

UML

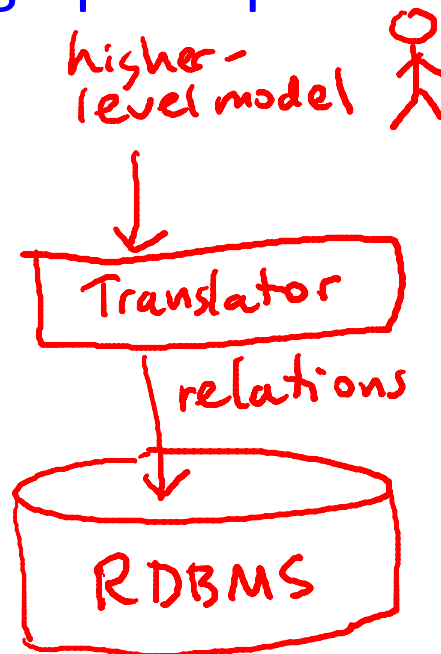
---

UML Data  
Modeling

# Data Modeling

How to represent data for application

- Relational model – with design principles
- XML
- Database design model
  - Not implemented by system
  - Translated into model of DBMS



## Higher-Level Database Design Models

- *Entity-Relationship Model (E/R)*
- *Unified Modeling Language (UML)*

Data modeling subset

- Both are graphical
- Both can be translated to relations automatically  
Or semi-automatically

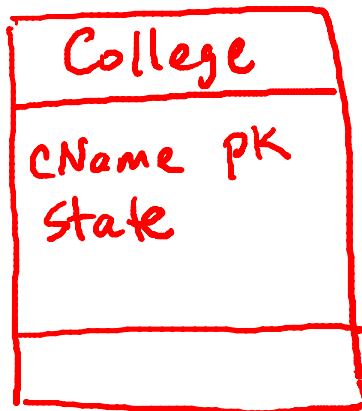
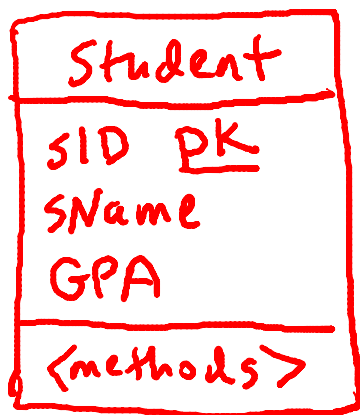
## UML Data Modeling: 5 concepts

- (1) Classes
- (2) Associations
- (3) Association Classes
- (4) Subclasses
- (5) Composition & Aggregation

## UML Data Modeling: **Classes**

Name, attributes, methods

*For data modeling: add "pk", drop methods*

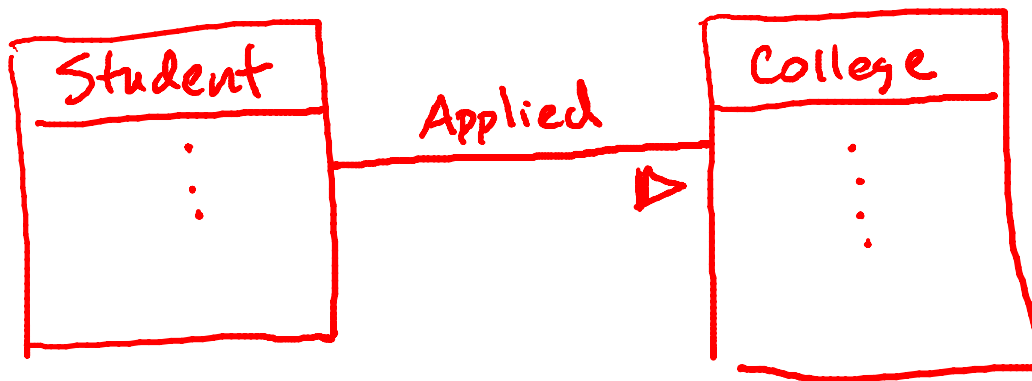


## UML Data Modeling: 5 concepts

- ✓ (1) Classes
- (2) Associations
- (3) Association Classes
- (4) Subclasses
- (5) Composition & Aggregation

## UML Data Modeling: **Associations**

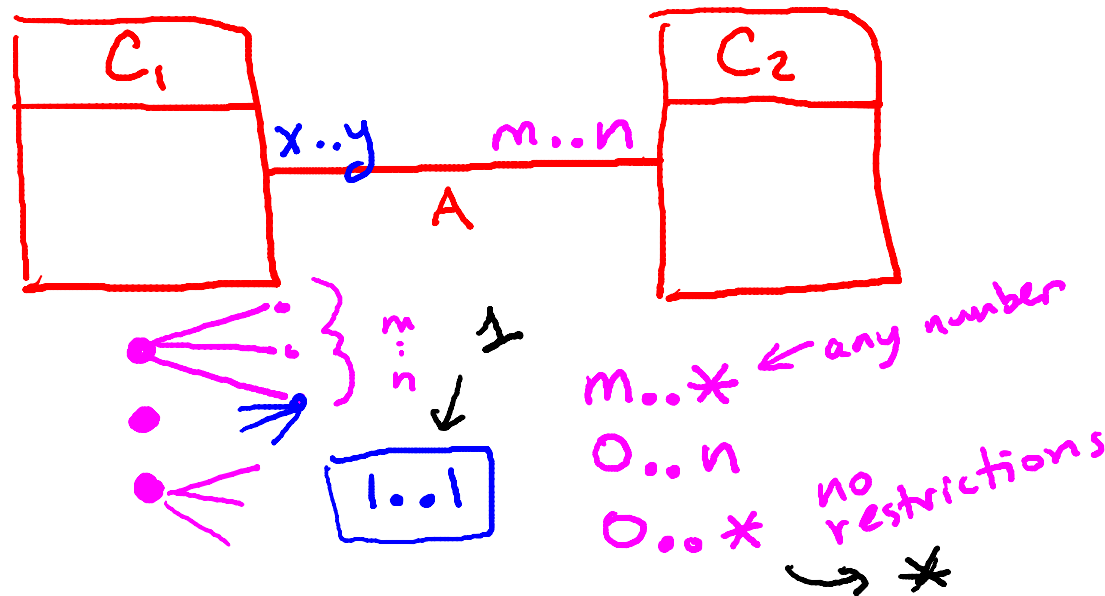
Relationships between objects of two classes



# Multiplicity of Associations

Relationships between objects of two classes

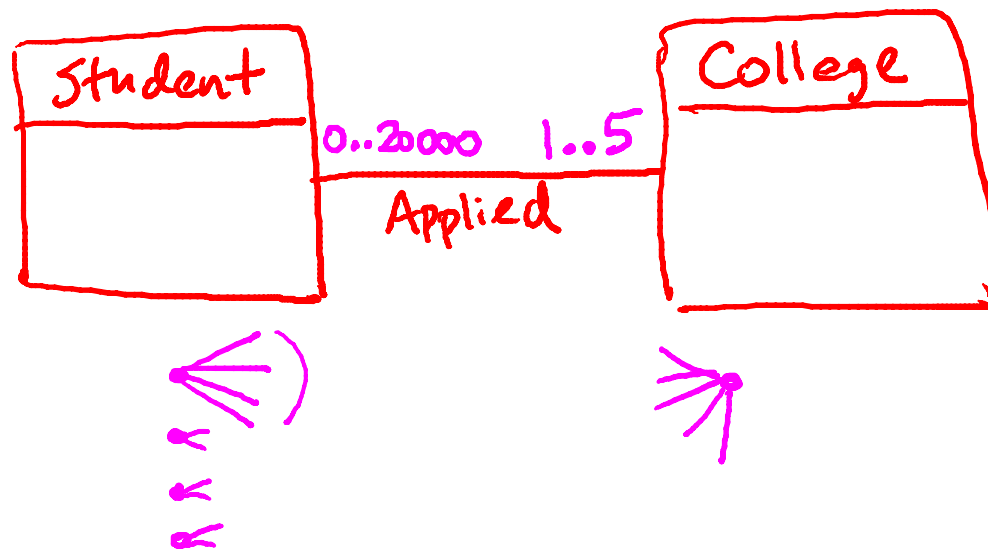
*Each object of class  $C_1$  is related to at least  $m$  and at most  $n$  objects of class  $C_2$*





## Multiplicity of Associations: Example

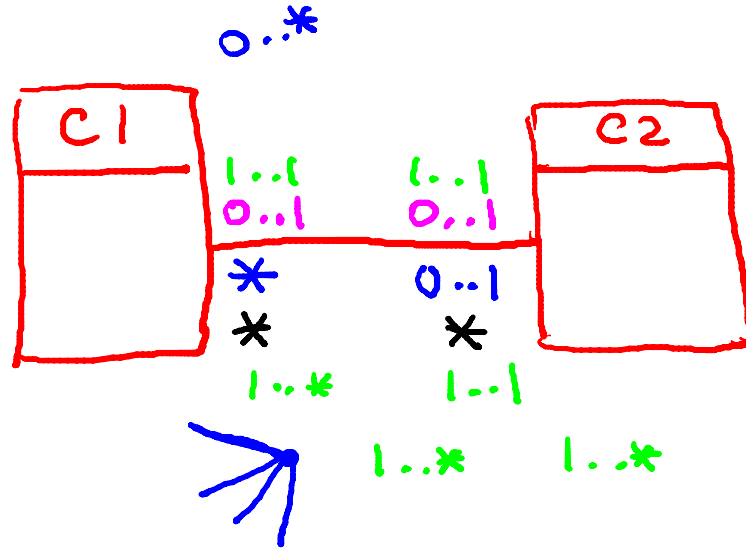
Students must apply somewhere and may not apply to more than 5 colleges. No college takes more than 20,000 applications.



## Multiplicity of Associations: Types of Relationships

- ■ One-to-One *Complete*
- ■ Many-to-One
- ■ Many-to-Many
- ■ Complete

*Default  
1..1 1..1*

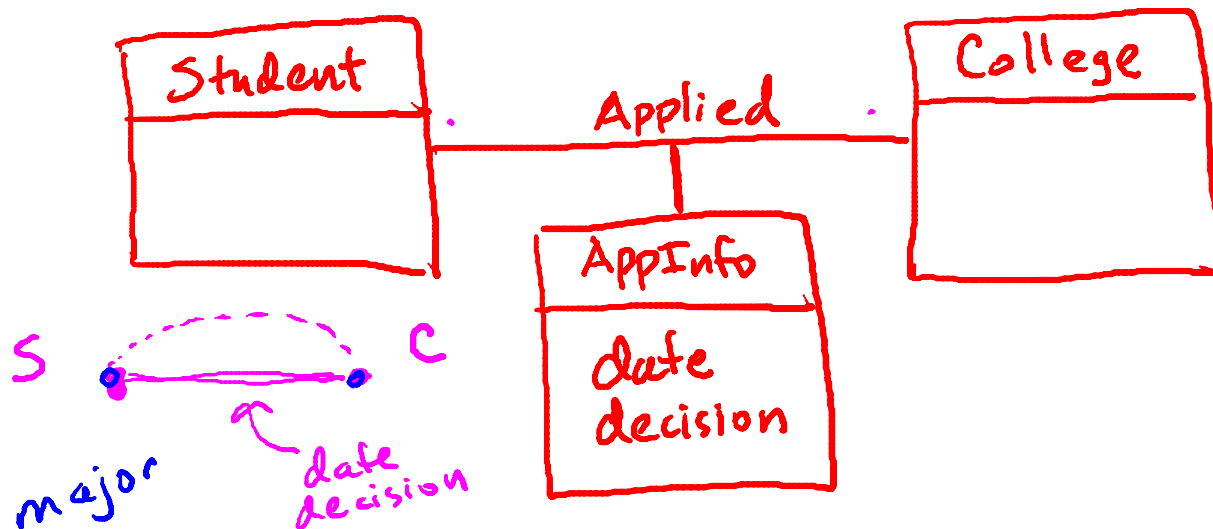


## UML Data Modeling: 5 concepts

- ✓ (1) Classes
- ✓ (2) Associations
- (3) Association Classes
- (4) Subclasses
- (5) Composition & Aggregation

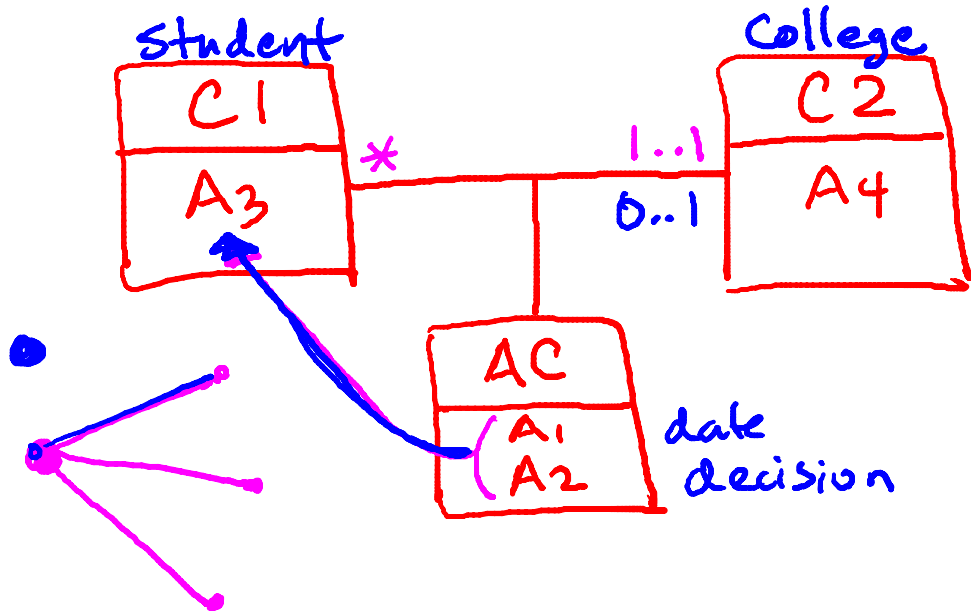
## UML Data Modeling: Association Classes

Relationships between objects of two classes,  
*with attributes on relationships*



# Eliminating Association Classes

Unnecessary if 0..1 or 1..1 multiplicity

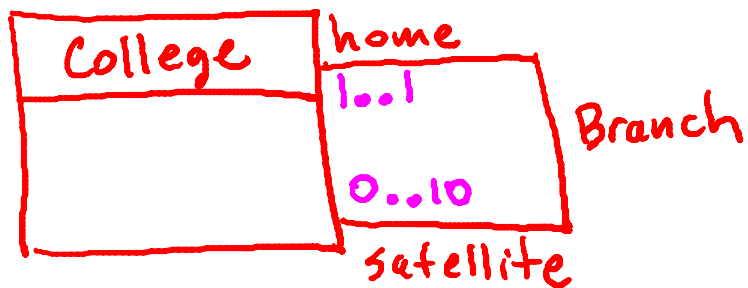
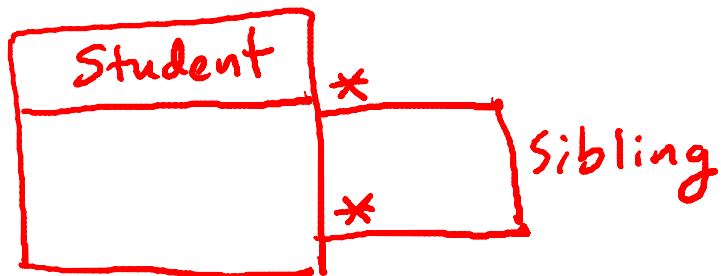


## **Self-Associations**

Associations between a class and itself

## Self-Associations

Associations between a class and itself

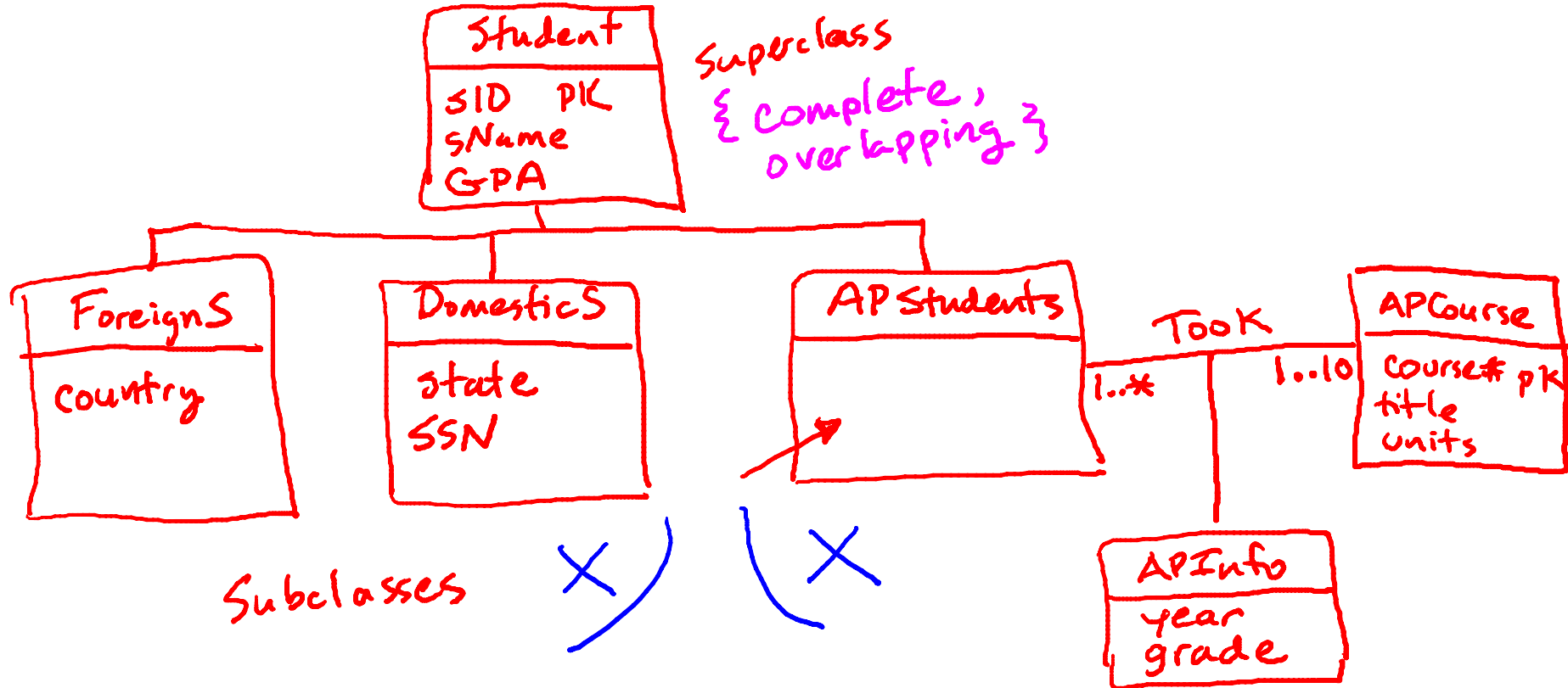


## UML Data Modeling: 5 concepts

- ✓(1) Classes
- ✓(2) Associations
- ✓(3) Association Classes
- (4) Subclasses
- (5) Composition & Aggregation



## UML Data Modeling: Subclasses



## Subclass Terminology & Properties

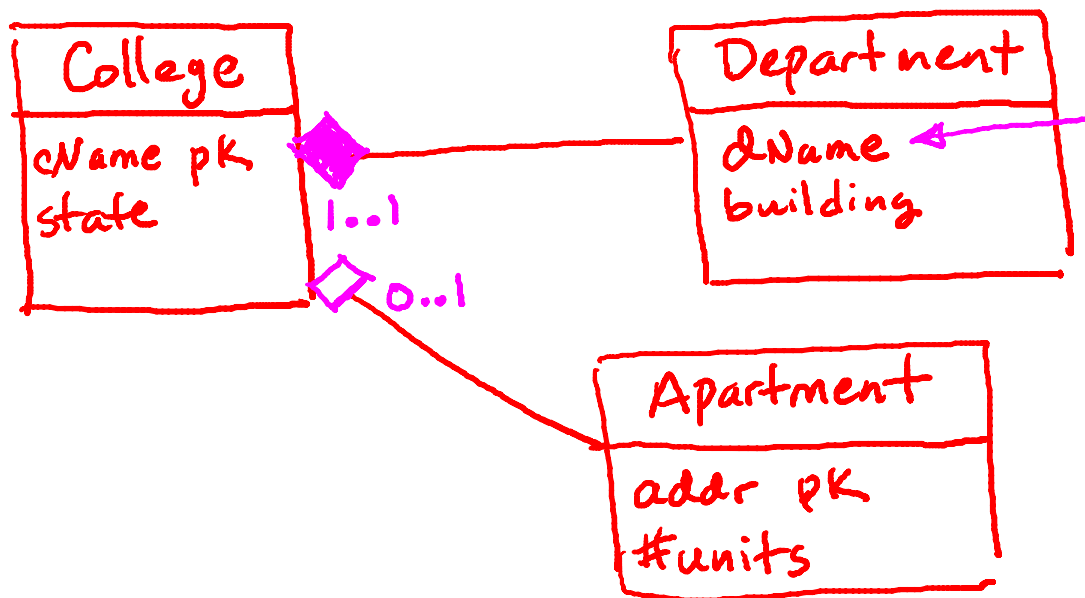
- *Superclass = Generalization* ✓
- *Subclass = Specialization* ✓
- Incomplete (Partial) vs. Complete  
*every obj. in at least subclass*
- Disjoint (Exclusive) vs. Overlapping  
*↳ .. at most...*

## UML Data Modeling: 5 concepts

- ✓ (1) Classes
- ✓ (2) Associations
- ✓ (3) Association Classes
- ✓ (4) Subclasses
- (5) Composition & Aggregation

## UML Data Modeling: Composition & Aggregation

Objects of one class belong to objects of another class



## Higher-Level Database Design

- *Unified Modeling Language (UML)*

  - Data modeling subset

- Graphical

- 5 concepts

  - (1) Classes

  - (2) Associations

  - (3) Association Classes

  - (4) Subclasses

  - (5) Composition & Aggregation

- ❖ Can be translated to relations automatically