UML and Classes, Objects and Relationships

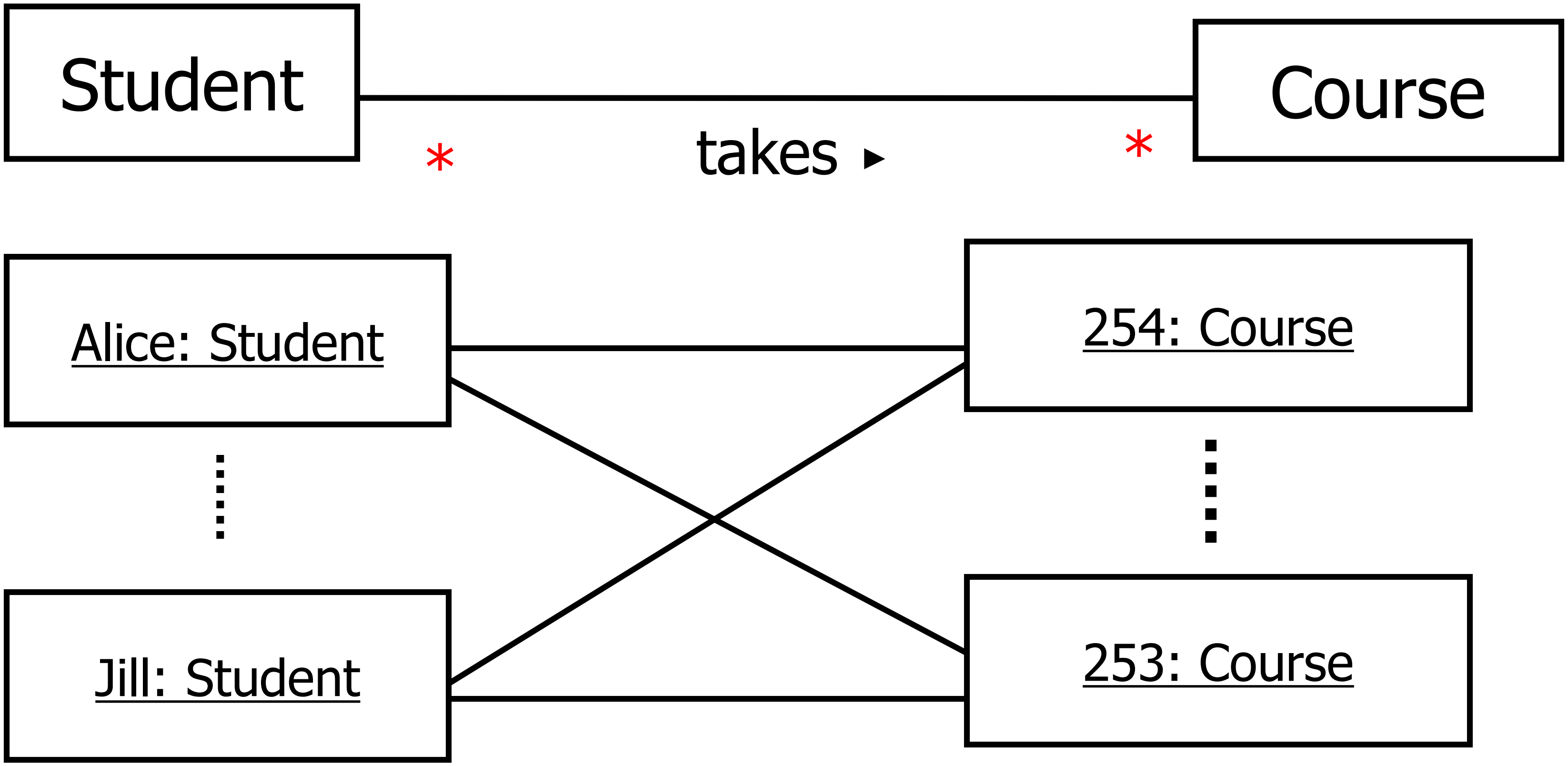
Defining Domain Models Using Class Diagrams

Agenda

- Class Relationships and UML Notations
  - Association - DONE
  - Generalization
  - Realization
  - Dependency
- Class Diagram
- Object Diagram
- Summary

Association - Multiplicity

- A Student can take many Courses and many Students can be enrolled in one Course.



Notes

- One class can be relate to another in a
  - One-to-one
  - One-to-many
  - One-to-one or more
  - One-to-zero or one
  - One-to-a bounded interval (one-to-two through twenty)
  - One-to-exactly n
  - One-to-a set of choices (one-to-five or eight)

Notes

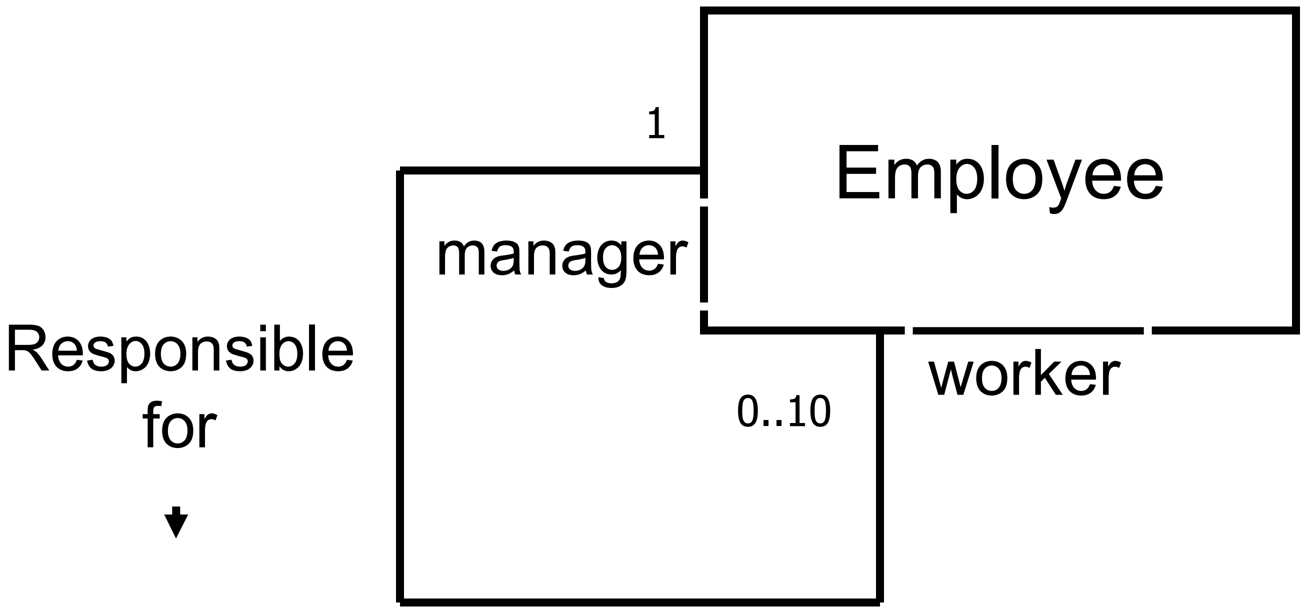
- Multiplicity can be expressed as,
  - Exactly one - 1
  - Zero or one - 0..1
  - Many - 0..* or *
  - One or more - 1..*
  - Exact Number - e.g. 3..4 or 6
  - Or a complex relationship – e.g. 0..1, 3..4, 6..* would mean any number of objects other than 2 or 5

Association - Self

- An association that connects a class to itself is called a self association.

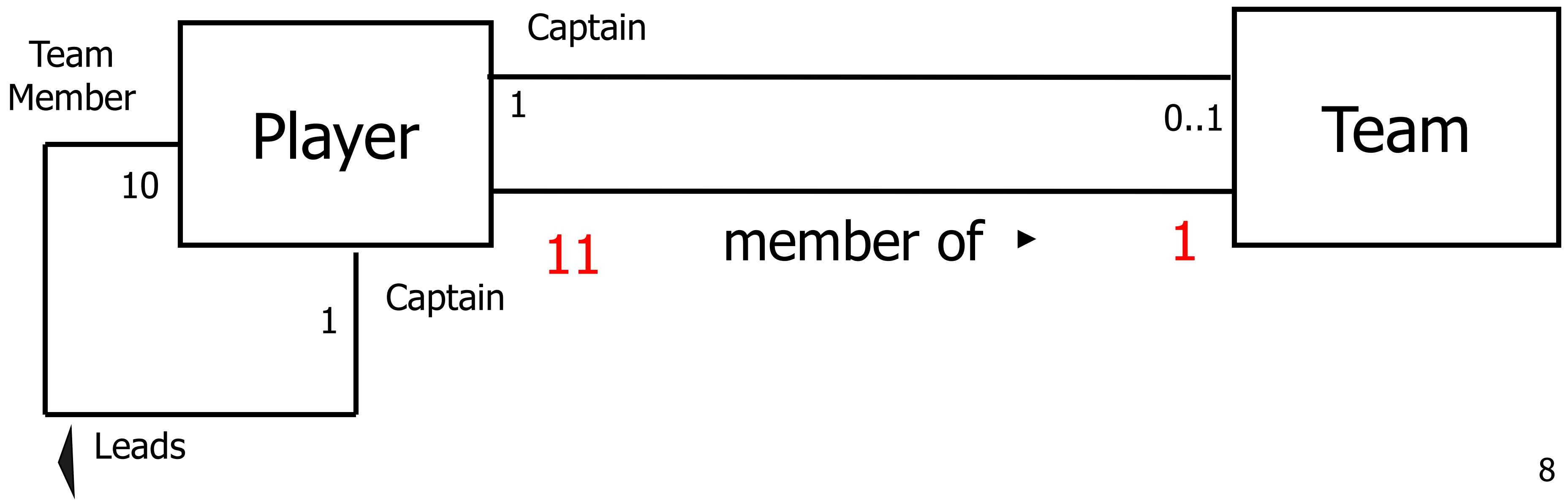
Association - Self

- A Company has Employees.
- A single manager is responsible for up to 10 workers.



Association - Multiplicity

- A cricket team has 11 players. One of them is the captain.
- A player can play only for one Team.
- The captain leads the team members.

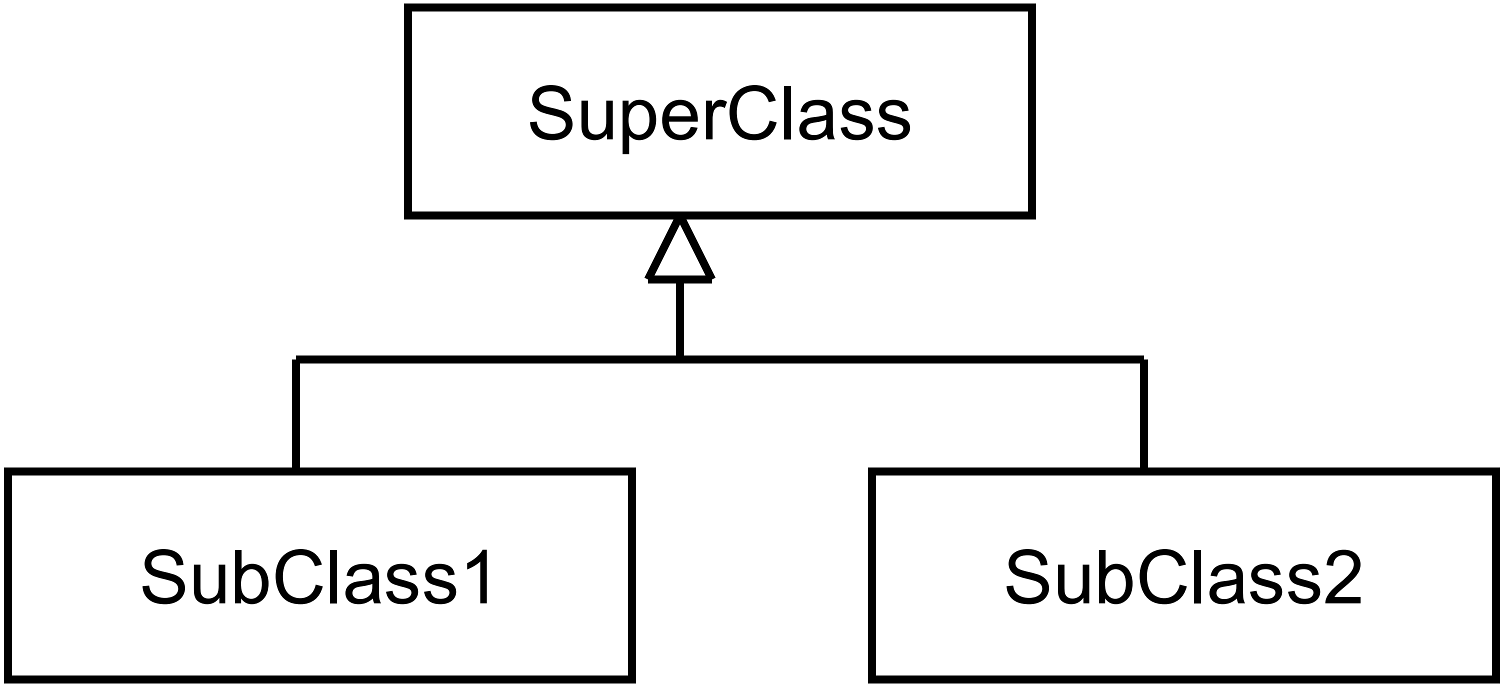


Class Relationships

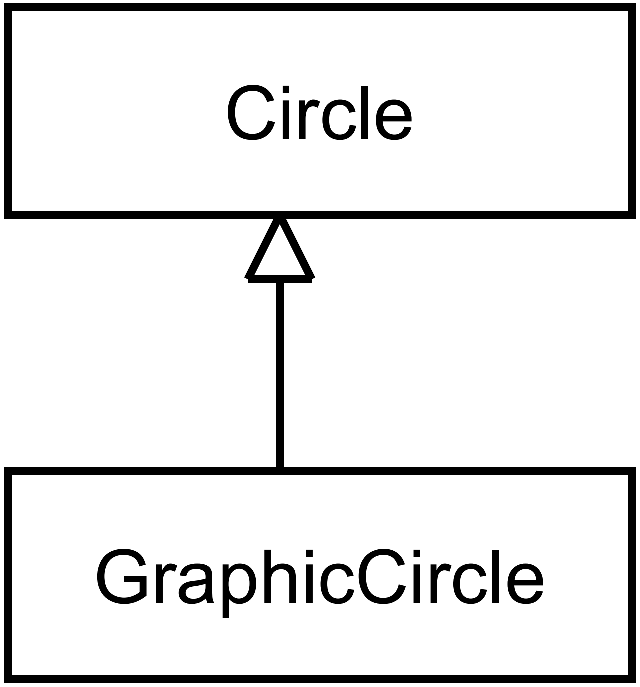
- Association
- *Generalization*
- Realization
- Dependency

Generalization (Inheritance)

- Child class is a special case of the parent class



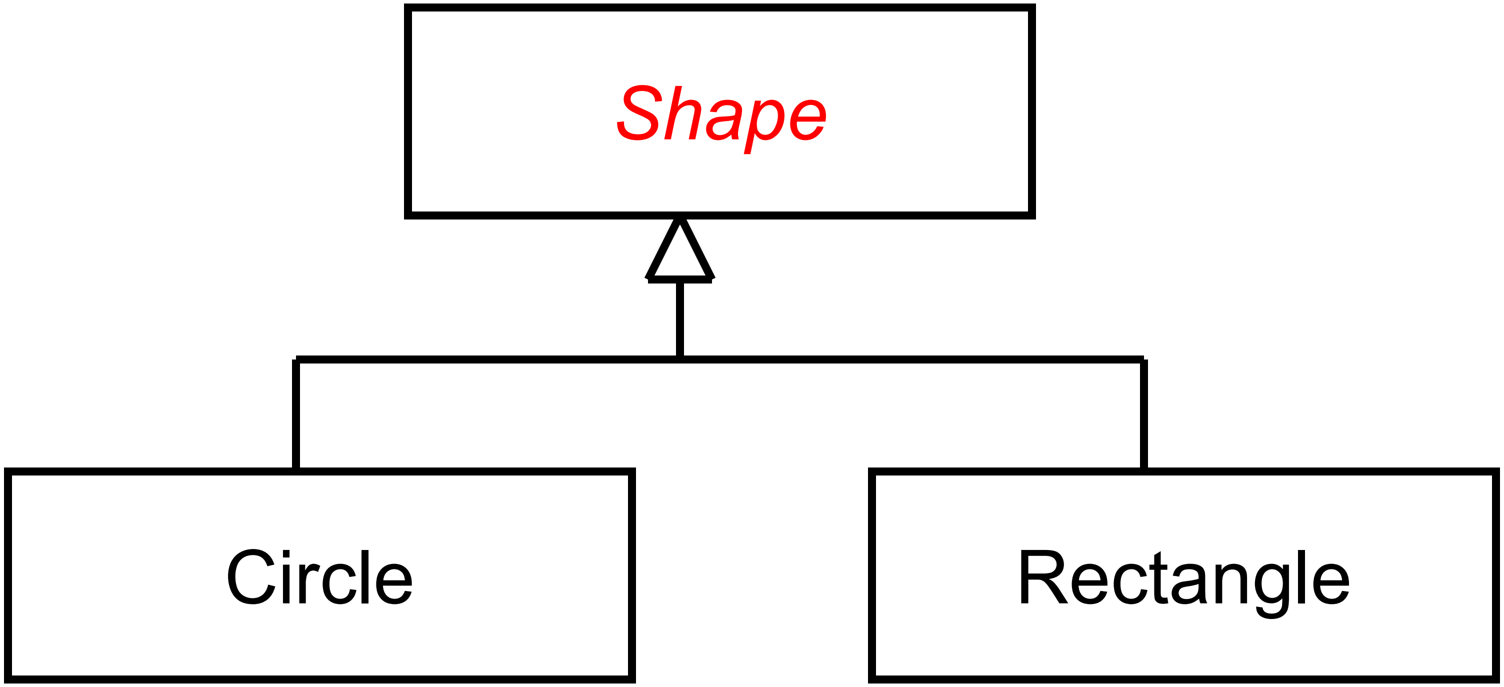
Generalization (Inheritance) e.g.



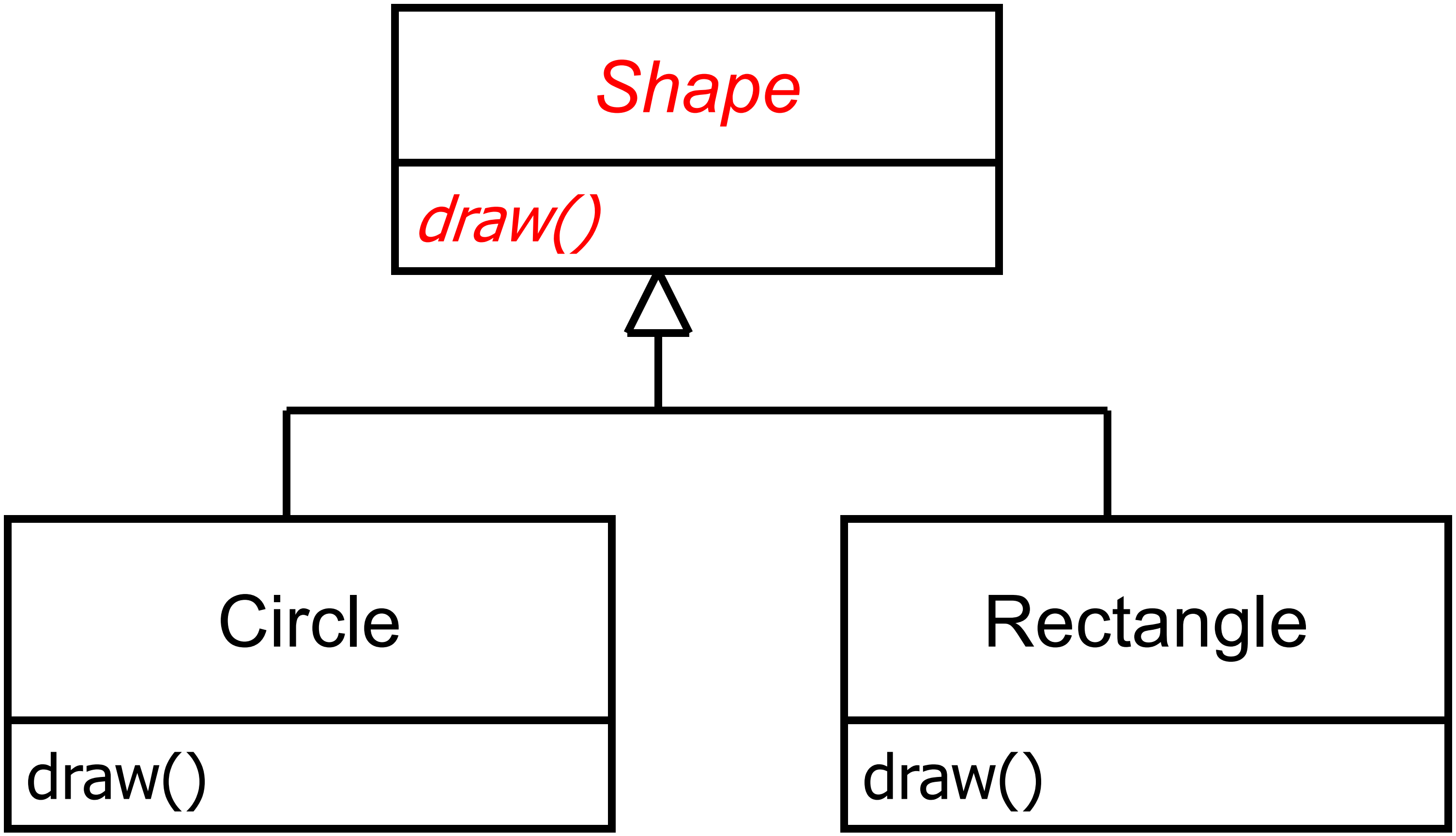
Inheritance - Implementation

```
public class Circle {

}



public class GraphicCircle extends Circle {

}
```

Abstract Class



Abstract Methods (Operations)



Abstract class and method Implementation

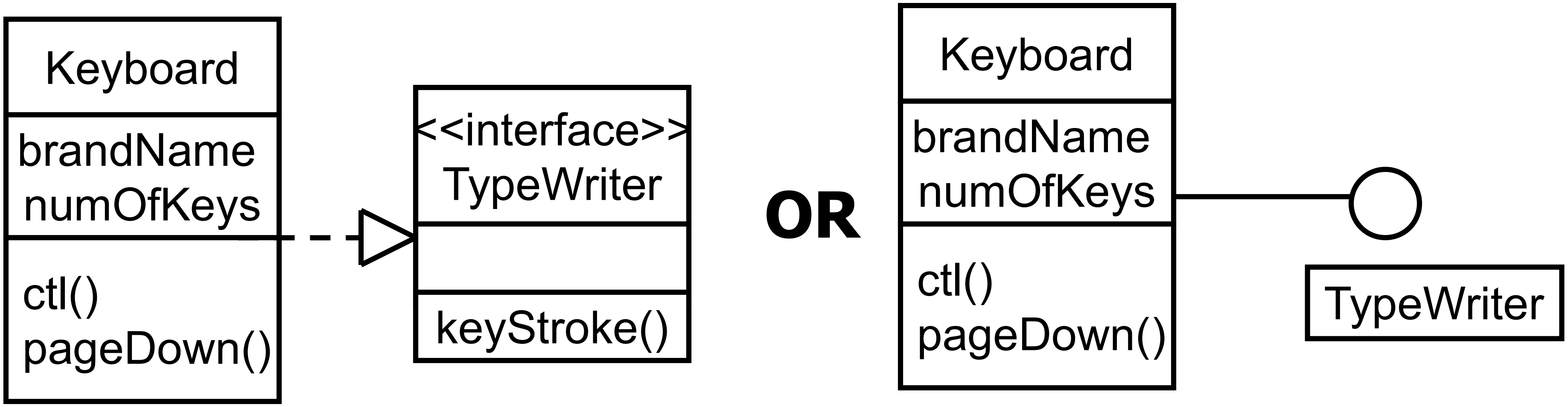
```
public abstract class Shape {
   public abstract draw(); //declare
without implementation
   .........
}

public class Circle {
   public draw(){
   .......
   }
   .....
}
```

Class Relationships

- Association
- Generalization
- *Realization*
- Dependency

Realization- Interface

- Interface is a set of operation the class carries out



Realization - Implementation

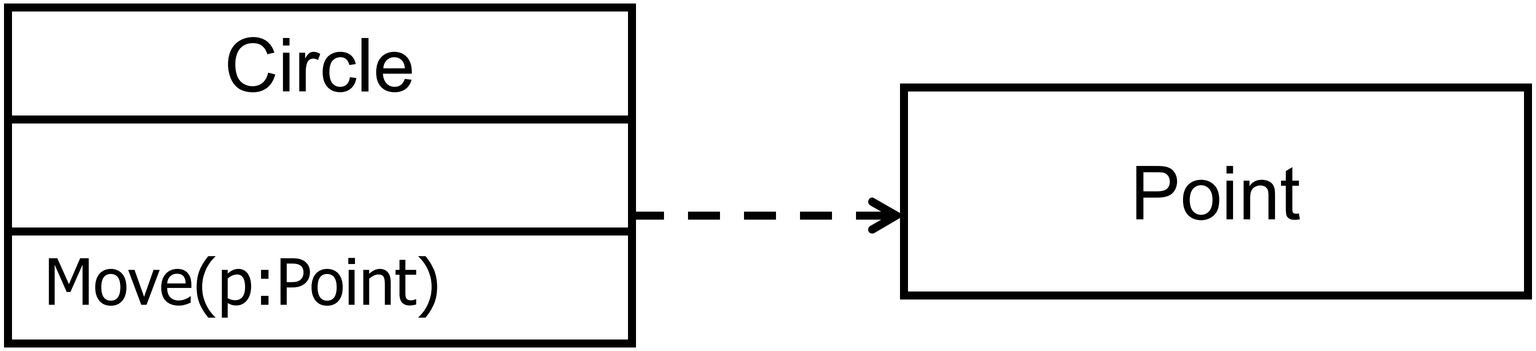
```
public interface TypeWriter {
   void keyStroke()

}

public class KeyBoard implements TypeWriter {
   public void  keyStroke(){
   .........

  }
}
```

Class Relationships

- Association
- Generalization
- Realization
- *Dependency*

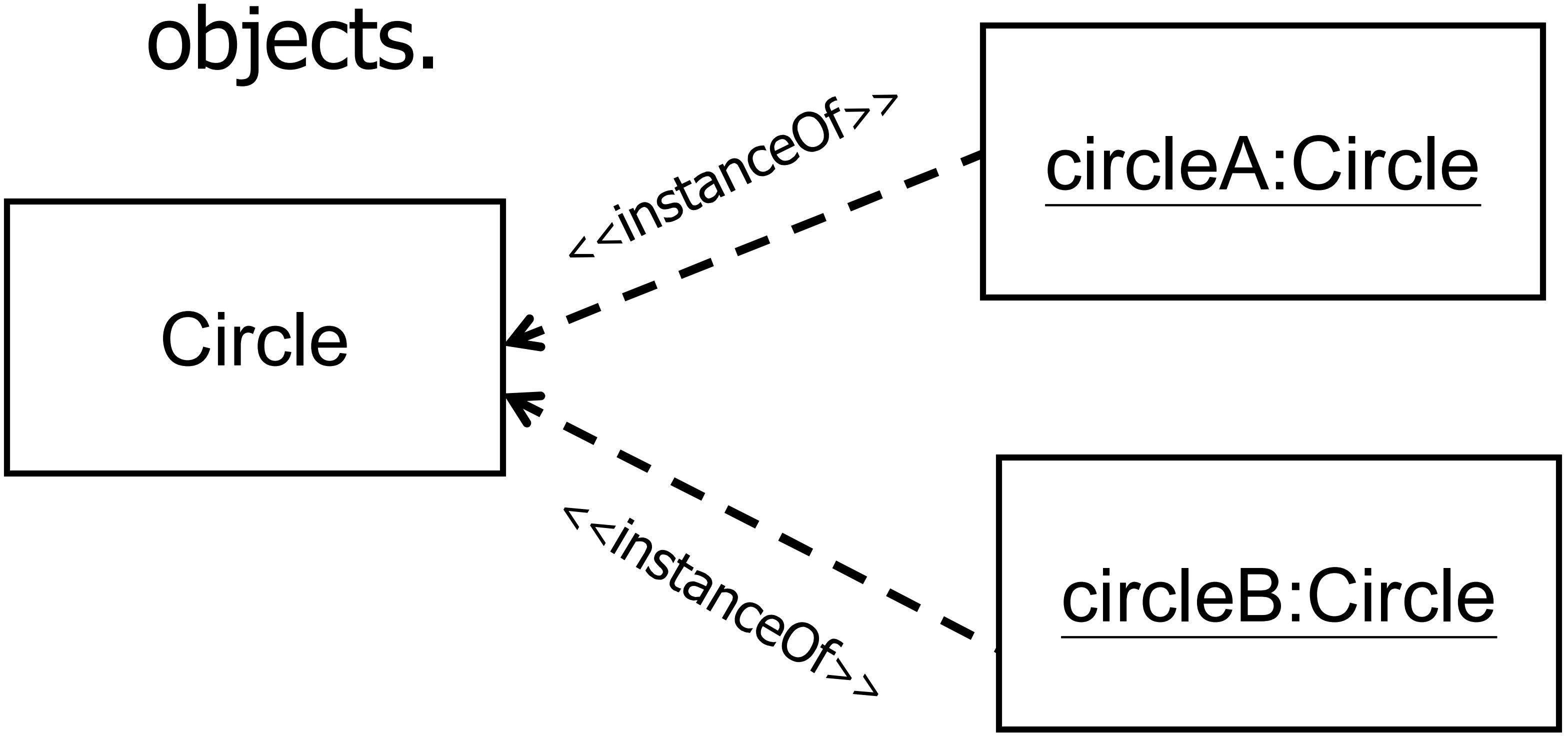
Dependency

- Change in specification of one class can change the other class. This can happen when one class is using another class.



Dependency cont

- Dependency relationship can be used to show relationships between classes and objects.



Class Diagrams

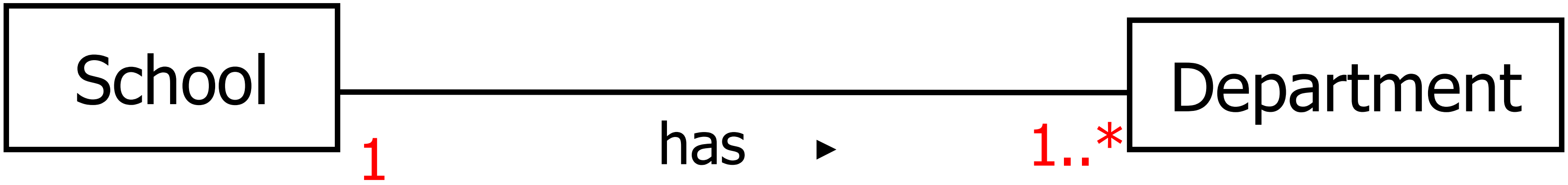
- The UML class diagram consists of several *Classes*, connected with *Relationships*.

Class Diagram - Example

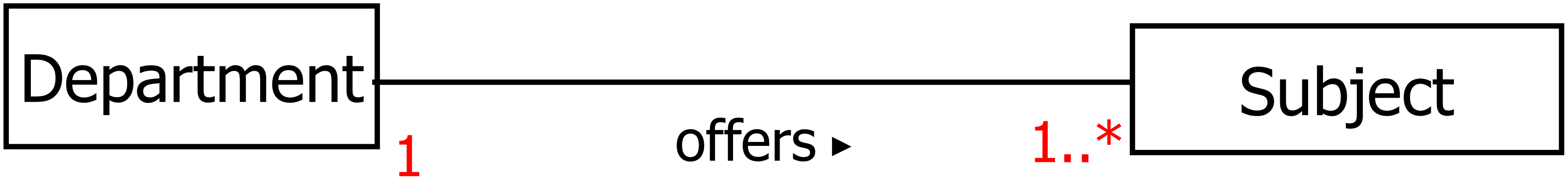
- Draw a class diagram for a information modeling system for a school.
  - School has one or more Departments.
  - Department offers one or more Subjects.
  - A particular subject will be offered by only one department.
  - Department has instructors and instructors can work for one or more departments.
  - Student can enrol in upto 5 subjects in a School.
  - Instructors can teach upto 3 subjects.
  - The same subject can be taught by different instructors.
  - Students can be enrolled in more than one school.

Class Diagram - Example

- School has one or more Departments.

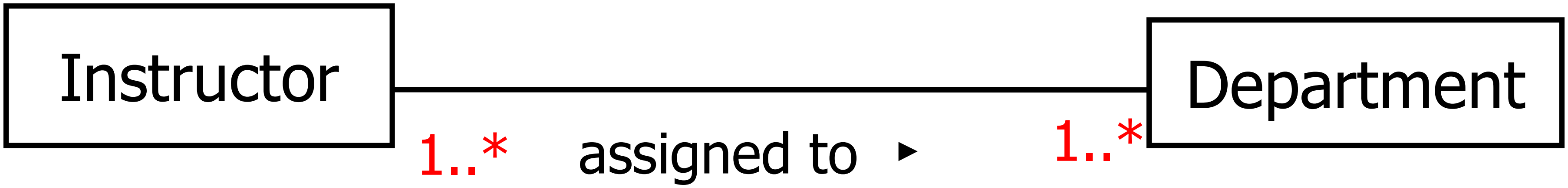


- Department offers one or more Subjects.
- A particular subject will be offered by only one department.



Class Diagram - Example

- Department has Instructors and instructors can work for one or more departments.



Instructor

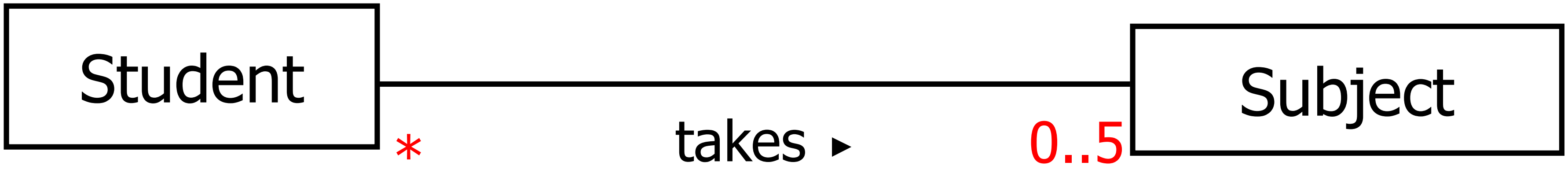
Department

1..*

assigned to ▸

1..*

- Student can enrol in upto 5 Subjects.



Student

Subject

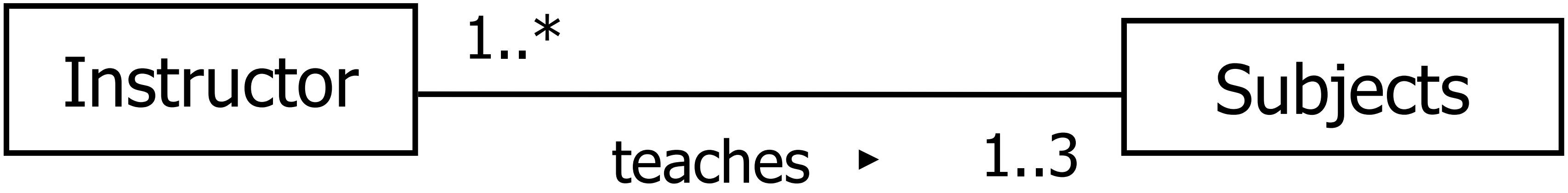
*

takes ▸

0..5

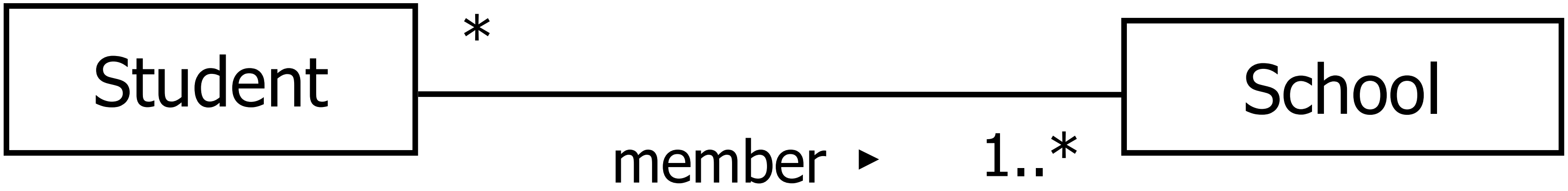
Class Diagram - Example

- Instructors can teach up to 3 subjects.
- The same subject can be taught by different instructors.

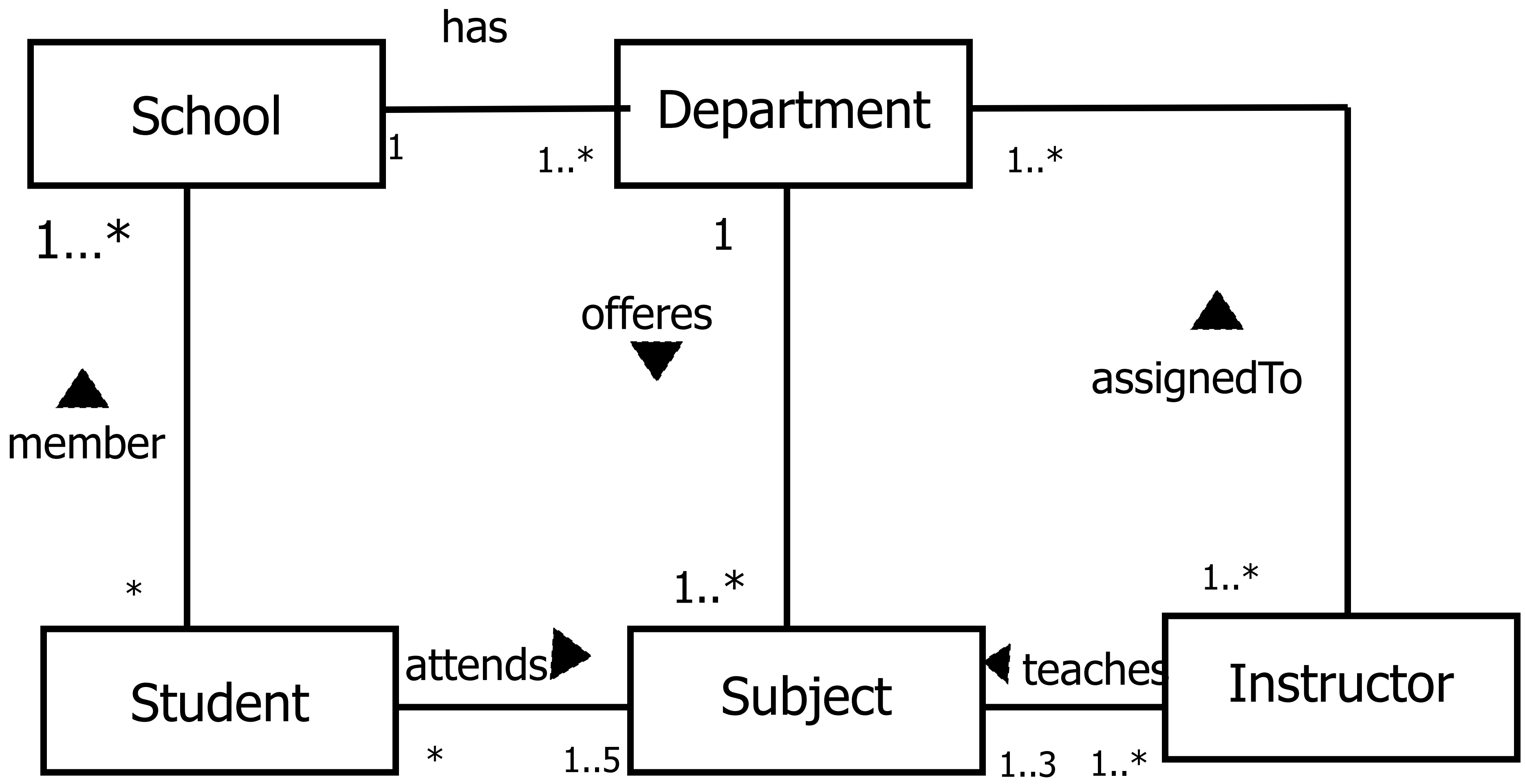


Class Diagram - Example

- Students can be enrolled in more than one school.



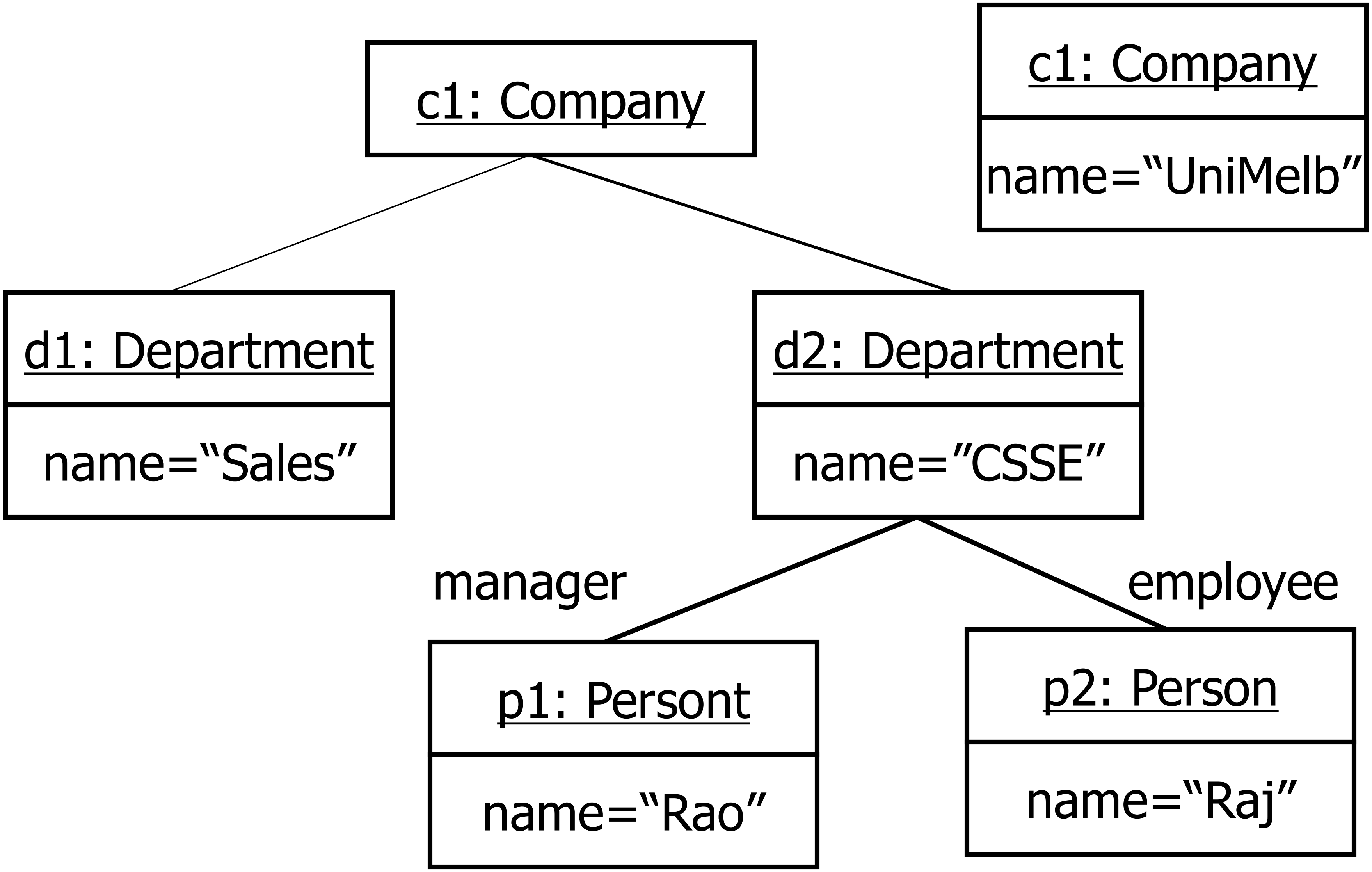
Class Diagram Example



Object Diagram

- Object Diagram shows the relationship between objects.
- Unlike classes objects have a state.

Object Diagram - Example



Summary

- We have discussed the following concepts and UML notations related:
  - Association
  - Generalization
  - Realization
  - Dependency
- How to create a Class Diagram that contains all the above relationships
- Object Diagram for Uni.Dept.system.