Design methods

- Understand the tools = understand the method
- Literally hundreds of SW design tools / methods
- Today: OO, ADT's
OO: A top-down method

- **OO as design method** *(vs OOP)*
- **Identify**
  - nouns = objects
  - verbs = methods
  - adverbs = fields
- **Class hierarchy**
Fundamental OO ideas

- Encapsulation
- Implicit reference
- **Dynamic dispatch**
- Inheritance subtyping
- Universality
Brief history of OO

- Simula '67: simulation objs
- Smalltalk '80: “everything is obj”, Kay “kids' lang”
- Object Pascal, C: objs for programmers
- C++: efficient objs
- Java, C#: modern objs
CRC Method

- Create “CRC cards”
  - One per class
  - CRC
    - Classes (super/sub)
    - Responsibilities (methods)
    - Collaborations (related)
- Trace to requirements?
CRC “classes”

- Note class commonalities
- Factor into hierarchy
- Leads to shallow inheritance trees
- Crucial role of subtypes in collection classes
CRC “collaboration”

- Apparently normally caller/callee
- Could be almost anything?
- Used to verify minimality, soundness
Critique of OO design

- Some nouns hard to represent as objects
  - abstract qualities / ideas
  - collective nouns
- Inheritance subtyping is not enough, too much
- Design sprawl / “spaghetti objects”
ADT's: A bottom-up method

- Abstraction as design method: (vs functional programming)
- Identify abstract values (nouns)
- Identify abstract operations (verbs)
Fundamental ADT ideas

- Information hiding
- Algebraic structure
- “Primitivization”
- Types as sets of values
- Parametric polymorphism
Brief history of ADT

- **Abstract Data Types:** Parnas “info hiding”
- **Algebraic Data Types:** formal method
- **OO as instance of**
- **FP as implementation of**
- **CLU '75 (Liskov)**
Design by ADT

- Identify
  - nouns in reqs
  - intrinsic representation
  - extrinsic interface
  - intrinsic ops (verbs)
- Check for closure,
  type safety
Critique of ADT

- Abstract values are by definition **immutable**
- Bottom-up style leads to redundancy
- Parametric polymorphism is hard to learn, use
“No Silver Bullet”

- Brooks' famous essay
- These and other techniques are powerful
- This is the engineering step, so don't expect to automate it