- Requirements, constraints, trade-offs, and priorities
- You get better with practice / experience / seeing good design / hearing critiques of others designs

Virtues of a good design (software)

- Manages complexity
- Loose coupling
- Reusability
- Ease of maintenance
- Standard techniques
- Extensibility
- High Fan-in
- Low-to-medium Fan-out



Good Design **Manages Complexity**

- “Seven plus or minus two” (Miller’s Law)
- The goal of all software-design techniques
 - Break complicated problems into simple problems
- Separation of concerns
 - Focus on one at a time

Keep Coupling Loose

- small interfaces (few methods, few arguments/method)
- obvious (interactions through parameter passing)
- flexible

How hard was fourth code review assignment?

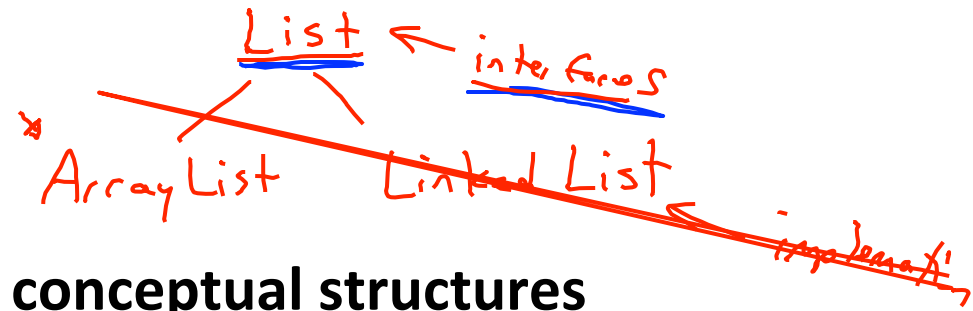
- A) Easy
- B) Moderate
- C) Challenging
- D) Unreasonable

How long did fourth assignment take?

- A) Less than 2 hours
- B) 2 to 4 hours
- C) 4 to 6 hours
- D) 6 to 8 hours
- E) More than 8 hours

a) show data
b) don't show data

Abstract Data Types

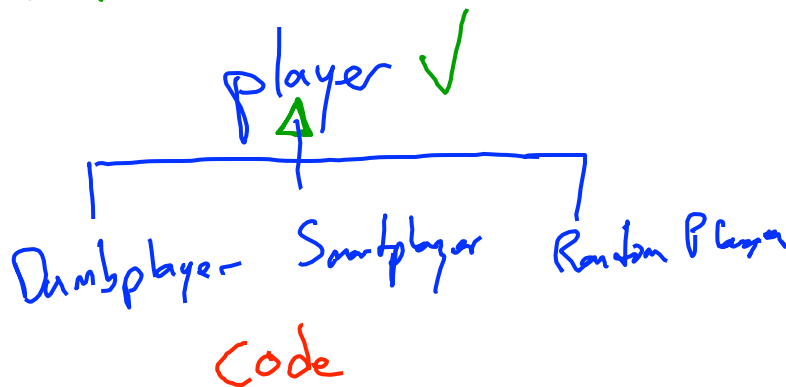


- Define a class based around conceptual structures
 - Encapsulation / information-hiding → coupling
 - Make interfaces more informative (self-documenting)
 - Easier to reason about correctness

■ Treat even simple items as ADTs

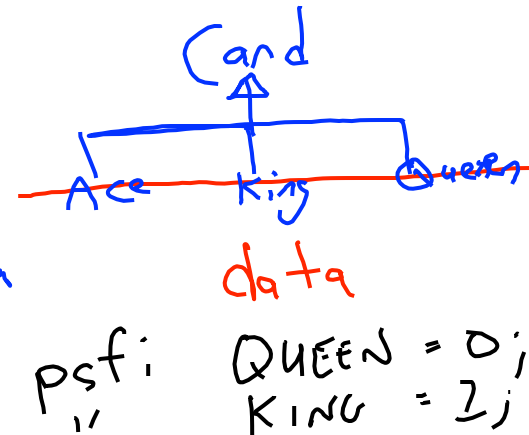
■ Good for extensibility

- use inheritance
- a) player & card
 - b) player
 - c) card
 - d) neither



What objects in AKQ?

Game ↔ Game state
 int card = QUEEN



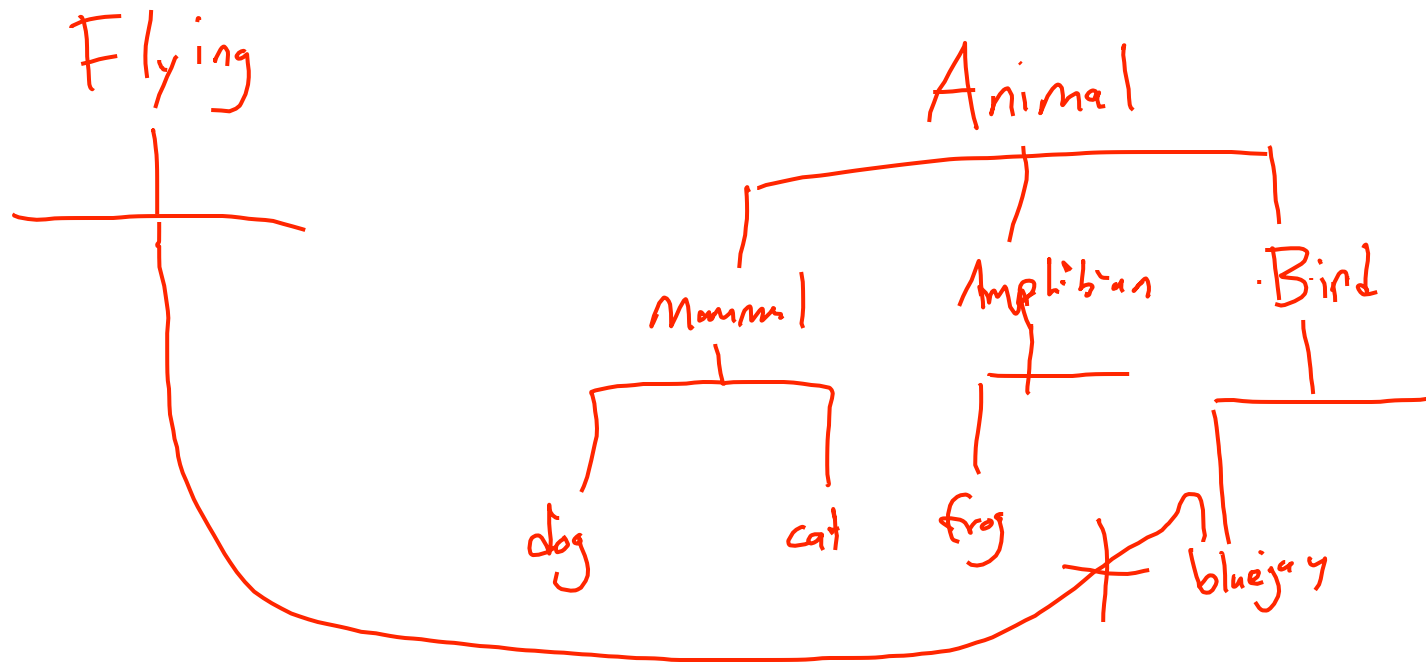
Inheritance can provides 2 things

- Shared interface: ✓
 - Public methods
 - Ensure methods have consistent meaning in derived class
 - Liskov Substitution Principle
- Shared implementation ✓
 - Code in shared super class, not replicated in each derived
 - Could be private data/methods

Inheritance vs. Interfaces

- Inheritance should be a isA relationship
- Interfaces are for capabilities (“mixin”s)

In Java,
single inheritance.



Designing Good Interfaces

- Sufficiently Expressive
- General
- Minimal

Player Strategy

Communicate dealt card

bet?

call?

what happened? what they did, what I did,
how much money, who won it,
what card they had.

Keys to design nirvana?

- **Treat design as a wicked, sloppy, heuristic process.**
- **Don't settle for the first design that occurs to you.**
- **Collaborate.**
- **Strive for simplicity.**
- **Prototype when you need to.**
- **Iterate, iterate, and iterate again.**
- **You'll be happy with your designs.**

(Code Complete, p. 119)

For Tuesday's Lecture

- Read Ch. 9 of the Textbook