Interactions

- Class diagrams illustrate *static* aspects of the system design.
- To represent *dynamic* (i.e., run time) aspects of the system we need to show *objects* and *events*.
- Interaction diagrams illustrate interaction between objects.
  - Sequence diagrams
    - A vertical line for each object involved.
    - Time runs top to bottom.
    - Events (method call) indicated by arrows.
  - Communication diagrams (a.k.a. collaboration diagrams)
    - Layout for clarity
    - Message/call sequence indicated by numbers.
    - Nested calls indicated by “.” (e.g., 1.1, 1.2 . . . ).
Interactions

Control Flow in Sequence Diagrams

- **loop** [for all thingies]
- **opt** [condition]
- **alt** [condition]
  - [other condition]
  - [else]
- **ref** name of interaction (args)
Formerly called collaboration diagrams

Communication Diagram

object name : class

role name

1: message ()

: class