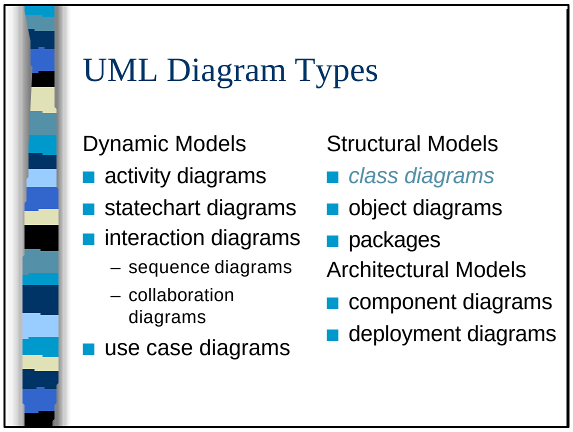


UML Diagram Types

Dynamic Models

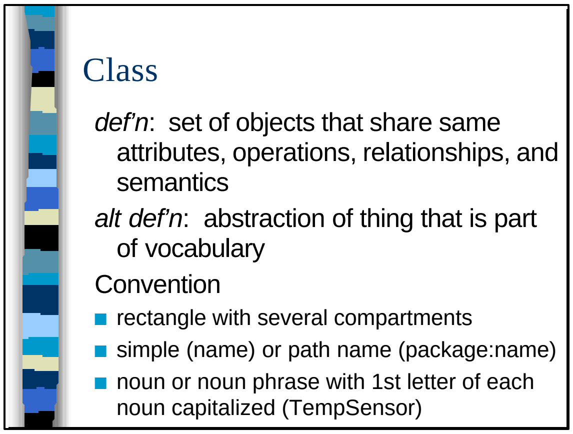
- activity diagrams
- statechart diagrams
- interaction diagrams
  - sequence diagrams
  - collaboration diagrams
- use case diagrams

Structural Models

- *class diagrams*
- object diagrams
- packages

Architectural Models

- component diagrams
- deployment diagrams



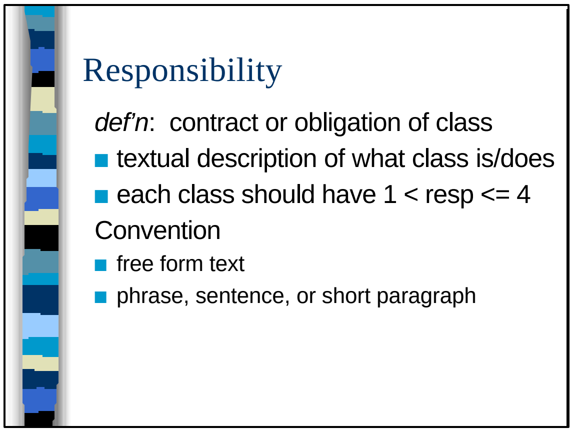
Class

def'n: set of objects that share same attributes, operations, relationships, and semantics

alt def'n: abstraction of thing that is part of vocabulary

Convention

- rectangle with several compartments
- simple (name) or path name (package:name)
- noun or noun phrase with 1st letter of each noun capitalized (TempSensor)



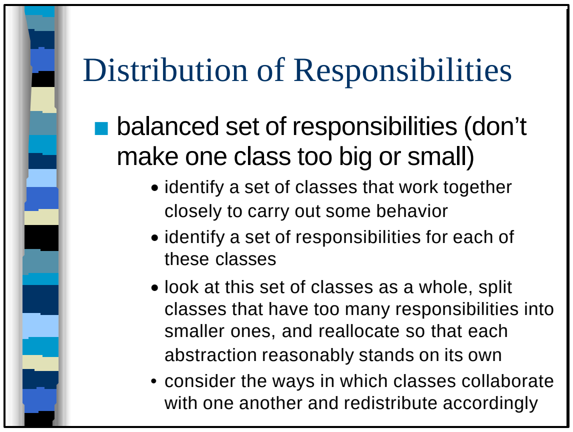
Responsibility

def'n: contract or obligation of class

- textual description of what class is/does
- each class should have 1 < resp <= 4

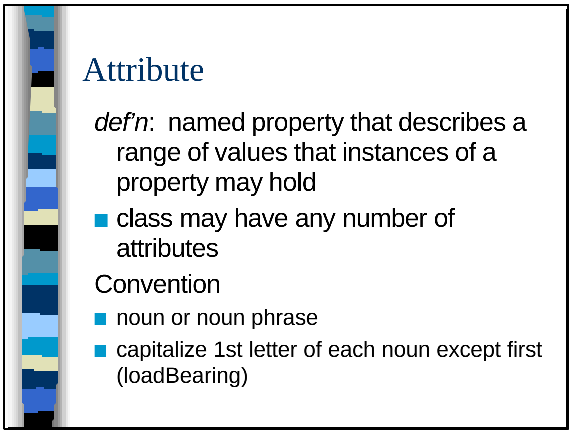
Convention

- free form text
- phrase, sentence, or short paragraph



Distribution of Responsibilities

- balanced set of responsibilities (don’t make one class too big or small)
  - identify a set of classes that work together closely to carry out some behavior
  - identify a set of responsibilities for each of these classes
  - look at this set of classes as a whole, split classes that have too many responsibilities into smaller ones, and reallocate so that each abstraction reasonably stands on its own
  - consider the ways in which classes collaborate with one another and redistribute accordingly



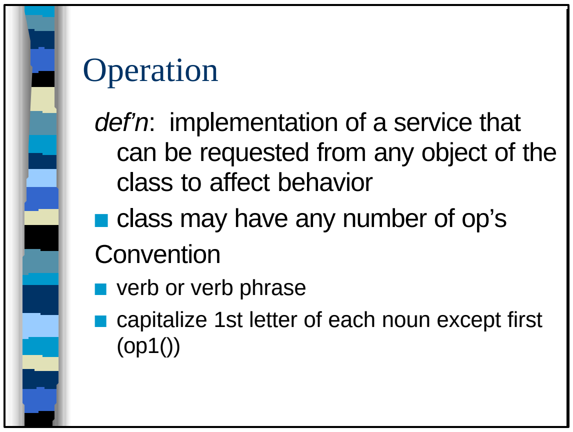
Attribute

def’n: named property that describes a range of values that instances of a property may hold

- class may have any number of attributes

Convention

- noun or noun phrase
- capitalize 1st letter of each noun except first (loadBearing)



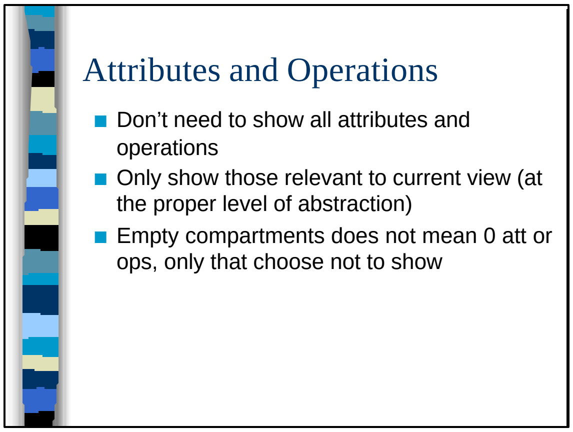
Operation

def’n: implementation of a service that can be requested from any object of the class to affect behavior

- class may have any number of op’s

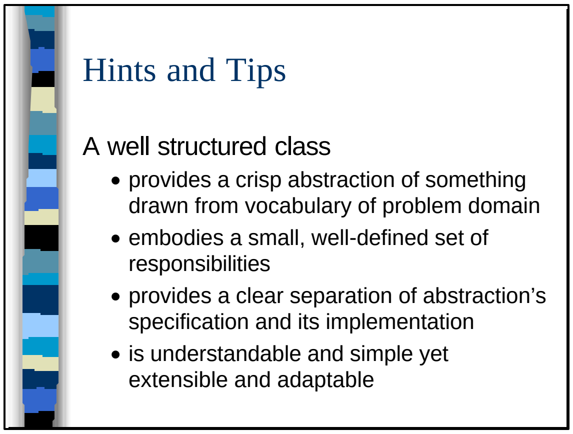
Convention

- verb or verb phrase
- capitalize 1st letter of each noun except first (op1())



Attributes and Operations

- Don’t need to show all attributes and operations
- Only show those relevant to current view (at the proper level of abstraction)
- Empty compartments does not mean 0 att or ops, only that choose not to show



Hints and Tips

A well structured class

- provides a crisp abstraction of something drawn from vocabulary of problem domain
- embodies a small, well-defined set of responsibilities
- provides a clear separation of abstraction’s specification and its implementation
- is understandable and simple yet extensible and adaptable