



## System Design: Static Diagram

- □ Class Diagram (System modeling, not only about the world)
  - □ *system level / complete view*
  - □ *DataType: Use whatever you know + enumeration*
  - □ *Multiplicity for all places: \* , 1, 0..100, 0, 3..\**
  - □ *Class*
    - □ Name (capital letter)
    - □ multiplicity: number of instance
      - □ \* (default), 1 (singleton)
    - □ attributes (state of the obj)
      - □ scope : instance (default) / class (static)
      - □ visibility: + (public), #(protected), - (private) , ~ (package)
        - □ language specific / use the minimum level (encapsulation)
      - □ multiplicity: 1 (default), \*, 0..5 (collection)
    - □ operation (interface)
      - □ scope : instance (default) / class (static)
    - □ object identity : physical identity (Jordan != Jordan)
  - □ *Association*
    - □ Name : verb phrase + arrow (optional)
    - □ role name (optional)
    - □ multiplicity (compulsory)
    - □ navigability
    - □ types: binary (most cases), self
  - □ *Generalization*
    - □ is-a relation
    - □ substitutability
    - □ polymorphism
    - □ abstract class (slant font)
  - □ *Inheritance (a process, invisible)*
    - □ attributes and operations
    - □ override operations
  - □ *Aggregation*
    - □ A strong form of association to show the containment relationships.
      - □ Diamond point to the container
    - □ antisymmetry: can not contain self. (you can not be your dad)
    - □ transitivity: 1 < 2, 2 < 3 --> 1 < 3
  - □ *Composition*
    - □ A strong form of Aggregation to show the dependency.
    - □ Parts can belong to one container at a time
    - □ parts are destroyed once the container is destroyed.

## Object Diagram

*a snapshot / scenarios*

*check the validity of class diagram*

: + underlined

non-capital letter

*Link : Association*

underlined