

Design Class Diagrams

- **Classes**
 - Domain
 - Implementation
- **Associations**
 - Navigability
 - Aggregation
 - Containment
- **Attributes**
 - Type information
- **Operations**
 - Method names
 - Interfaces
 - Accessors & mutators
- **Dependencies**

Develop in parallel with interaction diagrams.

Navigability

- **Association**
 - Connects objects of two (?) classes
 - E.g., class "A" and class "B"
- **Navigation**
 - Can you get from "A" to "B"?
 - Answer determined by navigability

What does this mean in a practical implementation?

Dependency

- **Related classes**
 - E.g., "A" and "B"
- **Does "A" know about "B"?**
 - Does A's code need access to B's definition?
 - Drives "#include" code generation

Class Member Details

- **Visibility**
 - Public, protected, private
 - Implementation?
- **Data members**
 - Type, initial value, accessor/mutator
- **Member functions (methods)**
 - Return value, arguments
- **Code generation properties**

Design Exercise with Rose

- **Faculty schedule system**
- **Pick one of these system events**
 - Assign section to professor
 - Calculate a professor's workload
- **Choose design and justify**
- **Document**
 - Collaboration diagram
 - Design class diagram
