Class Diagram Design and Code Generation

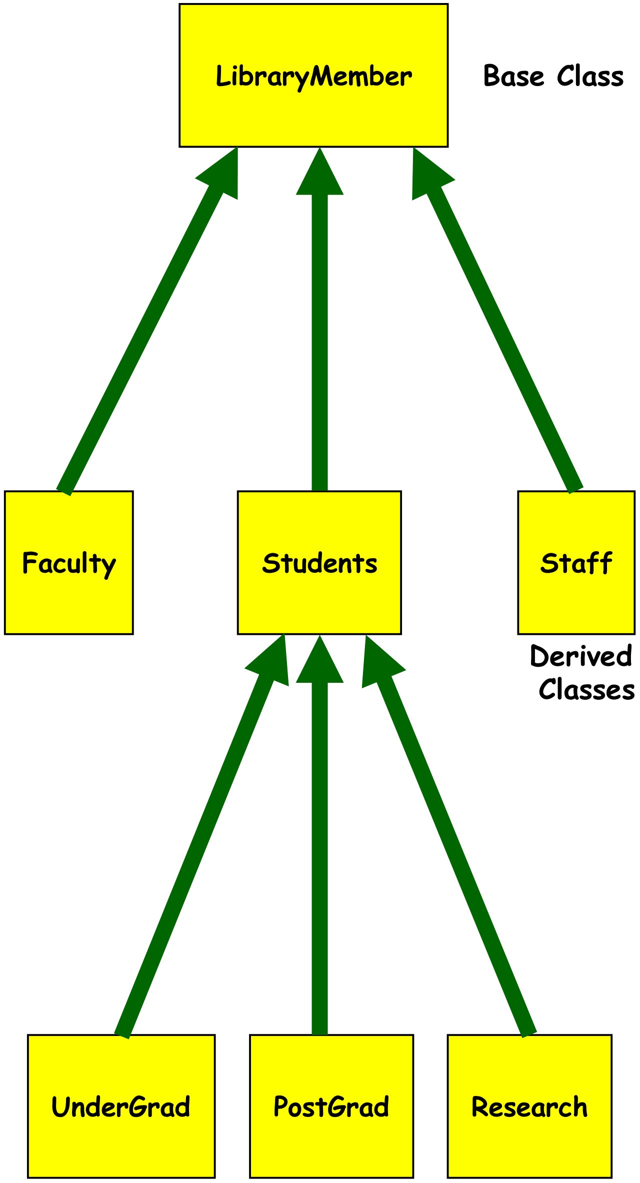
Lect 2—3
8-8-2023

What Are the Different Types of Relationships That May Exist Among the Classes in an OO Program?

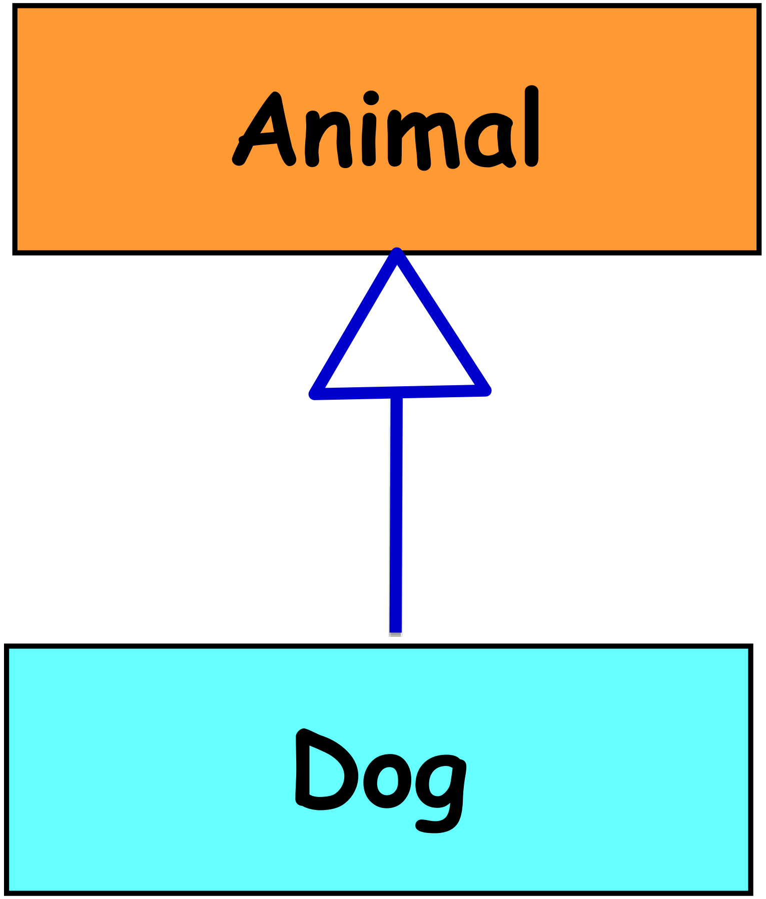
- Four types of class relationships:
  - **Inheritance**
  - **Association**
  - **Aggregation/Composition**
  - **Dependency**

Inheritance

- Allows us to define a new class (derived class) by extending an existing class (base class).
  - Represents generalization -specialization relationship.
  - Allows redefinition of the existing methods (method overriding).



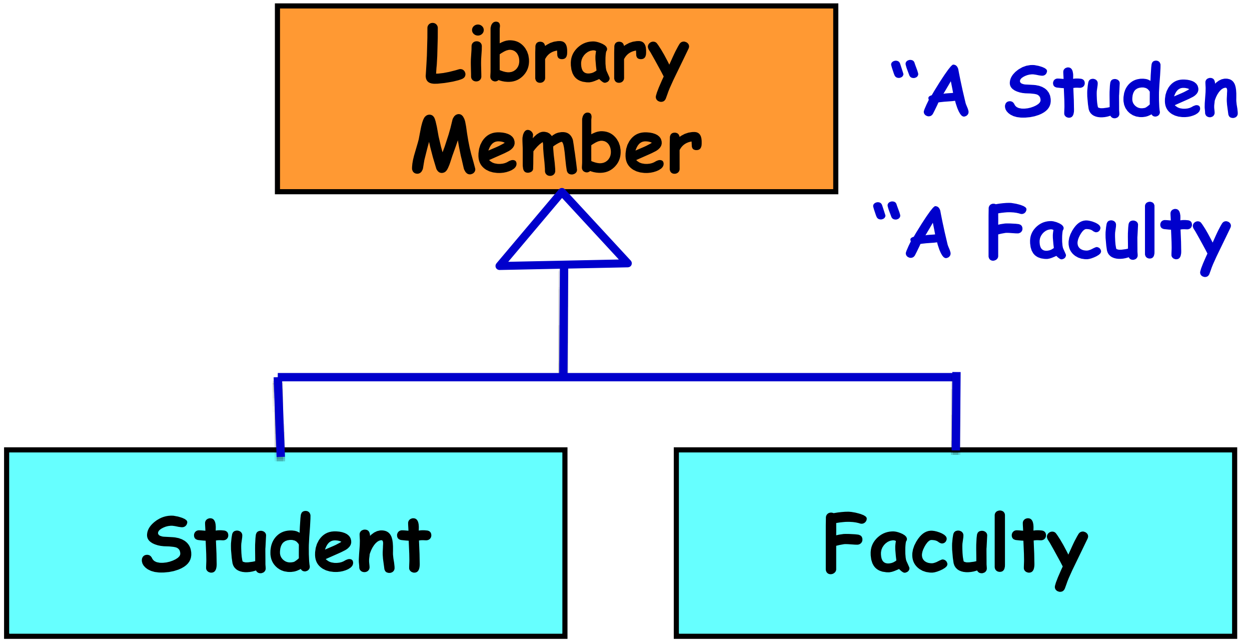
Inheritance Example



“A Dog <u>ISA</u> Animal”

Hmmm… not really…

Inheritance: One More Example

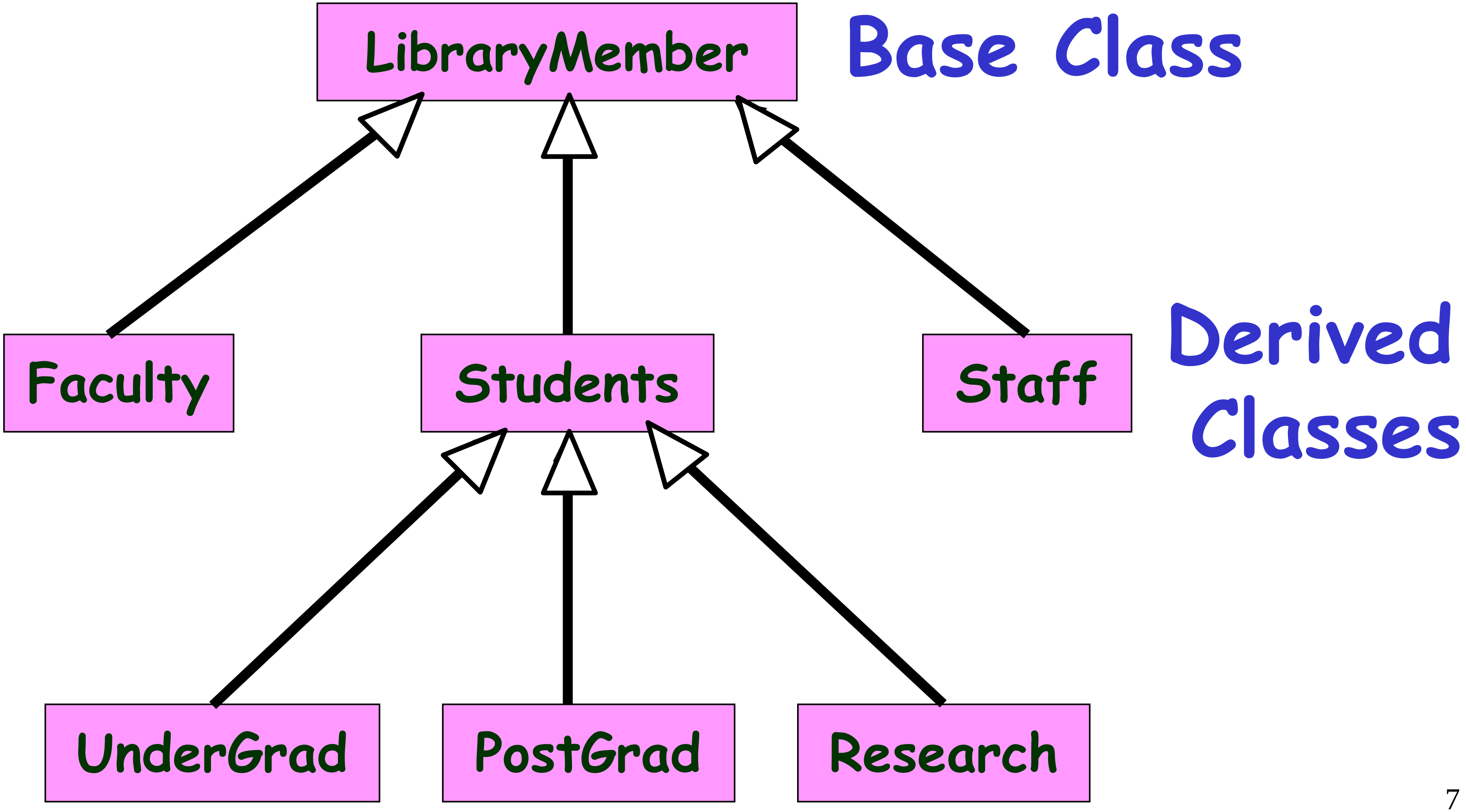


“A Student *ISA* Library Member”

“A Faculty *ISA* Library Member”

Inheritance: Semantics

- Lets a subclass inherit attributes and methods of a base class.



LibraryMember

Faculty

Students

Staff

UnderGrad

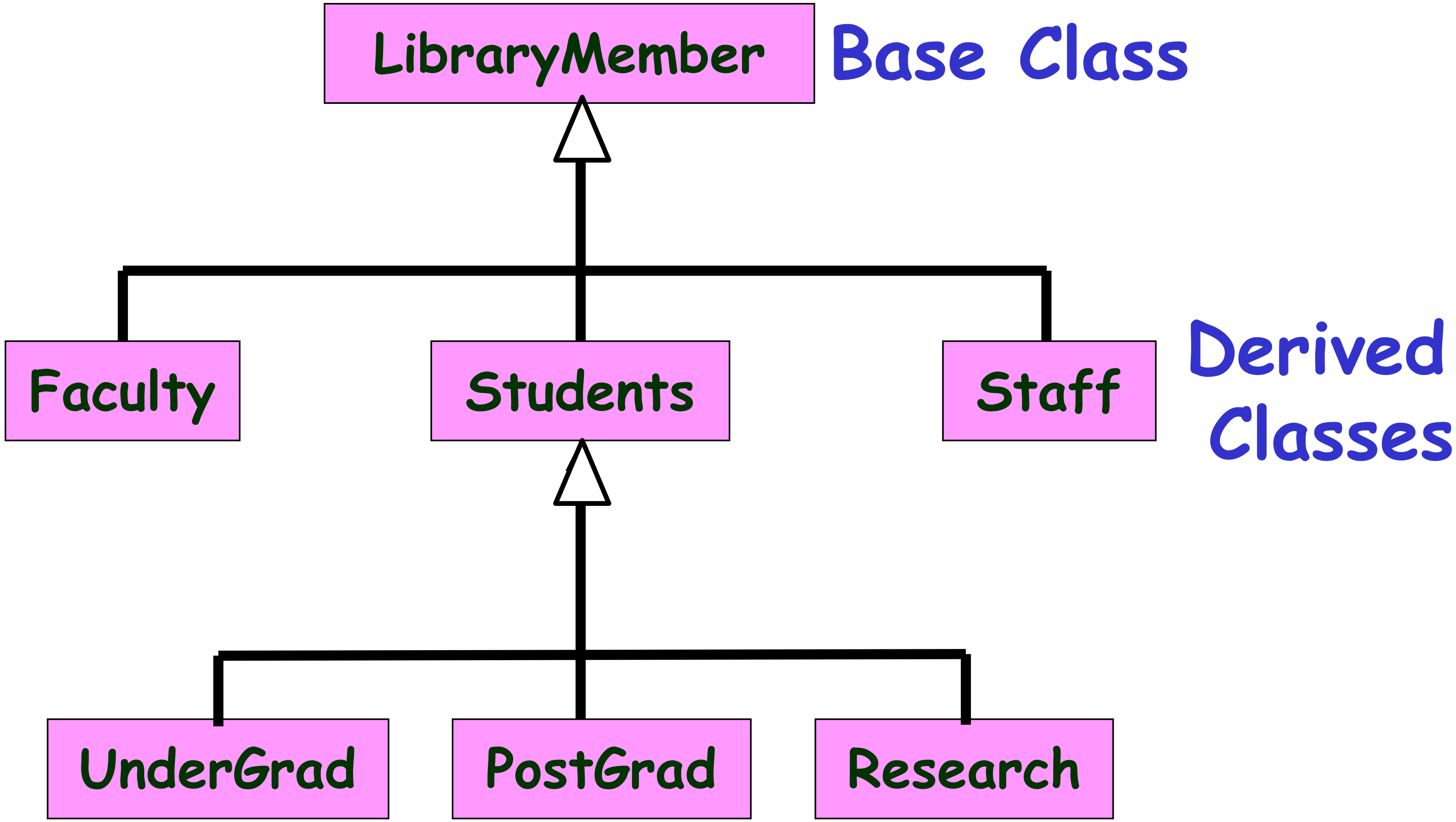
PostGrad

Research

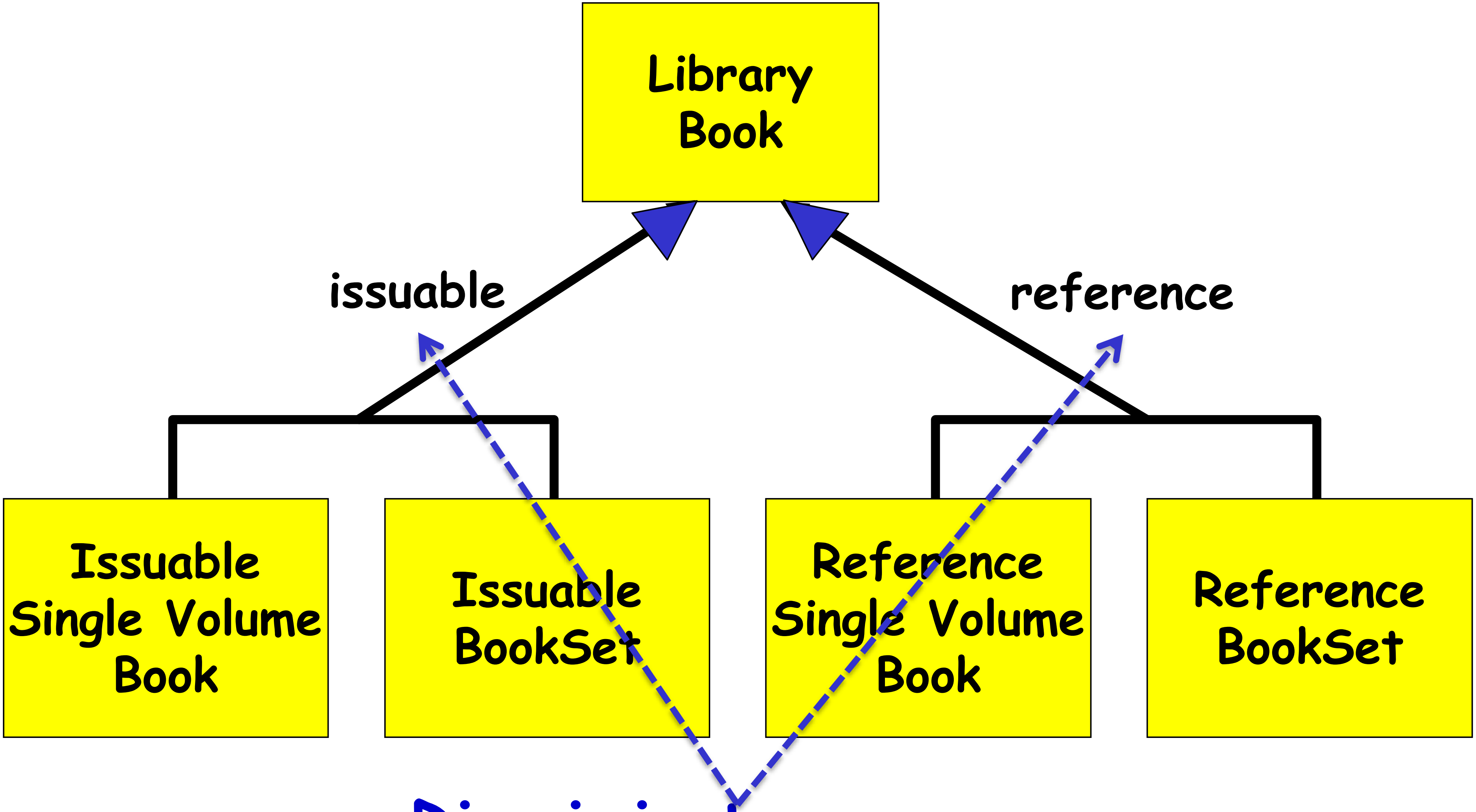
Base Class

Derived Classes

Inheritance: An Alternate Representation

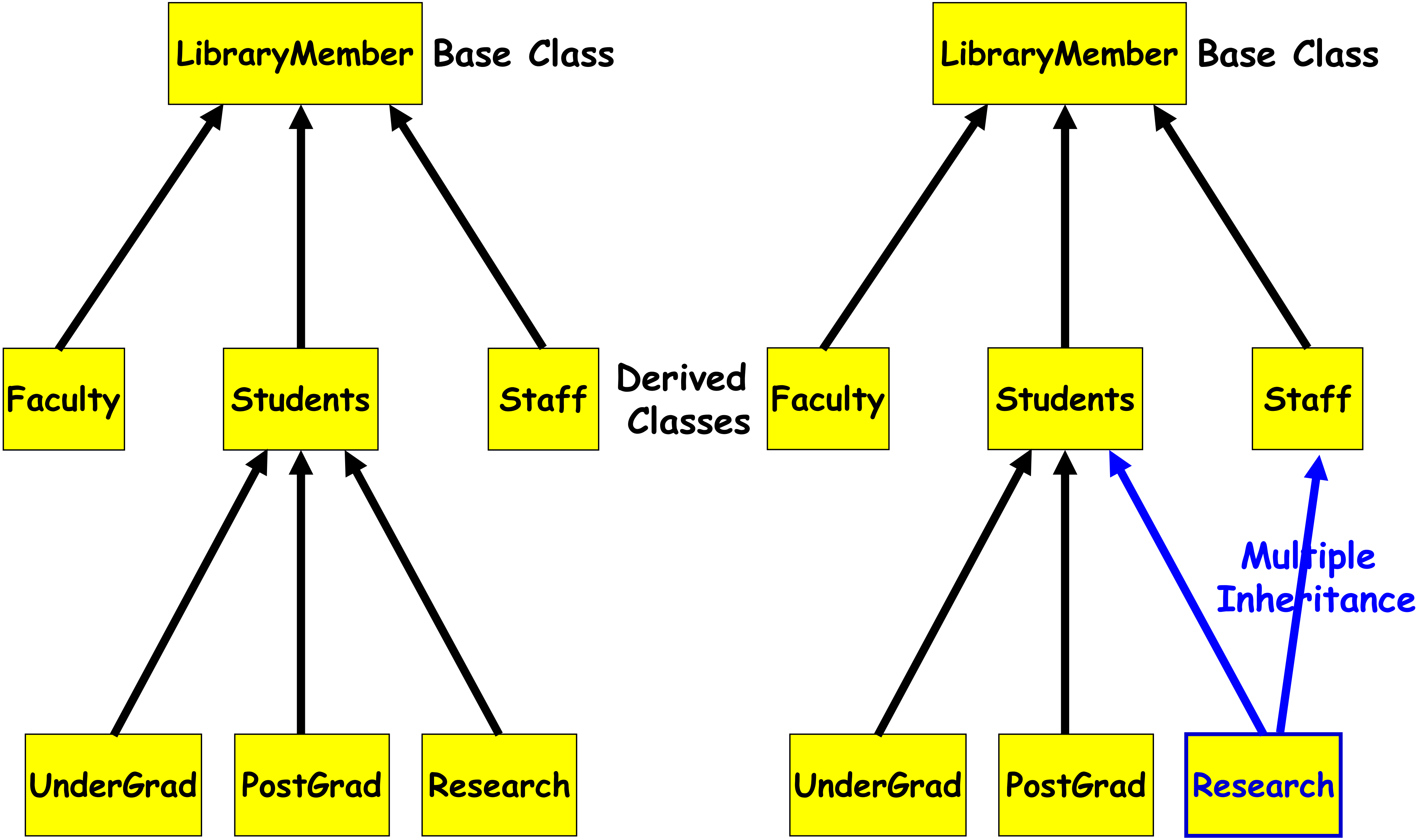


Inheritance Example



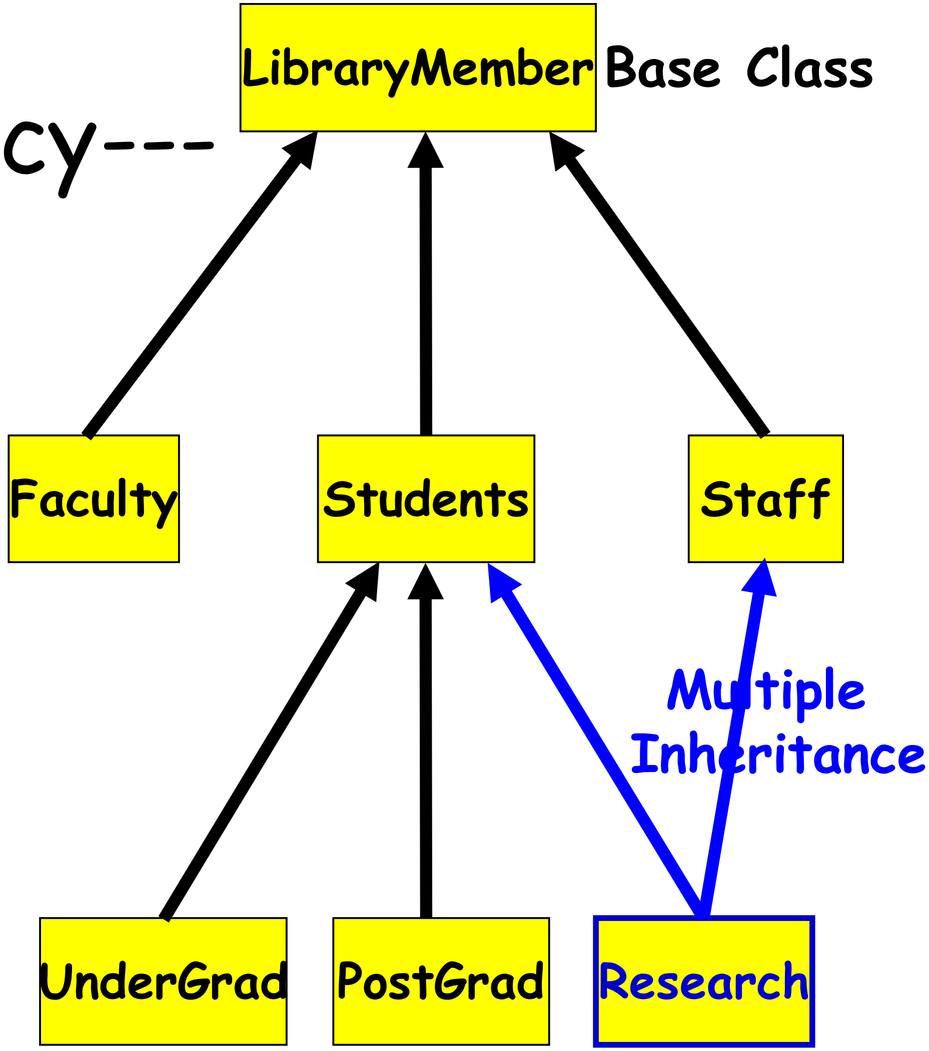
Discriminator: allows to group subclasses into clusters that correspond to a semantic category.

Multiple Inheritance



Multiple Inheritance: Can Cause Repeated Inheritance…

- May lead to inconsistency--- very difficult bugs
- In C++ handled by using virtual base class
- **In Java handled by using Interface class**

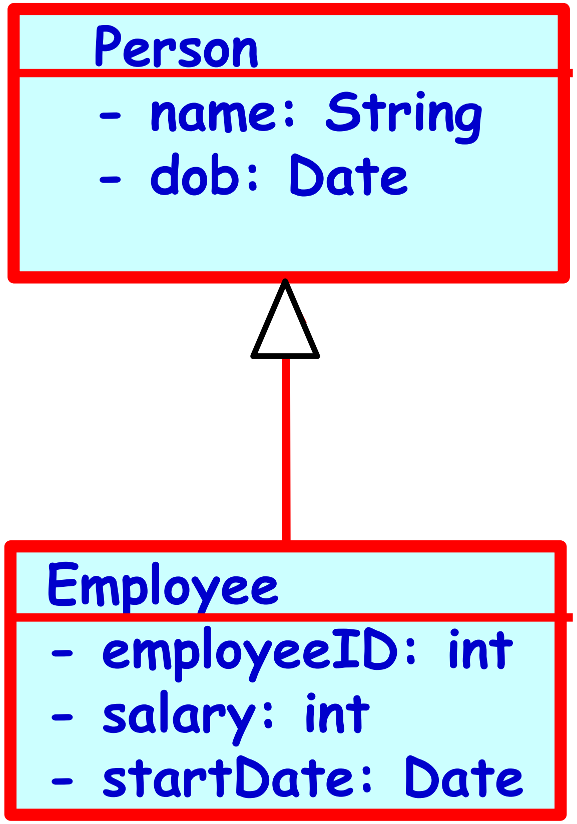


Inheritance Implementation in Java

- Inheritance is declared using the "extends" keyword
  - Even when no inheritance is defined, in Java a class implicitly extends a class called Object.

```
class Person{
  private String name;
  private Date dob;
  ...
}
```

```
class Employee extends Person{
  private int employeeID;
  private int salary;
  private Date startDate;
  ...
}
```



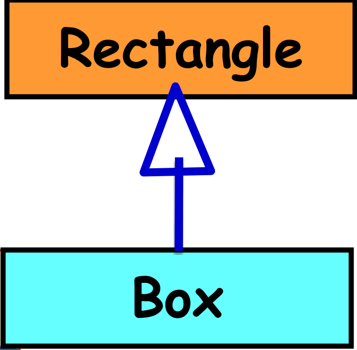
Person
- name: String
- dob: Date

Employee
- employeeID: int
- salary: int
- startDate: Date

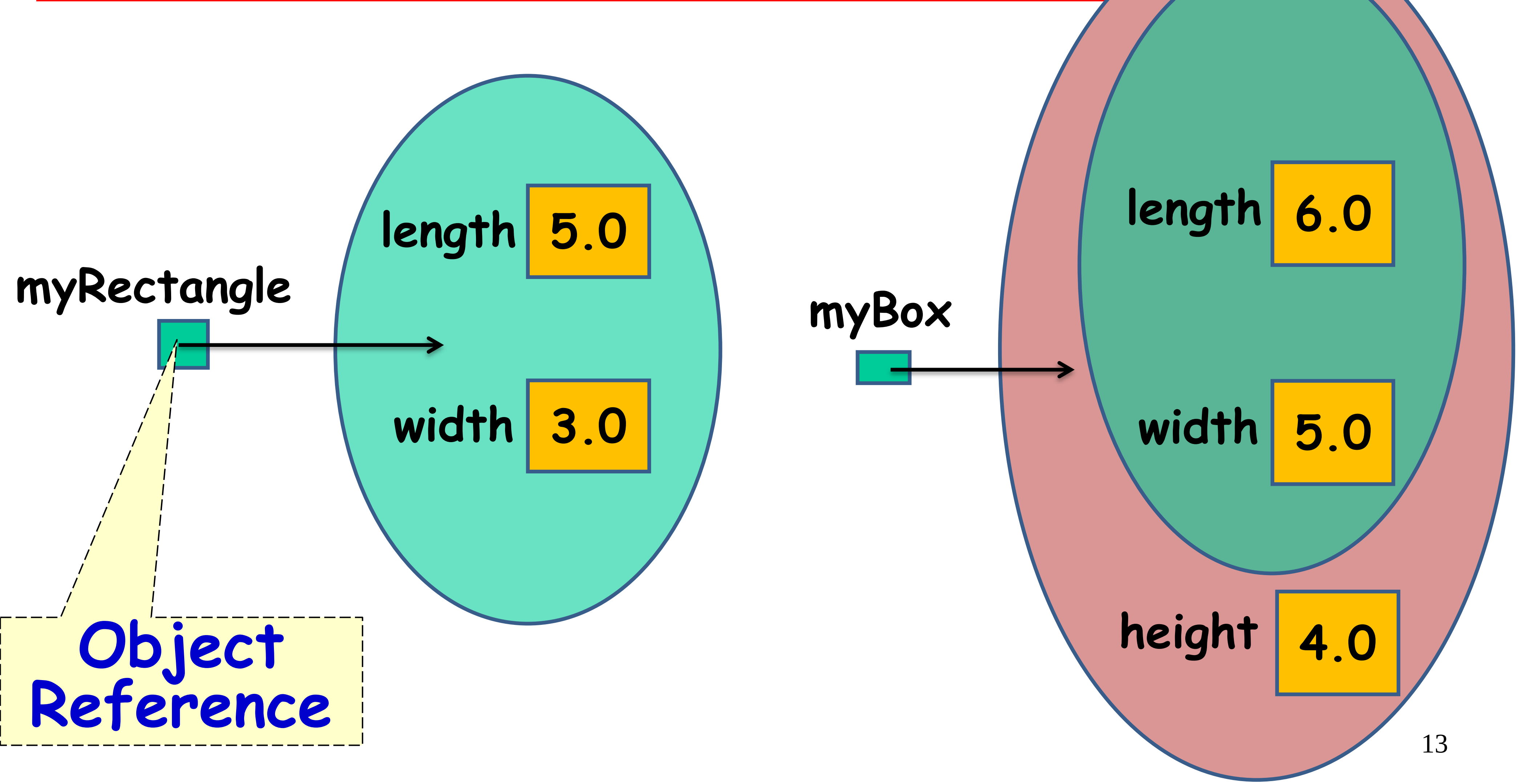
Object Reference

```
Employee anEmployee = new Employee();
```

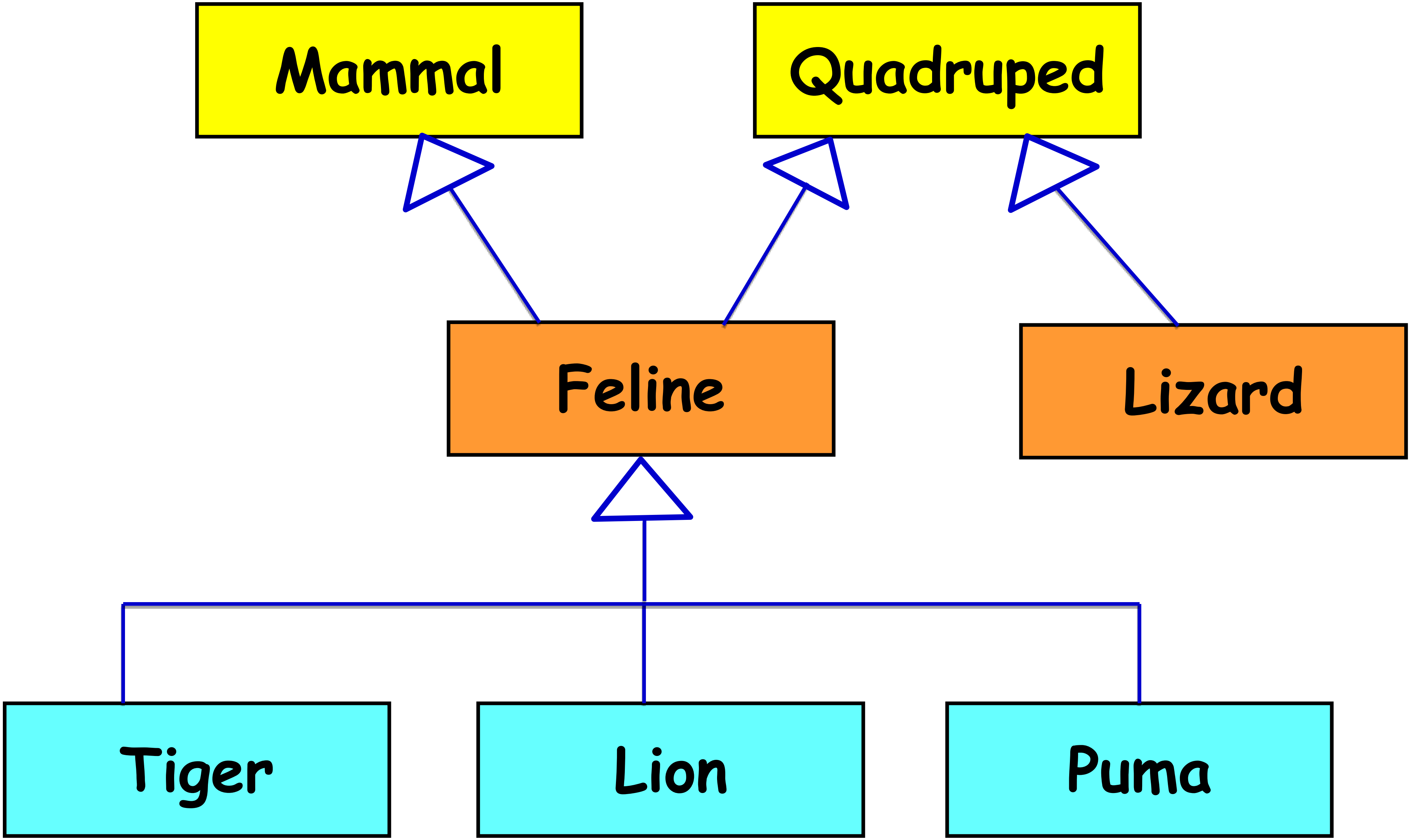
Objects myRectangle and myBox



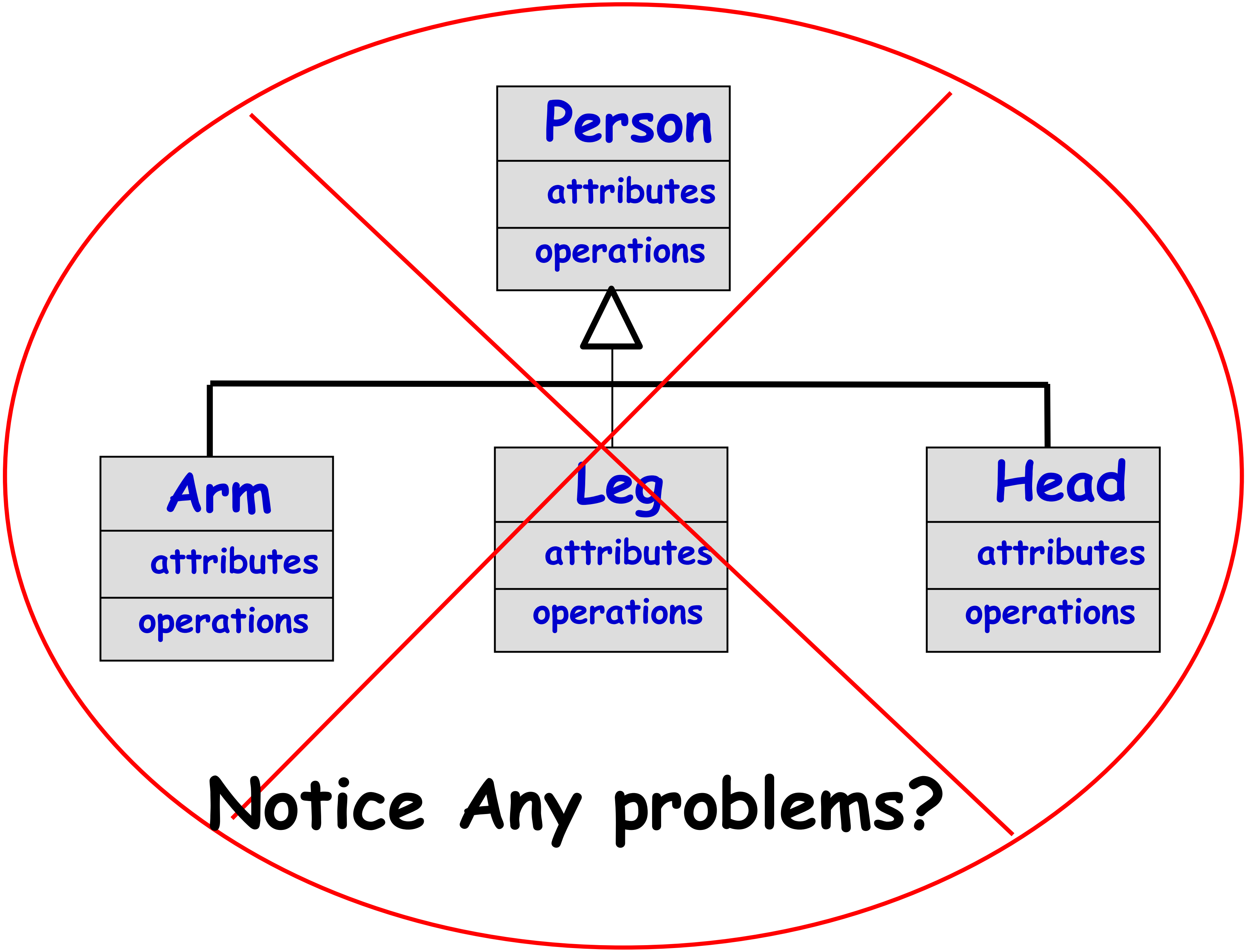
```
Rectangle myRectangle = new Rectangle(5, 3);
Box myBox = new Box(6, 5, 4);
```



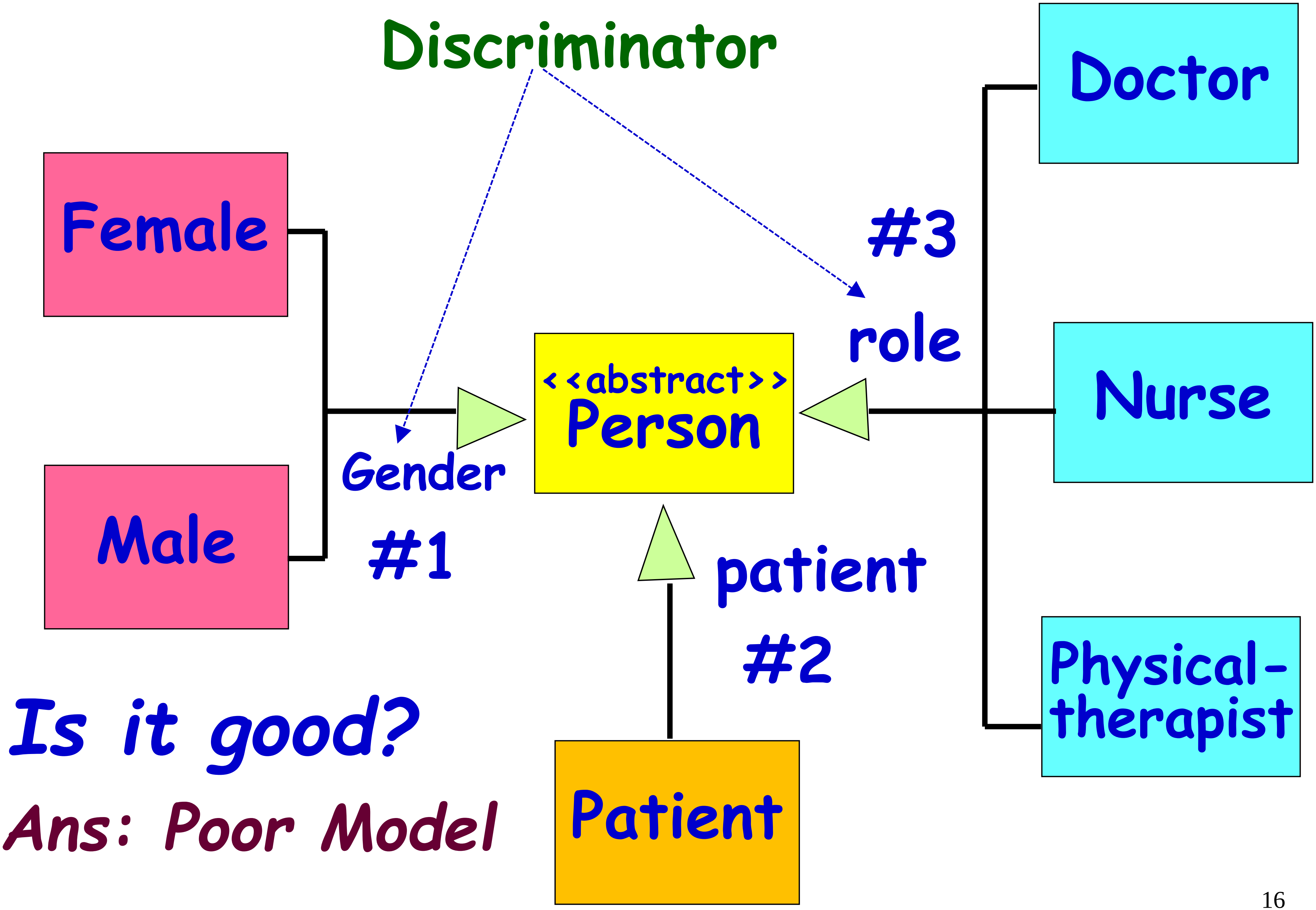
More Generalization Examples…



Poor Generalization
(violates "is a" or "is a kind of" heuristic)



Notice Any problems?



Discriminator

Female

Male

Gender

#1

<<abstract>>
Person

patient

#2

Patient

#3

role

Doctor

Nurse

Physical-
therapist

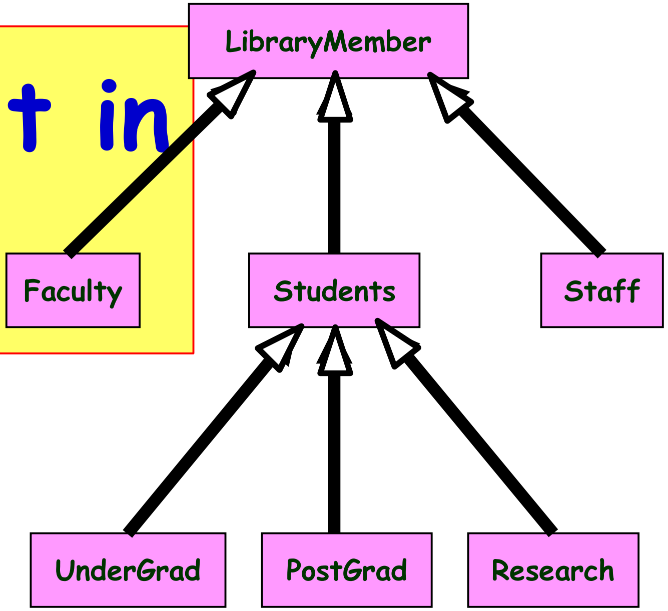
Is it good?

Ans: Poor Model

Inheritance Pitfalls

- Inheritance certainly promotes reuse…

- **Indiscriminate use can result in poor quality programs.**



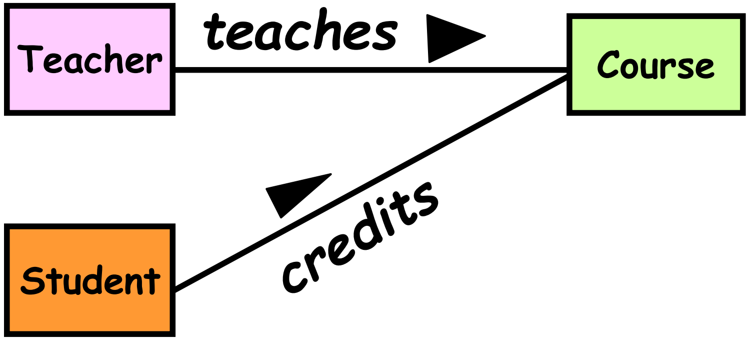
- Base class attributes and methods visible in derived class…
  - **Leads to tight coupling and break of encapsulation**

Association Relationship

- Enables two objects to communicate with each other…

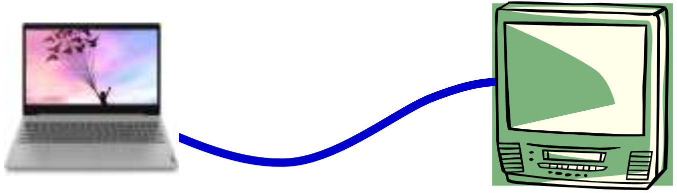
- **How implemented in a program?**
  - One object must "**know**" the object reference (ID) of the corresponding object in an association.

- Usually binary:
  - But in general can be n-ary.

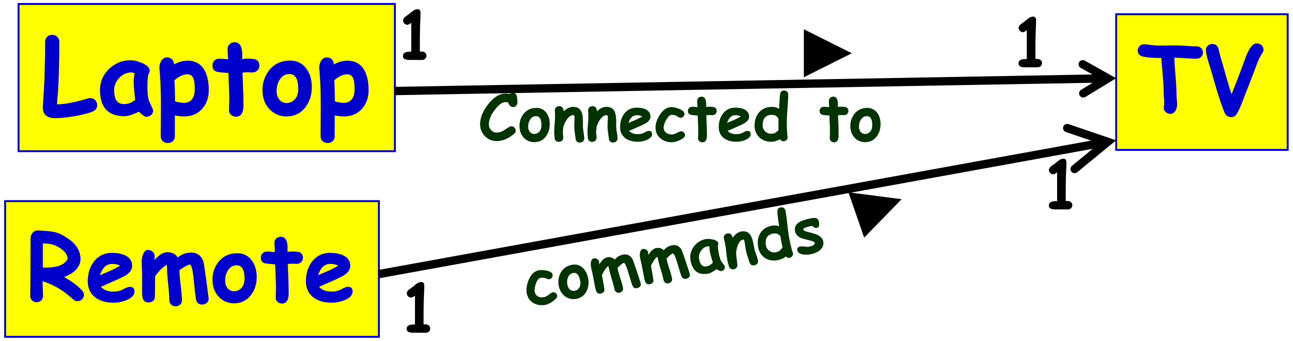


Association – Example

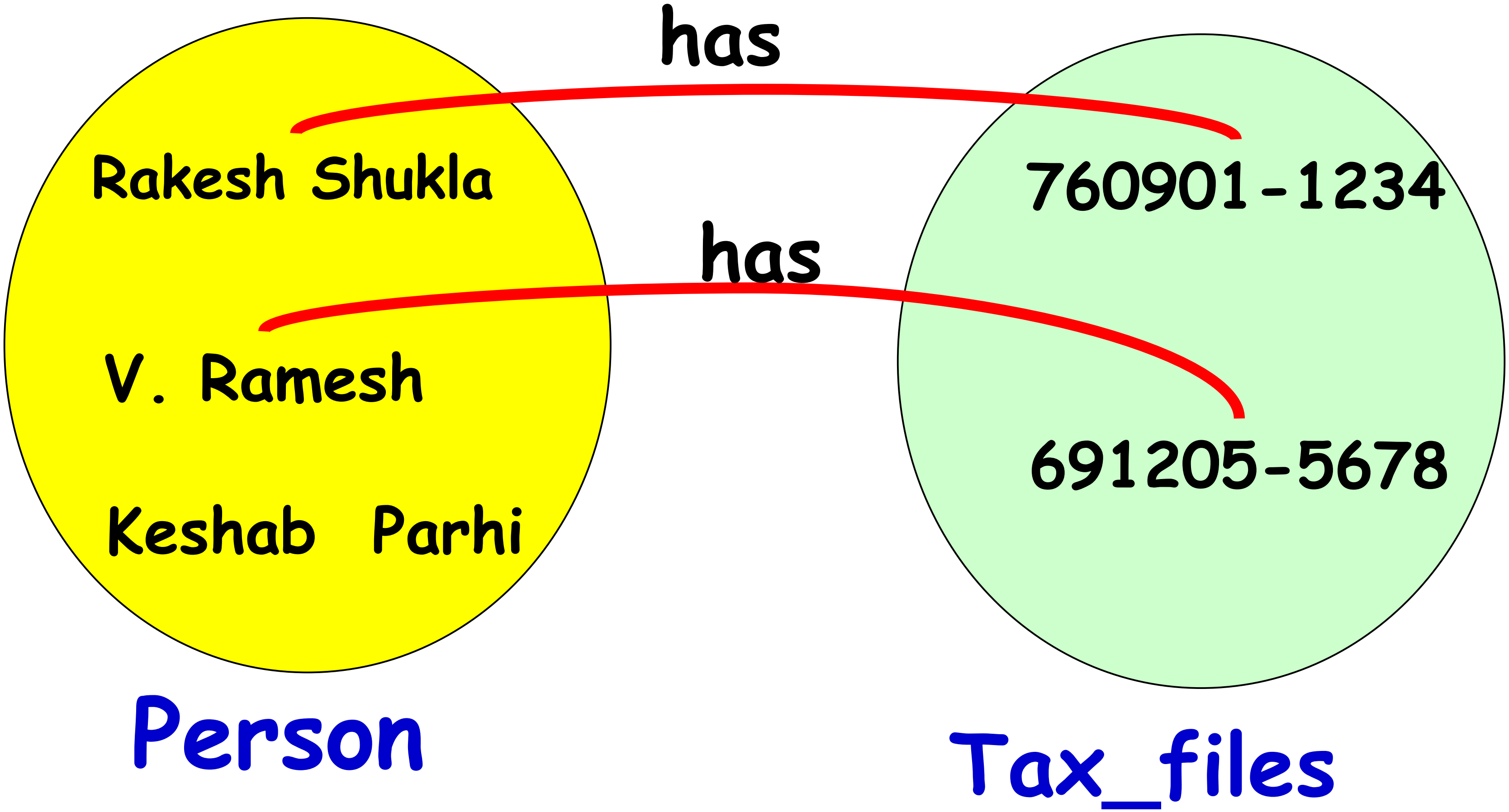
- In a home TV system,
  - A TV object has an association with a Laptop object



    - It may receive signals from the Laptop
  - TV may be associated with remote
    - TV may receive a signal (command)



1-1 Association – Example



has

Rakesh Shukla

760901-1234

has

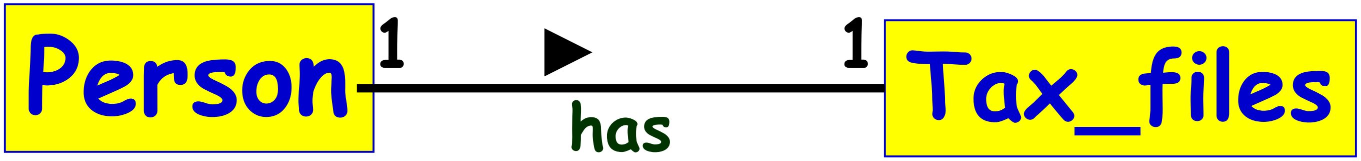
V. Ramesh

691205-5678

Keshab Parhi

Person

Tax_files



Person

1

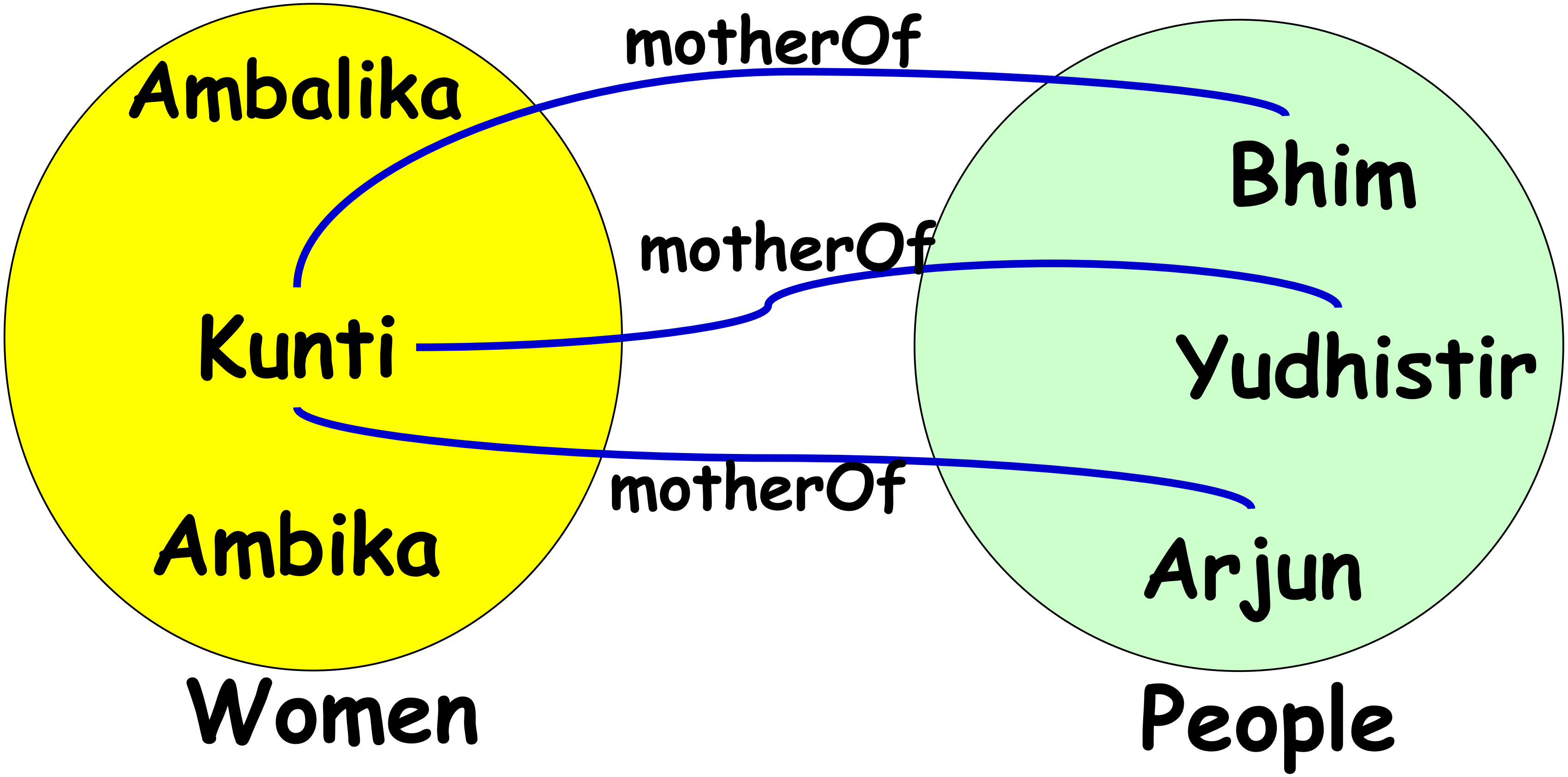
▶

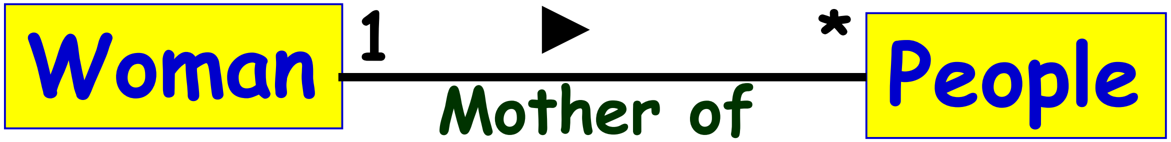
has

1

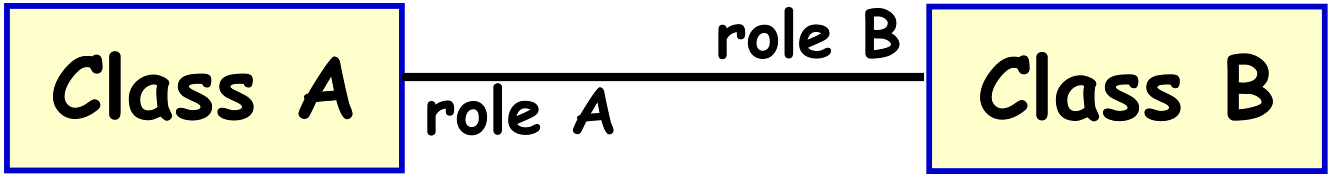
Tax_files

Association-end Multiplicity - Example





Association: UML Syntax

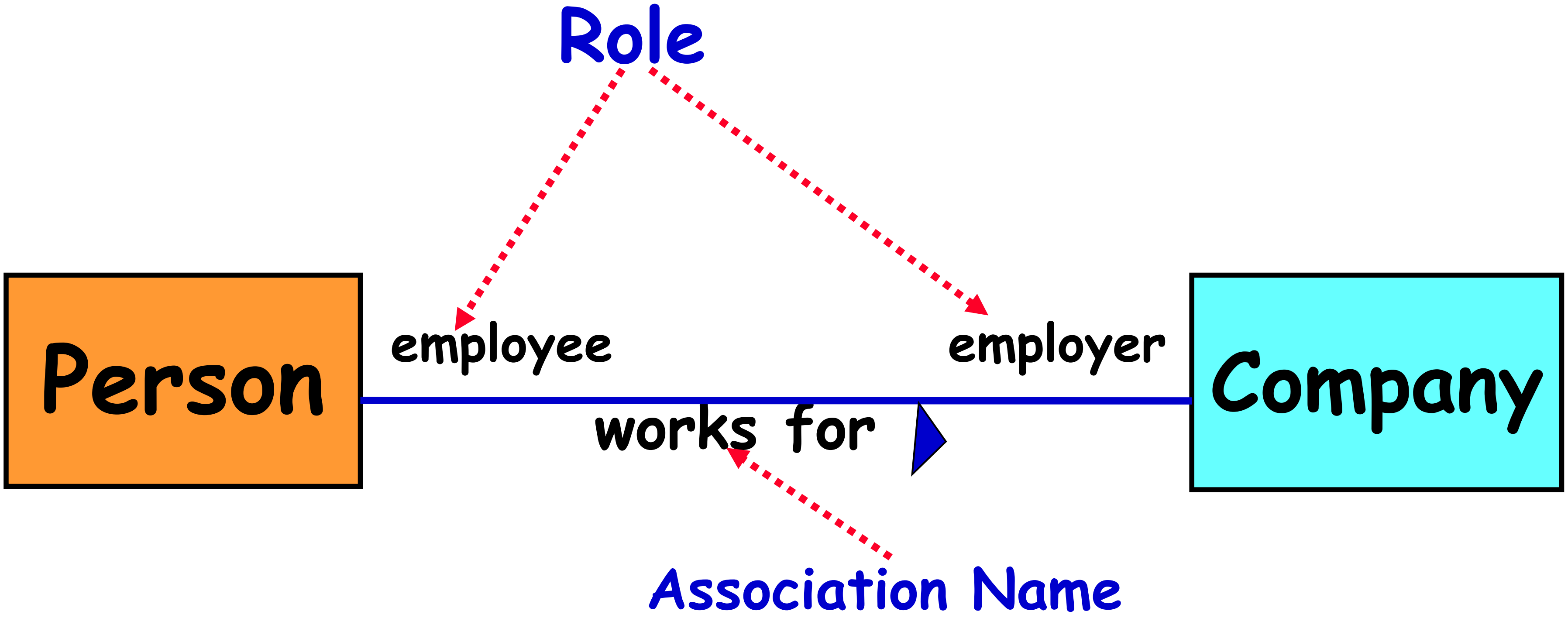

Class A

role B

role A

Class B

- A Person works for a Company.


Role

employee

employer

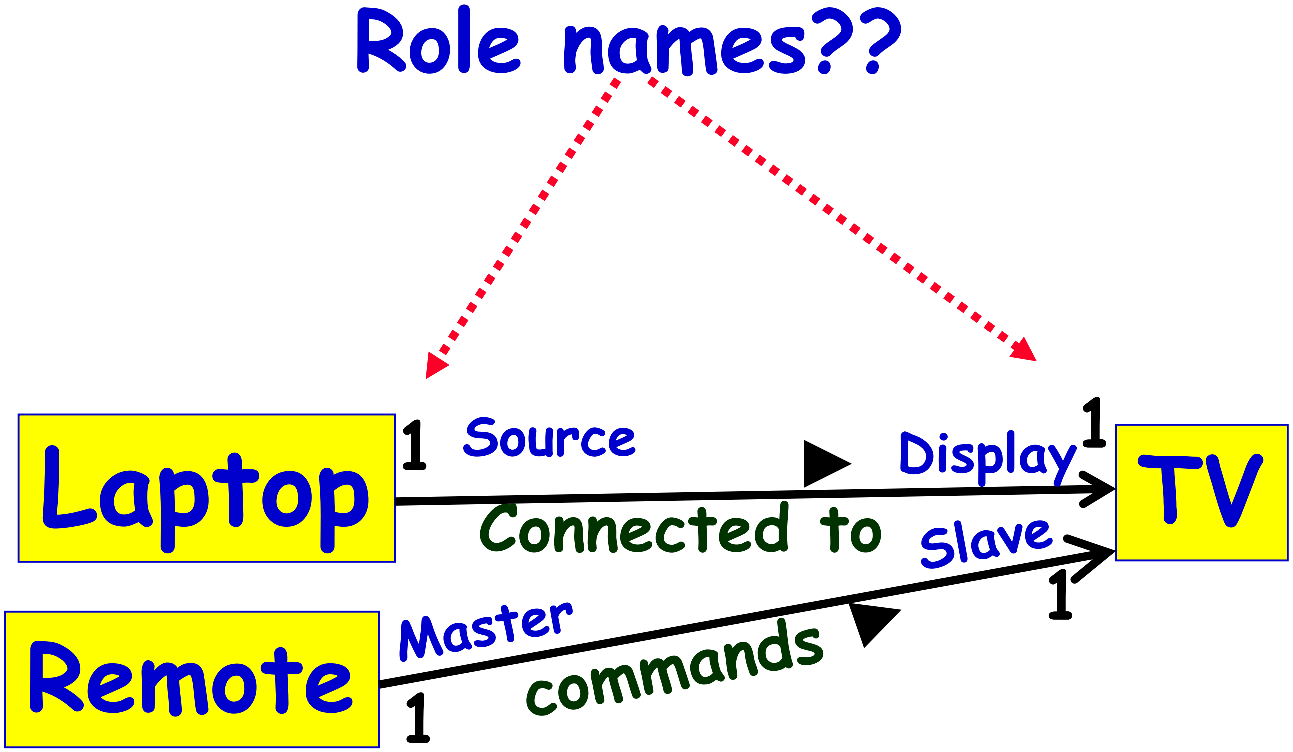
Person

works for

Company

Association Name

Roles in Association


Role names??

Laptop

1

Source

Display

1

TV

Connected to

Slave

1

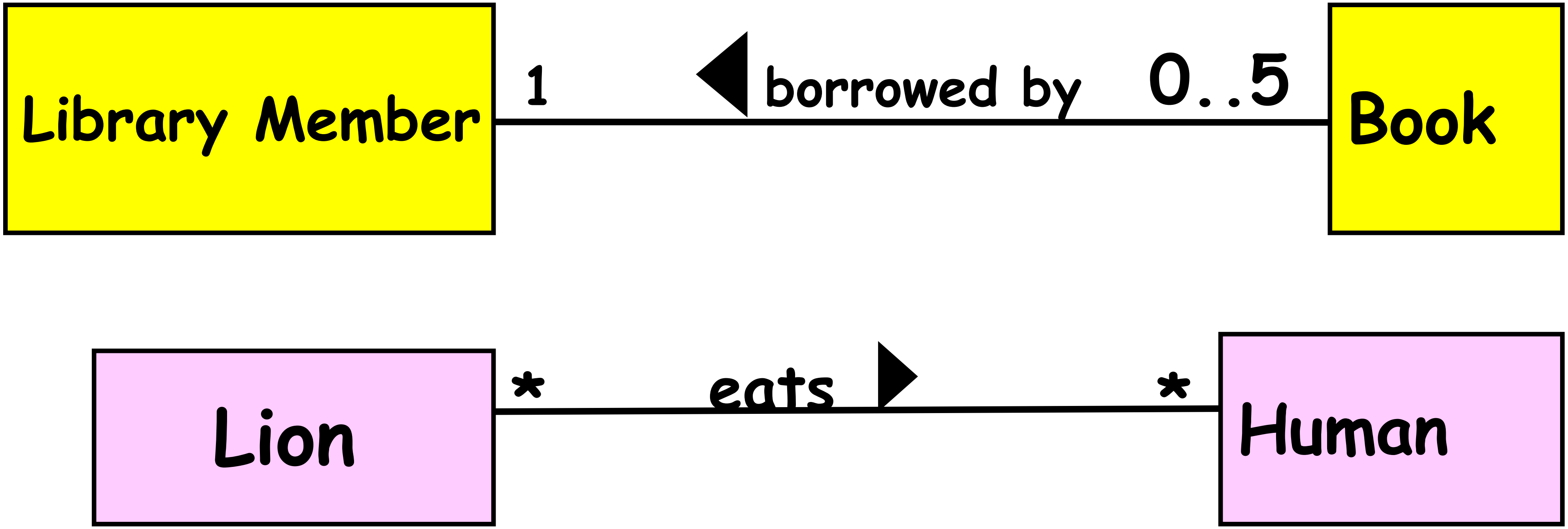
Remote

Master

1

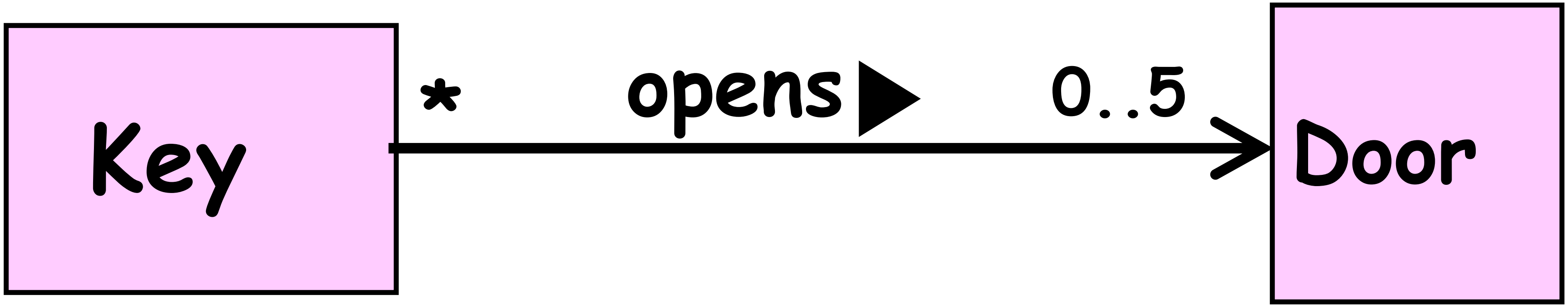
commands

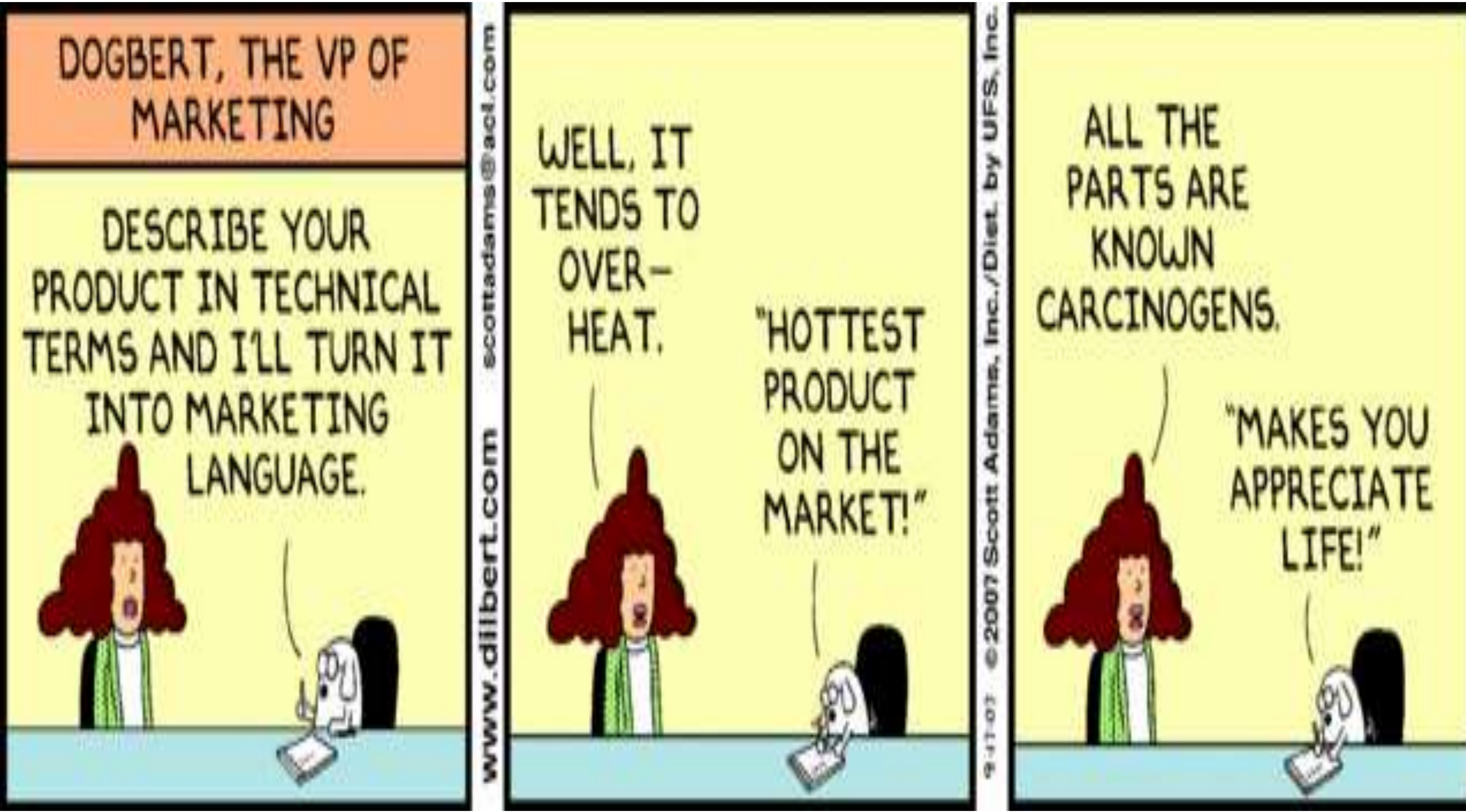
Multiplicity: The number of objects from one class that relate with a single object in an associated class.



Association -- More Examples

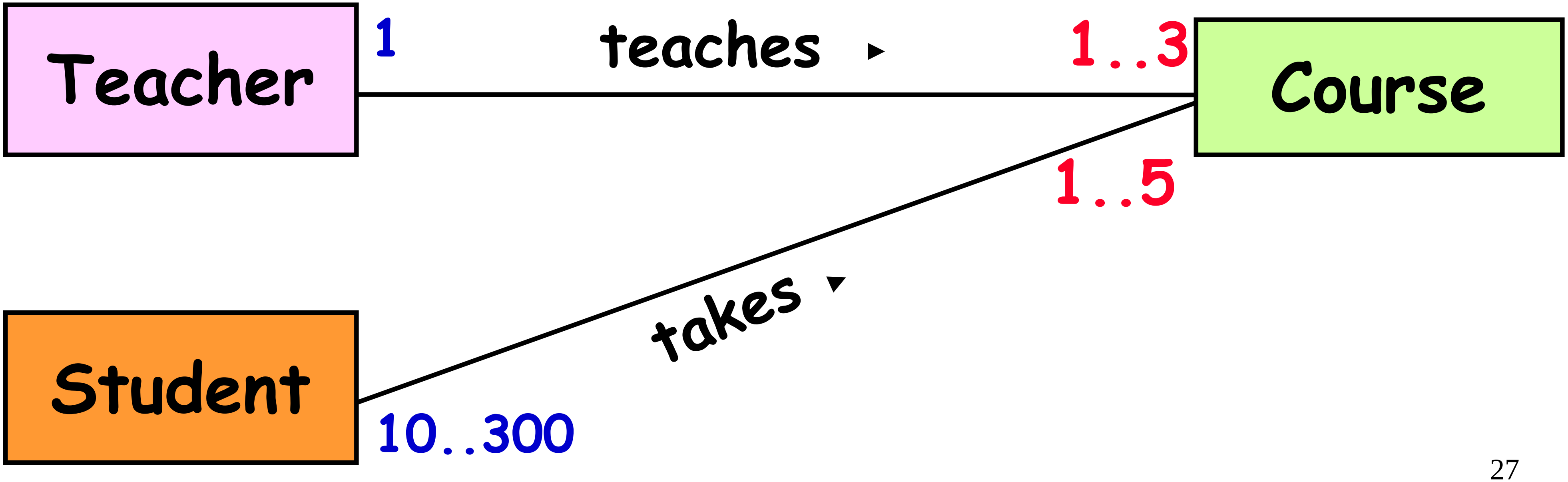
Navigability




DOGBERT, THE VP OF MARKETING
DESCRIBE YOUR PRODUCT IN TECHNICAL TERMS AND I'LL TURN IT INTO MARKETING LANGUAGE.
www.dilbert.com scottadams@aol.com
WELL, IT TENDS TO OVER-HEAT.
"HOTTEST PRODUCT ON THE MARKET!"
9-11-07 © 2007 Scott Adams, Inc./Dist. by UFS, Inc.
ALL THE PARTS ARE KNOWN CARCINOGENS.
"MAKES YOU APPRECIATE LIFE!"

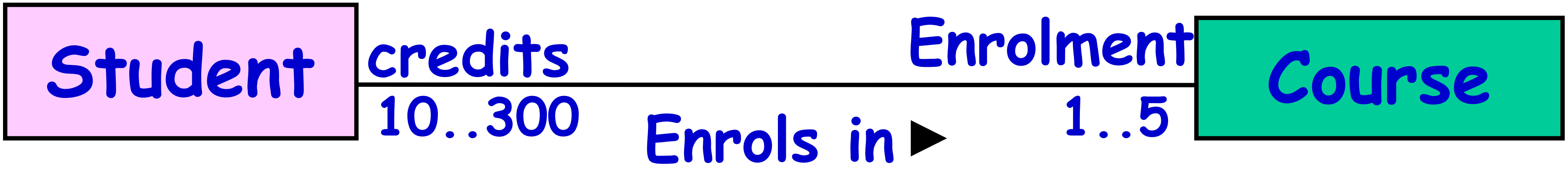
Association – Multiplicity: Exercise

- A teacher teaches 1 to 3 courses (subjects)
- Each course is taught by only one teacher.
- A student can take between 1 to 5 courses.
- A course can have 10 to 300 students.

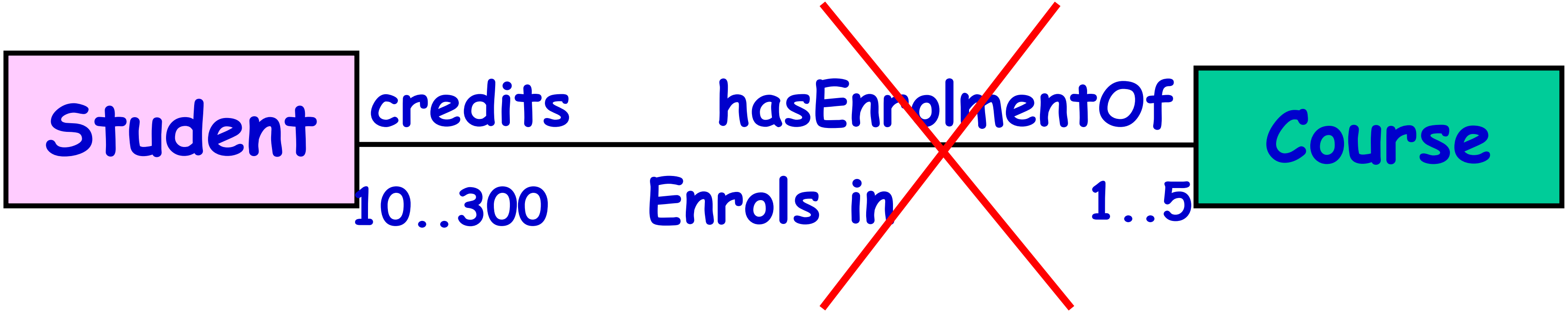


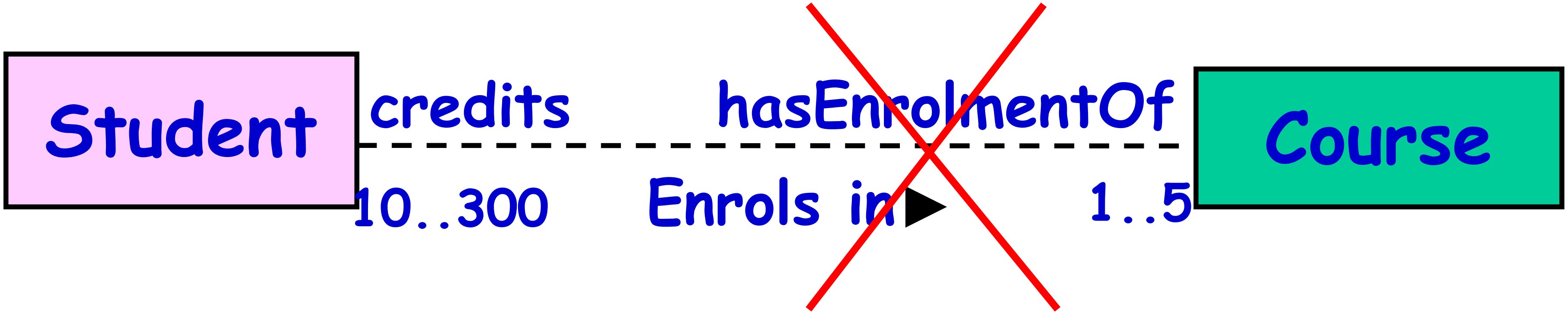
Quiz: Draw Class Diagram

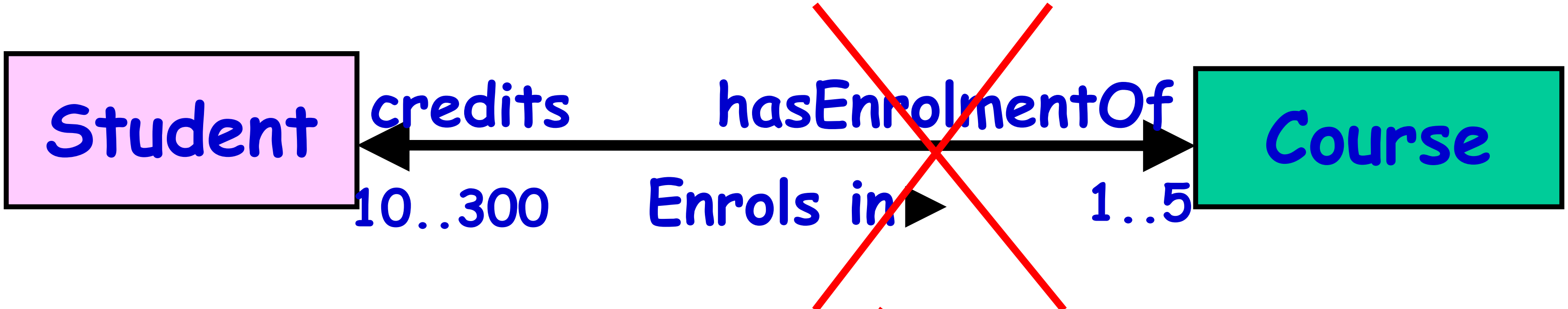
- A Student can take up to five Courses.
- A student needs to enroll in at least one course.
- Up to 300 students can enroll in a course.
- An offered subject in a semester should have at least 10 registered students.

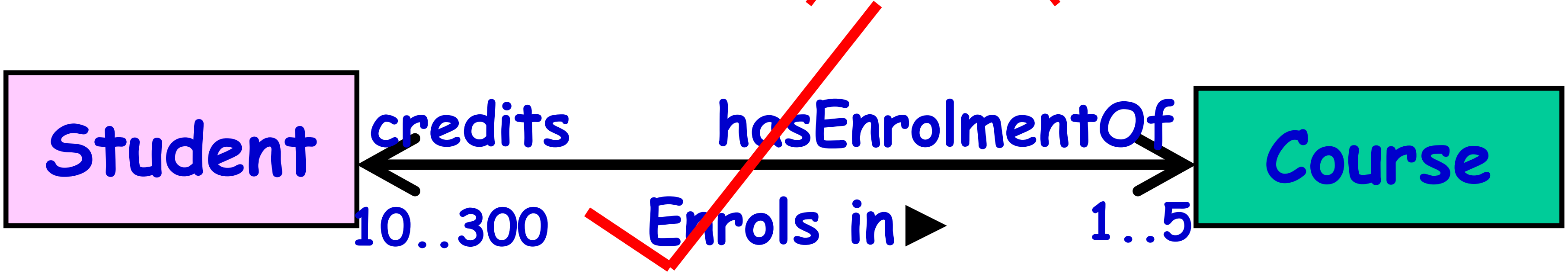

Student
credits
10..300
Enrols in ▶
Enrolment
1..5
Course

Identify Whether Correct or Wrong

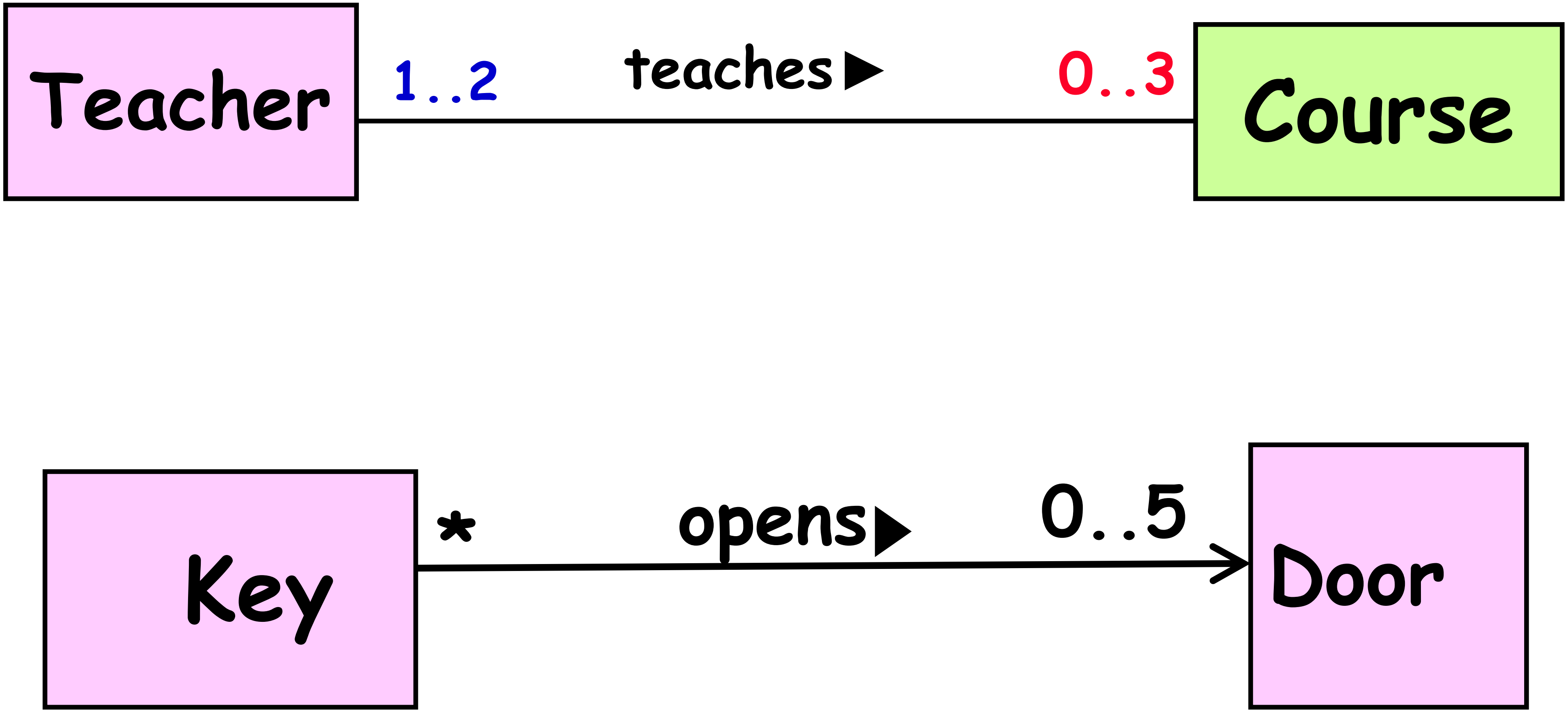






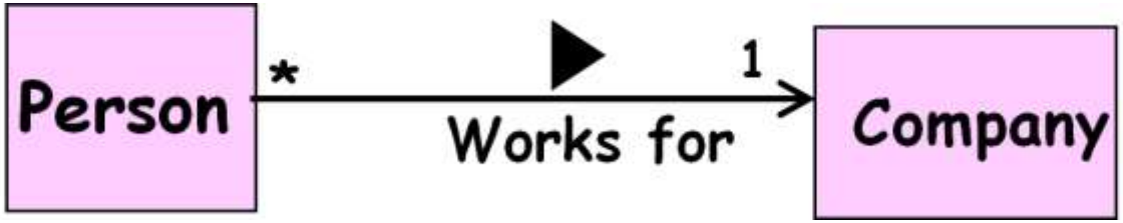


Quiz: Read the Diagram



Association and Link

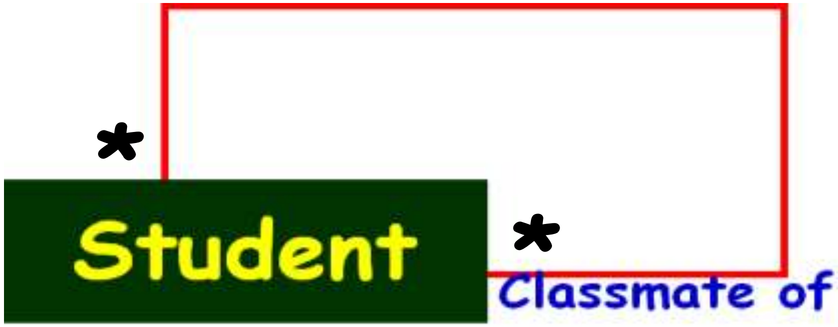
- **A link:**
  - An instance of an association
  - Exists between two or more objects
  - **Dynamically created and destroyed as a run of a system proceeds**
- For example:



  - An employee joins an organization.
  - Leaves that organization and joins a new organization etc.

Association Relationship

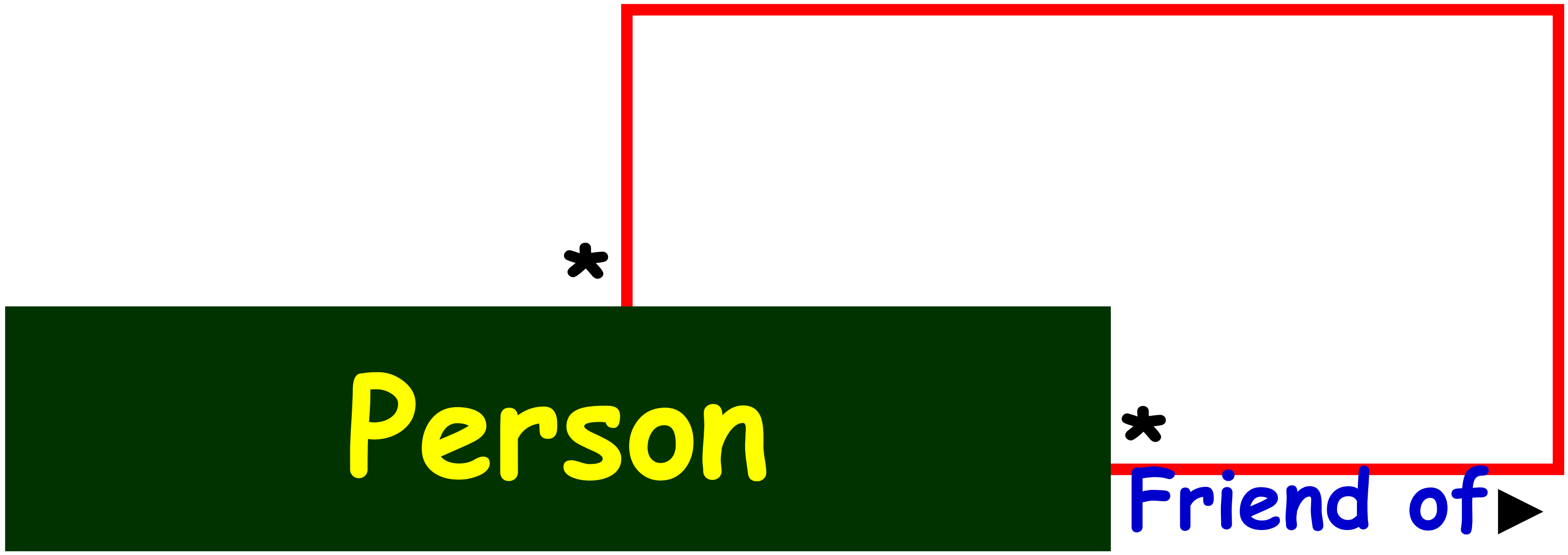
- A class can be associated with itself (**unary** association).
  - **Give an example?**



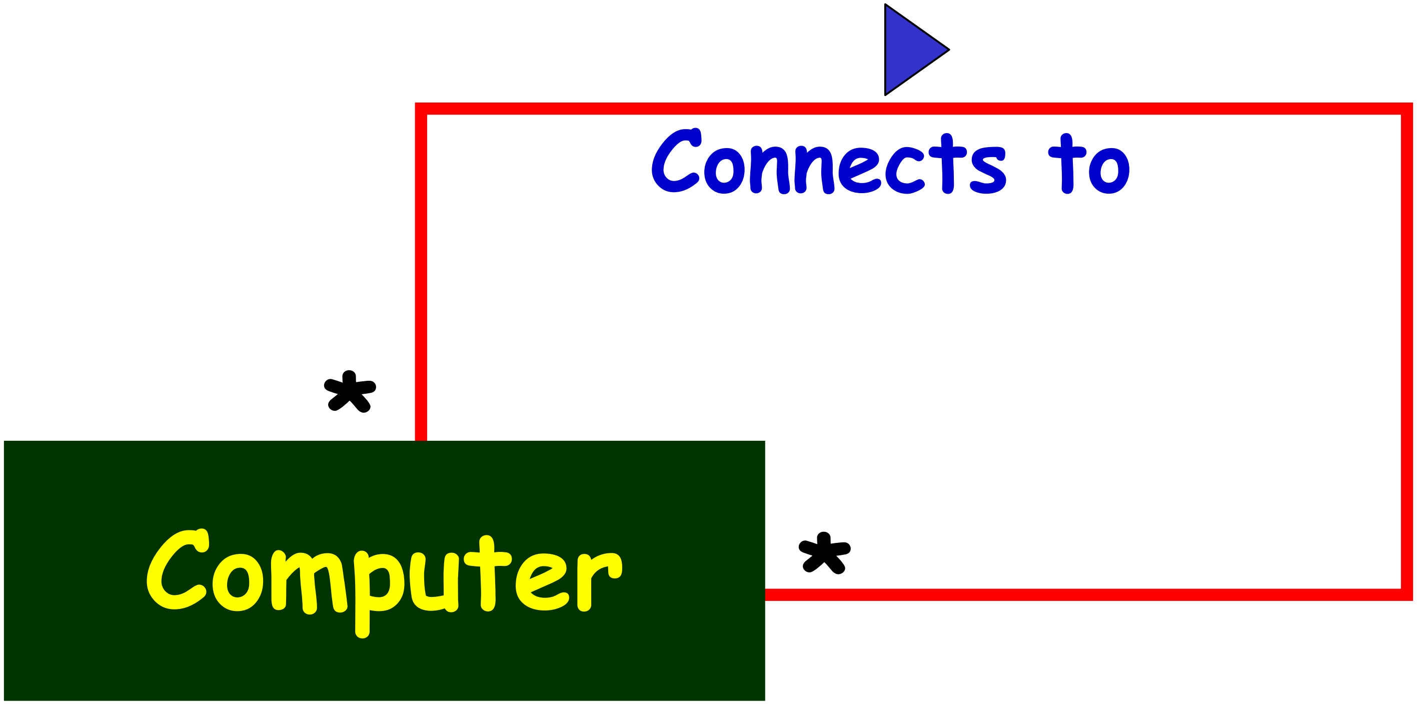
- An arrowhead used along with name:
  - Indicates direction of association.

- Multiplicity indicates # of instances taking part in the association.

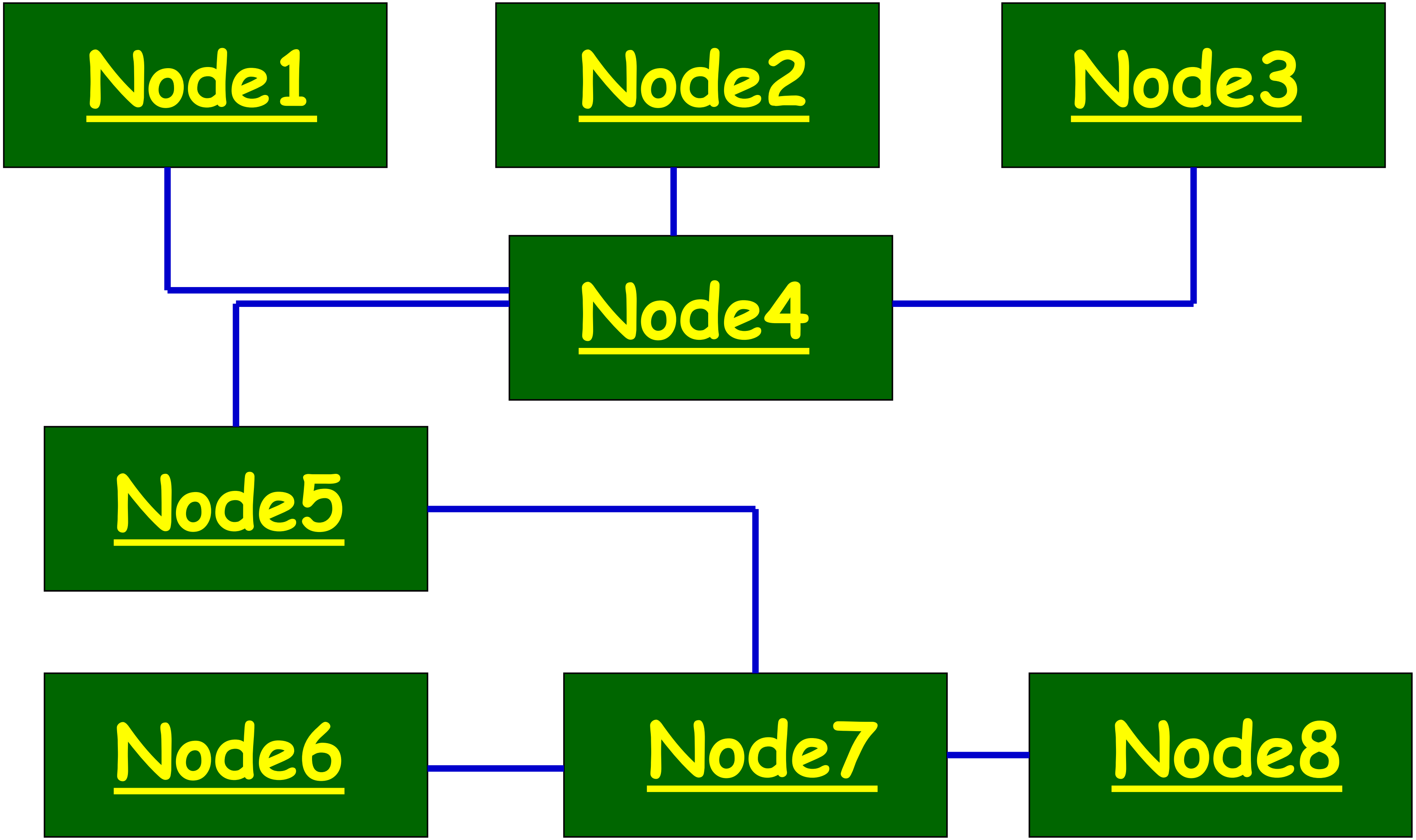
Unary (Self) Association: Example 0



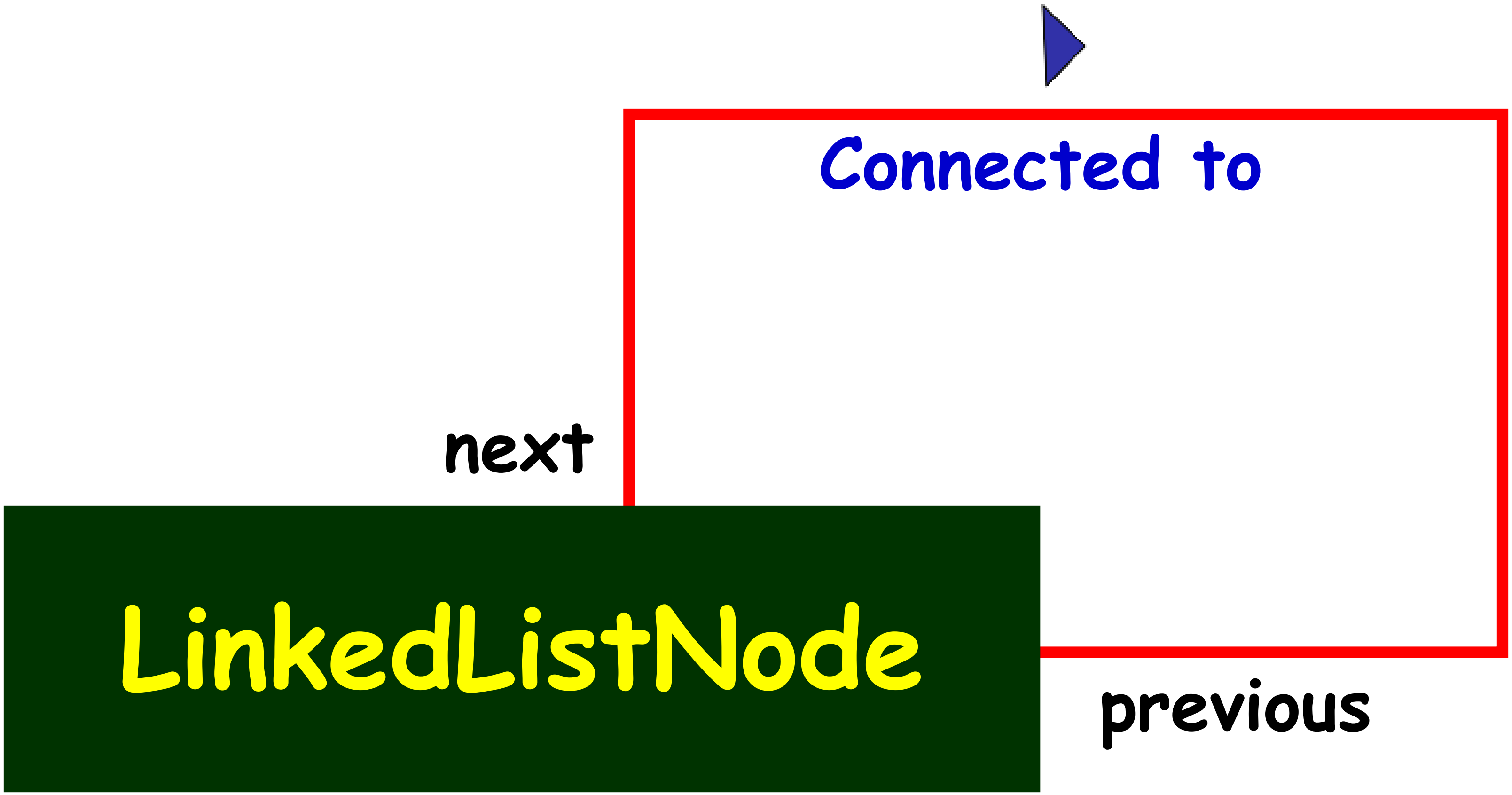
Unary Association: Example 1
Computer Network



Computer Network: Object Diagram

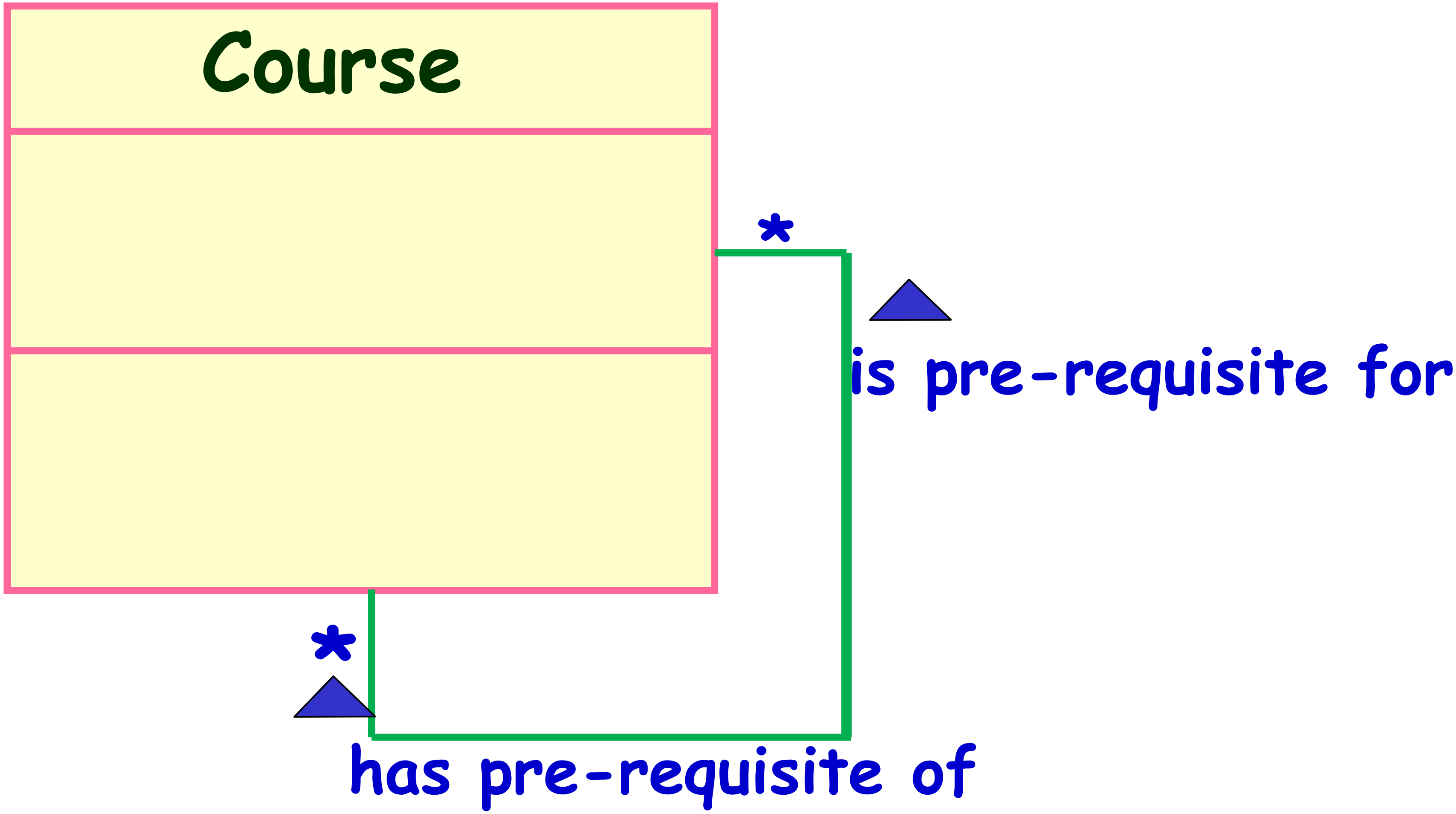


Unary Association: Example 2



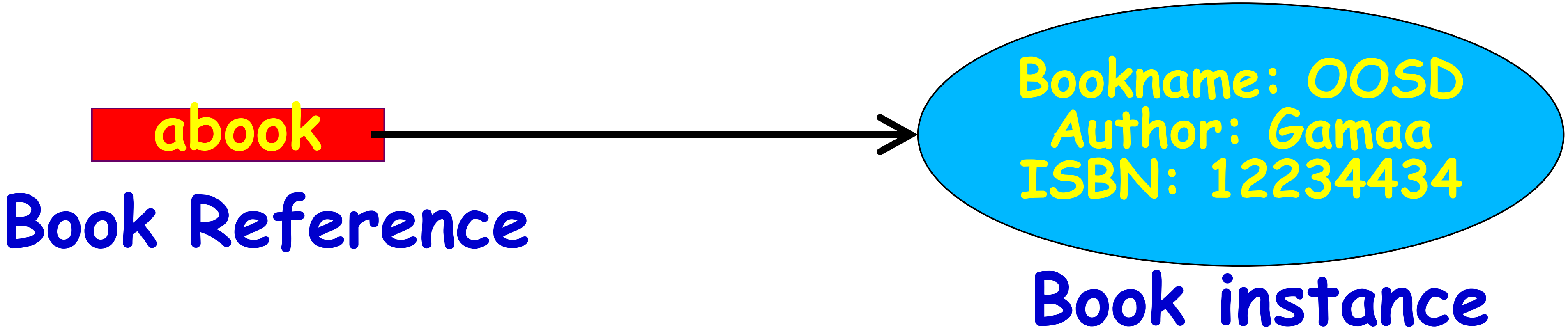
WE DON'T HAVE ANY OPENINGS FOR REGULAR INTERNS, BUT I CAN OFFER YOU A JOB AS AN INTERN TO OUR INTERN.
Dilbert.com DilbertCartoonist@gmail.com
WE WON'T PAY YOU, OF COURSE, BUT YOU MIGHT ACQUIRE AN IMPERCEPTIBLE AMOUNT OF SEMI-RELEVANT JOB EXPERIENCE.
9-7-12 ©2012 Scott Adams, Inc./Dist. by Universal Uclick
AND SOME-TIMES WE'LL SLAP YOU FOR NO REASON.
STUPID ECONOMY!
I'LL TAKE IT.

Unary (Reflexive) Association: Example 3



Implementing Association Relationship: Example 1

- Store a **reference variable** of the associated class.

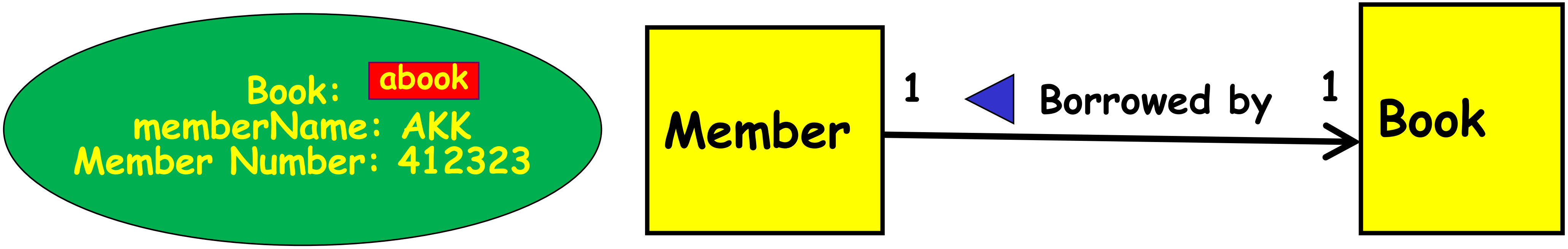


abook

Book Reference

Bookname: OOSD
Author: Gamaa
ISBN: 12234434

Book instance



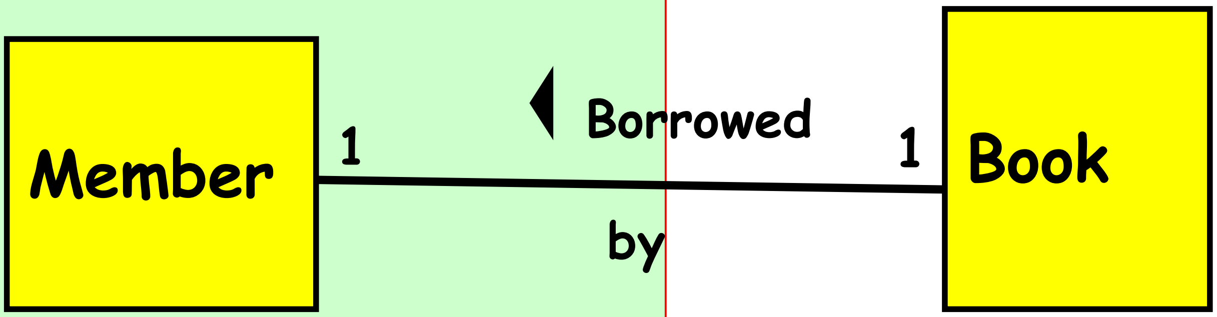
Book: abook
memberName: AKK
Member Number: 412323

Member

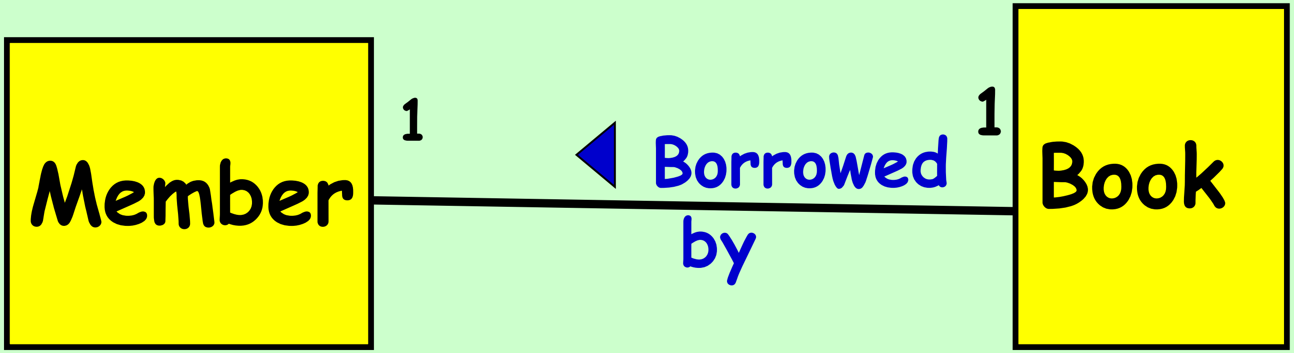
1 Borrowed by 1

Book

```
public class Member{
private Book book;
public issueBook(Book abook){
     setBook(abook);
     abook.setLender(this);
     }
private setBook(Book abook){
     book=abook;
   }
   ...
}
```

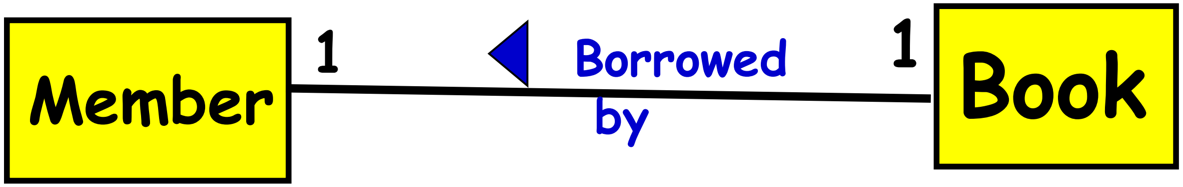


```
public class Book{
private Member member;
public setLender(Member aLender){
        member=aLender;
    }
    ...
}
```

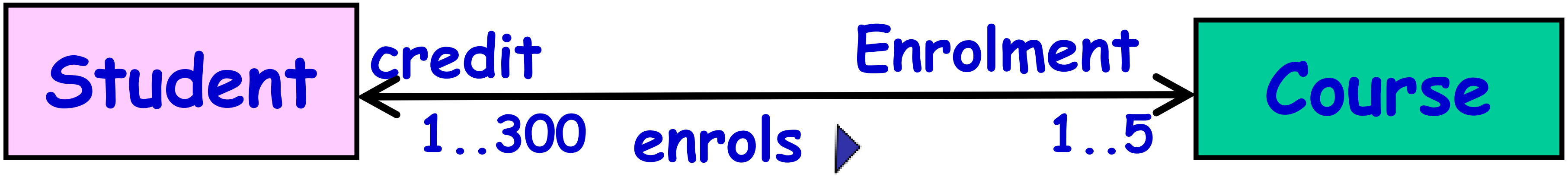


Naming Association Attribute

- The association attribute name is derived from:
  - Role names, if present
  - If not present, name of the class to which it is associated



Association Implementation: Quiz



Student

credit

1..300

enrols ▶

Enrolment

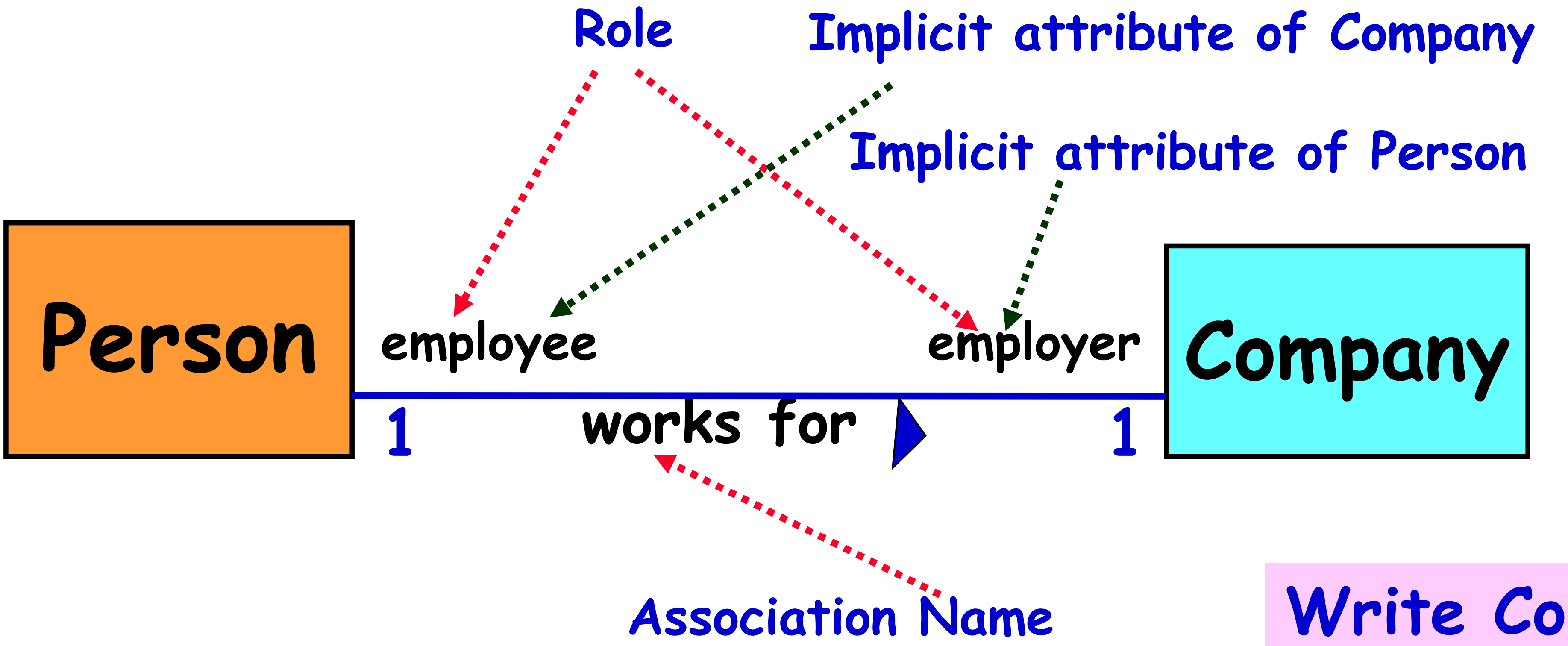
1..5

Course

```
Class Student {
     Course Enrolment[5];
     …
}
Class Course {
   Student credit[300];
     …
}
```

Association Exercise 1

- **Develop class model:** A Person works for a Company. A Company employs one person.



Write Code

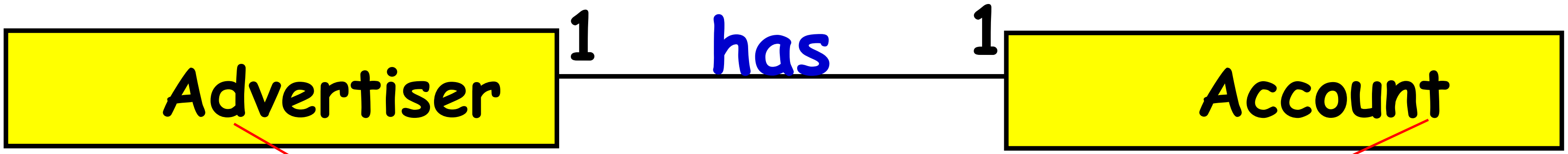
Observe: Bidirectional navigation.

Implement in Java…

Exercise 1: Solution

```
public class Company {
        private Person employee;
        public void employPerson(Person p){
                employee=p;
                 p.setCompany(this);
}
public class Person {
  private Company employer;
  public Company getWorksFor() {
        return employer;
  }
  public void setCompany(Company c) {
        employer=c;
  }
}
```

1-1 Association: Example 3

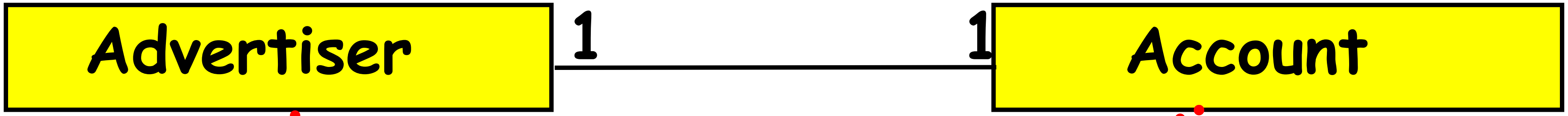


```
public class Advertiser {
        private Account account;
        public Advertiser() {
                account = new Account(this);
        }
        public Account getAccount() {
                return account;
        }
}
```

Write code for Advertiser

Now, Write code for Account

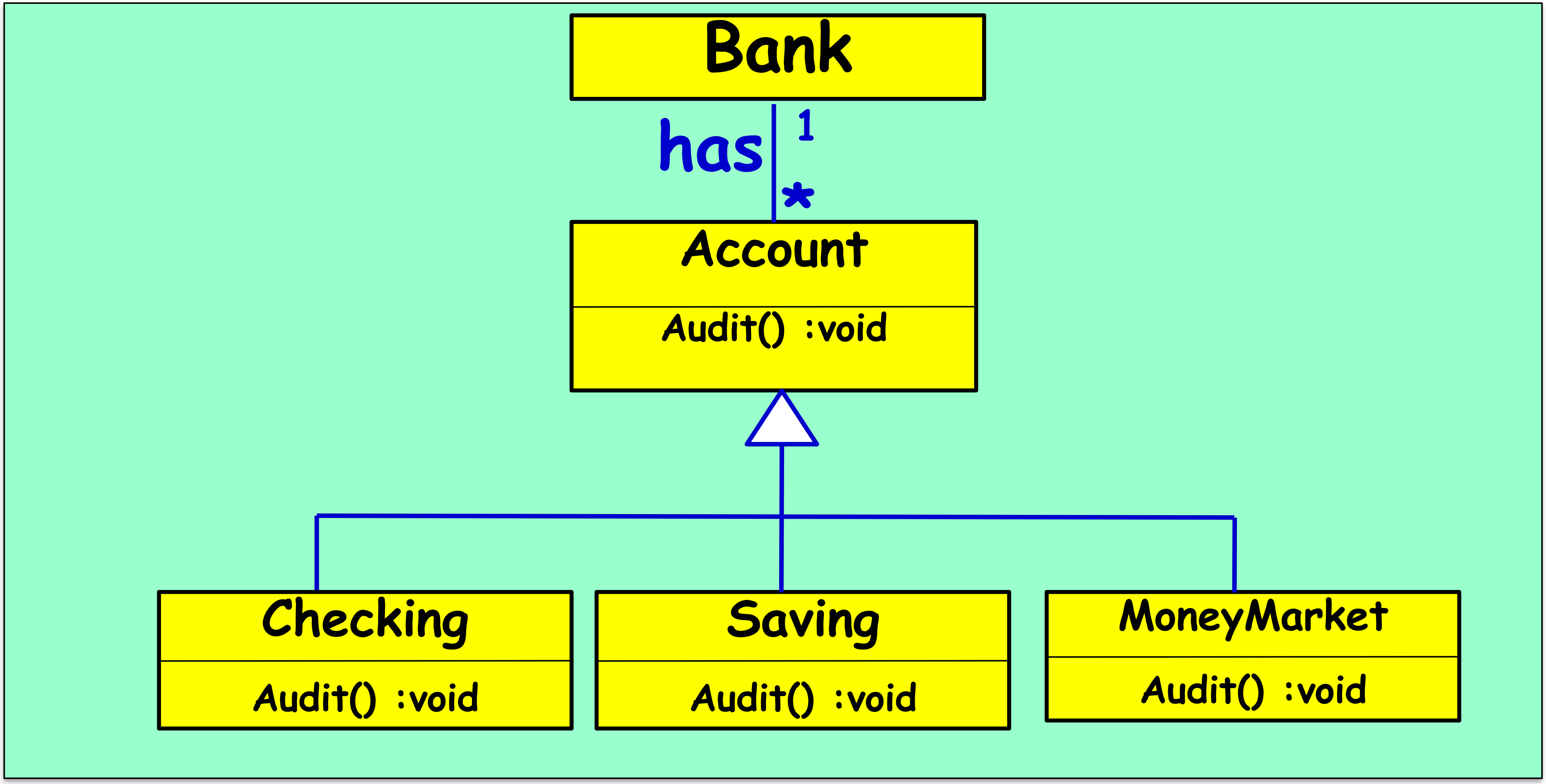
1-1 Association



```
public class Advertiser {
   private Account account;
   public Advertiser() {
   account = new Account(this);
   }
   public Account getAccount()
   {
         return account;
   }
}
```

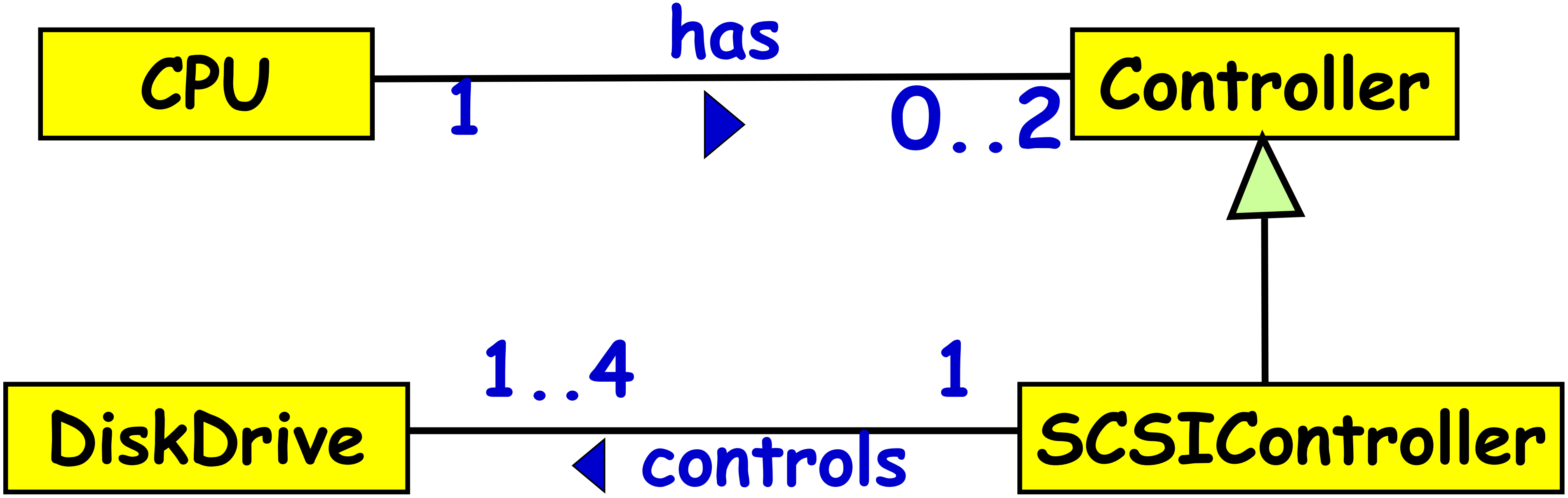
```
public class Account {
         private Advertiser owner;
   publicAccount(Advertiser
   owner) {
         this.owner = owner;
   }
   public Advertiser getOwner() {
         return owner;
   }
}
```

Quiz: Read Banking System Class Model



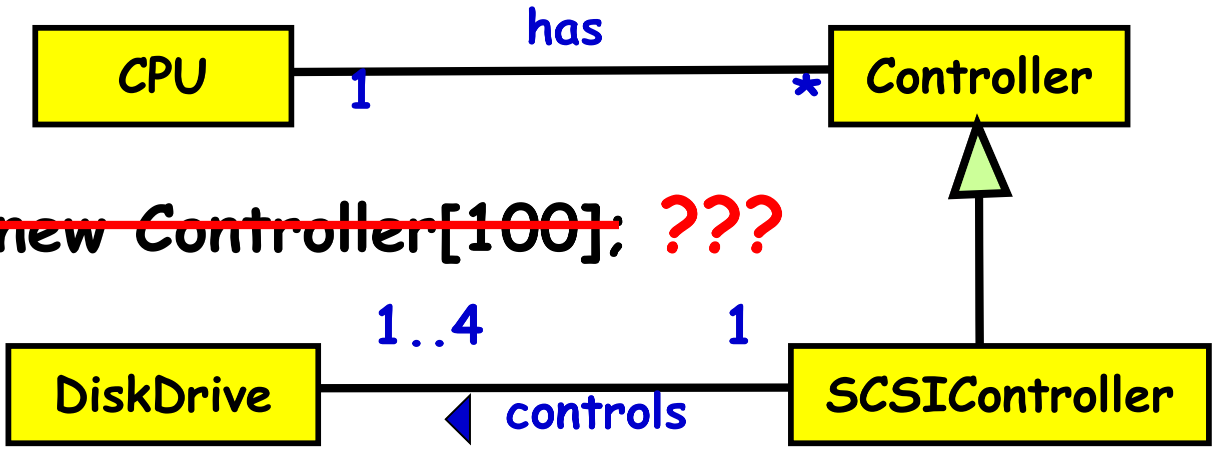
- A Bank has many Accounts
- Checking, Savings, MoneyMarket are Account types

Quiz: Read UML class diagram



- A CPU has upto two Controllers
- SCSIController is a special type of Controller
- A SCSIController controls 1-4 DiskDrives

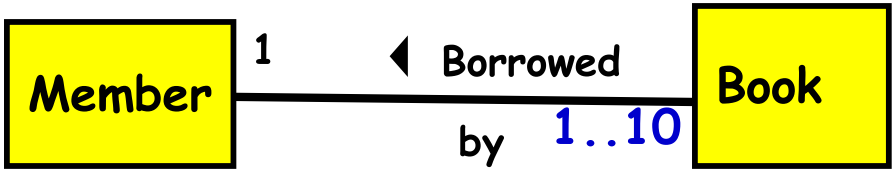
Java Code?



```
class CPU {
   ~~Controller [ ]  myCtlrs=new Controller[100];~~ ???
}
class Controller {
   CPU myCPU;
}
class SCSIController extends Controller {
   DiskDrive [ ] myDrives= new DiskDrive[4];
}
Class DiskDrive {
    SCSIController mySCSI;
}
```

Implementing Association Multiplicities

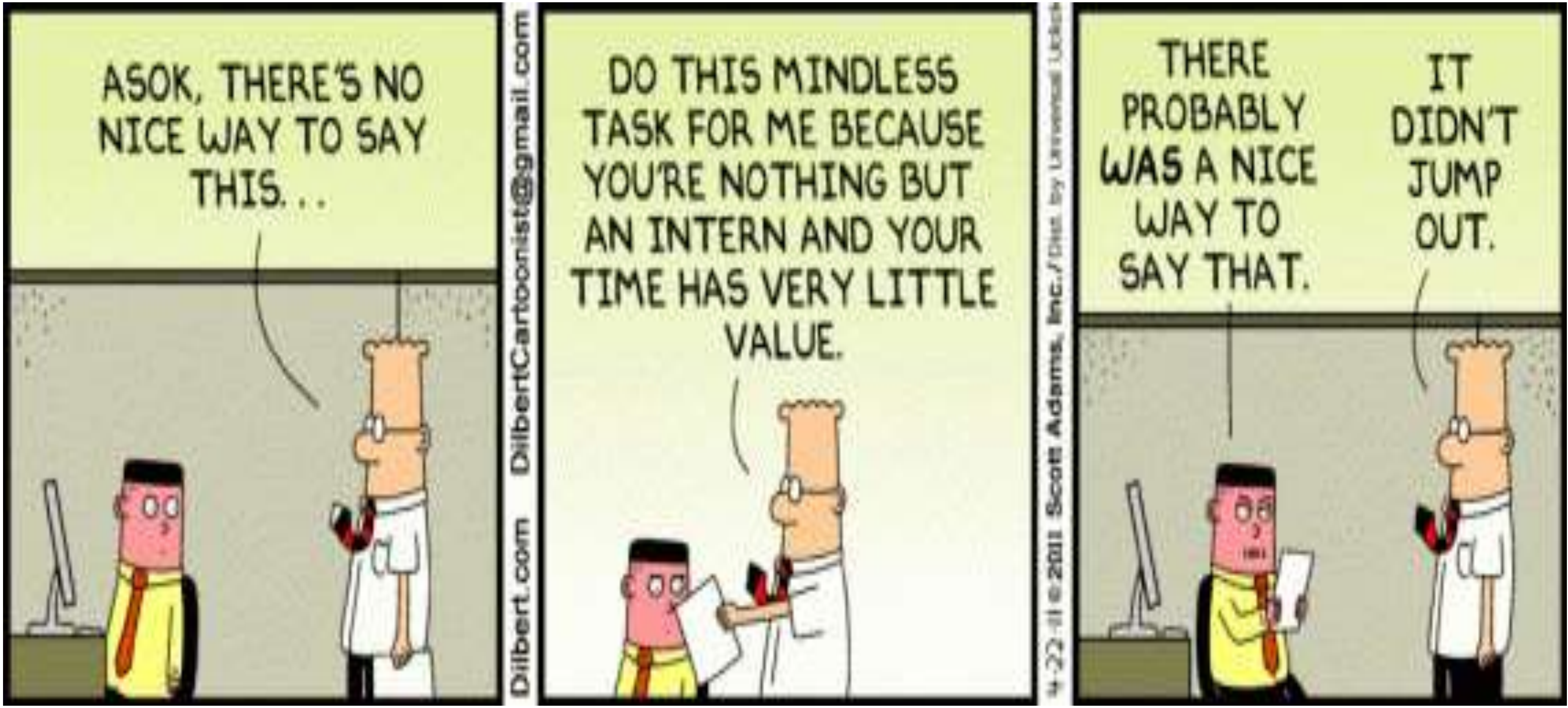
- Fixed multiplicity:



- The reference field can be

  Book [] book=new Book[10];.

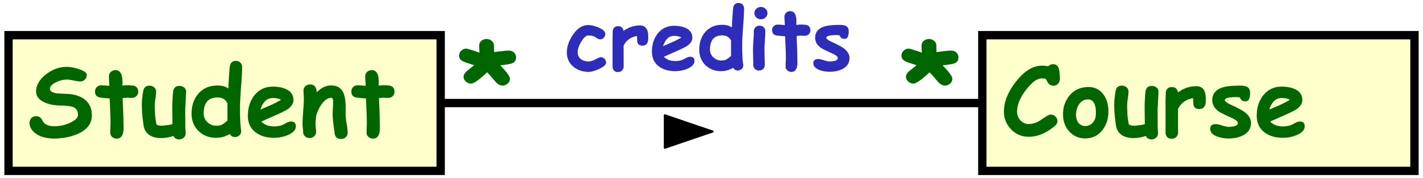
- When the upper end is *:
  - Java collections need to be used
  - Collections facility was added to Java as a part of JDK 1.2


ASOK, THERE'S NO NICE WAY TO SAY THIS. . .
Dilbert.com DilbertCartoonist@gmail.com
DO THIS MINDLESS TASK FOR ME BECAUSE YOU'RE NOTHING BUT AN INTERN AND YOUR TIME HAS VERY LITTLE VALUE.
4-22-11 © 2011 Scott Adams, Inc./Dist. by Universal Uclick
THERE PROBABLY WAS A NICE WAY TO SAY THAT.
IT DIDN'T JUMP OUT.

Association Class

Need for Association Class

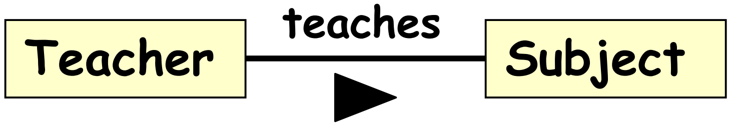
- In many cases information associated with the links between objects needs to be maintained.



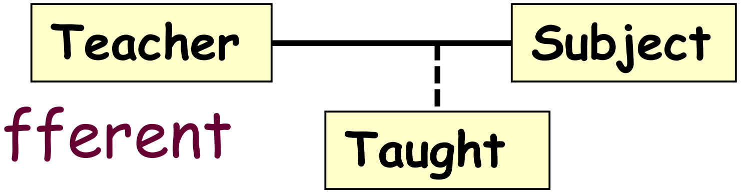
- **Example:**
    - A student takes a course and gets a grade for it.
    - The grade only makes sense if we know the student and the course. Also the semester to be remembered.
    - The obtained grade is not an attribute of either class.

Association Class

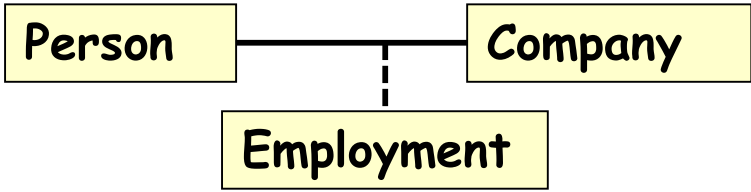
- Not unusual for an association relation has its own attributes and methods:



  - **Example 1: A teacher teaches a subject:**
    - A teacher may teach the same subject multiple times over many semesters



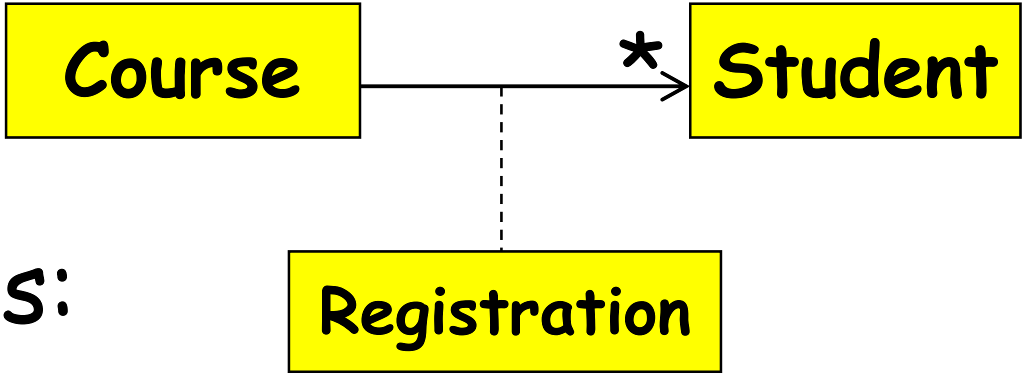
    - Each time the students may be different
  - **Example 2: A person works for a company**
    - The date he joined a company



    - Pay scale and grade at which joined, and so on
    - A person may join a company multiple times

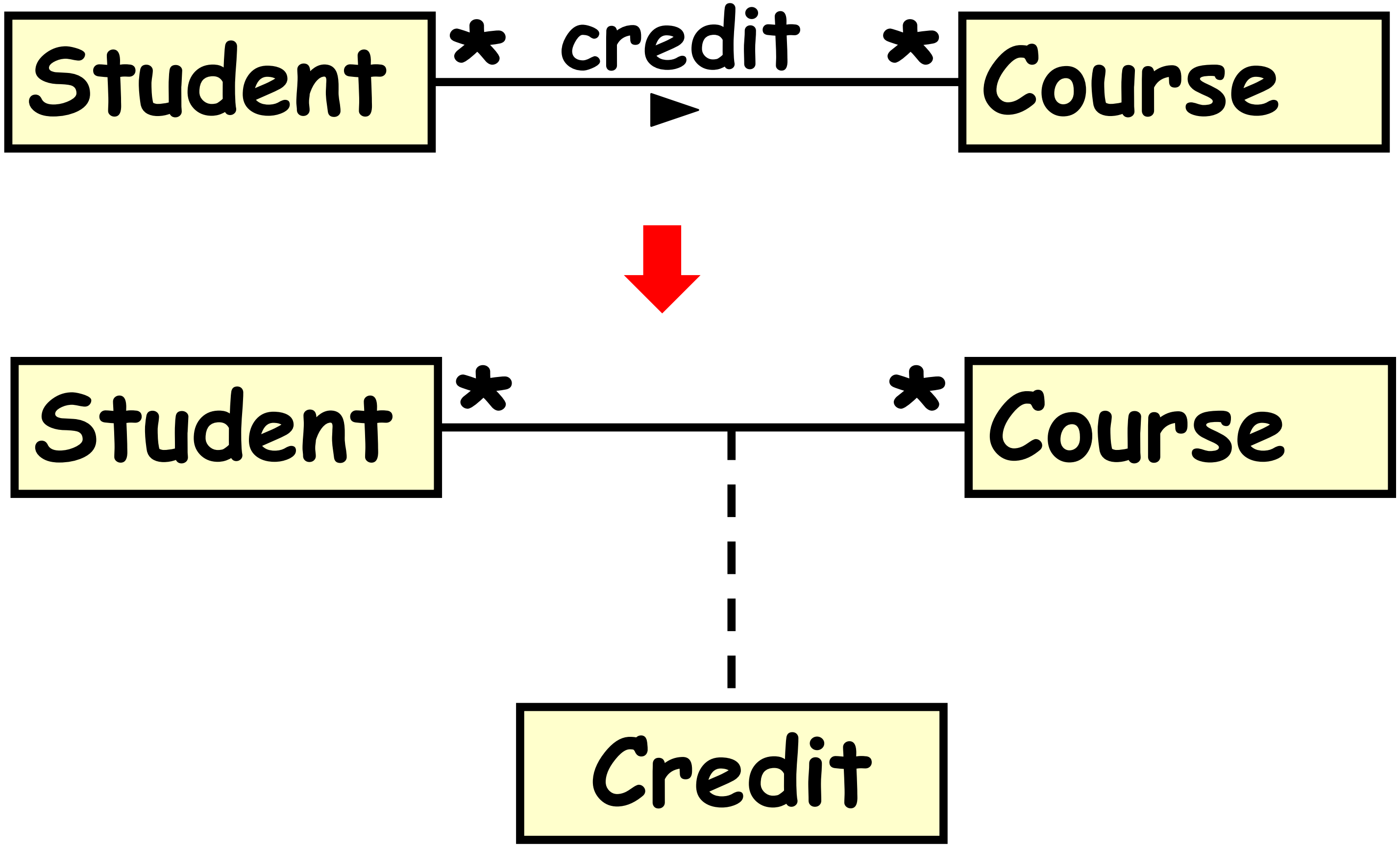
Association Class

- An association class is shown as a class symbol:
  - Attached to the association symbol by a dashed line.

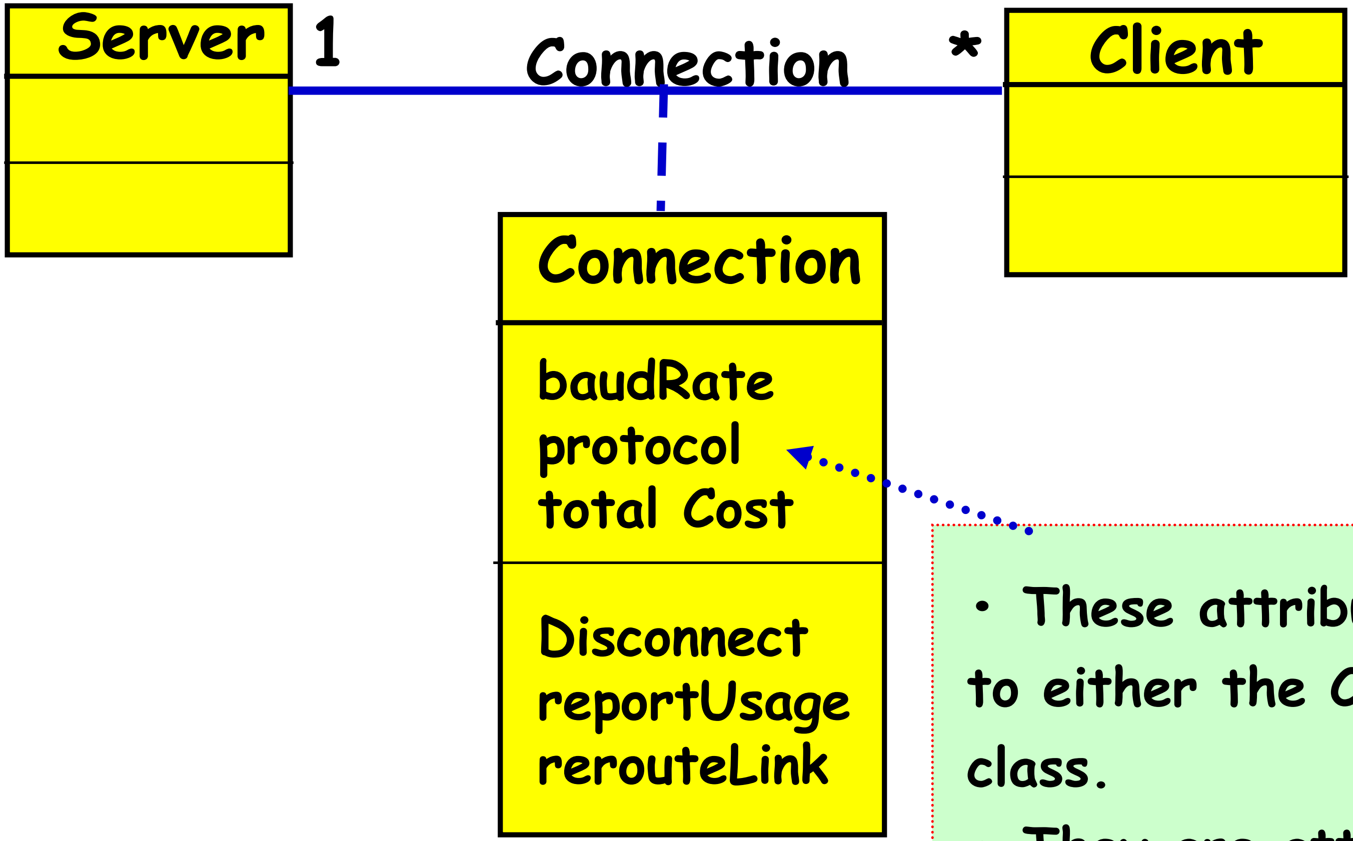


- Association class has:
  - Same name as the association relation because...
  - It represents the association!

Association Class



Association Class: Example 1



- These attributes don't belong to either the Client or Server class.
- They are attributes of the connection itself.

- An association class can have methods as well as attributes.