Object Oriented Development with Java

(CT038-3-2 and Version VC1)

Class Diagram

System Modeling

Topic & Structure of The Lesson

- Class diagram

Learning outcome

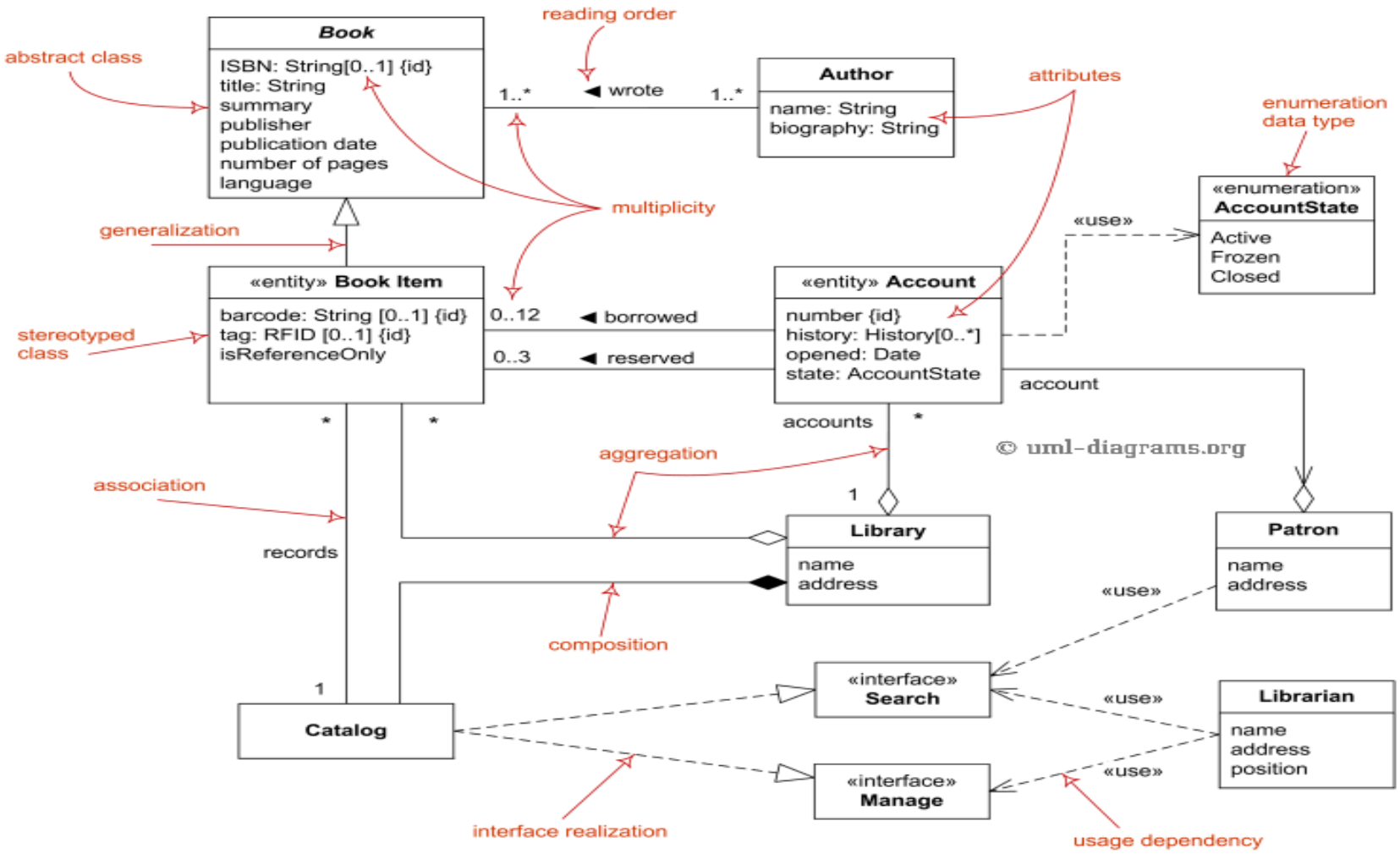
- At the end of this lesson, you will be able to:
  - Model a system using a class diagram

Key terms you must be able to use

If you have mastered this topic, you should be able to use the following terms correctly in your assessments:

-class- association,aggregation,composition

Overall Structure of a Class Diagram



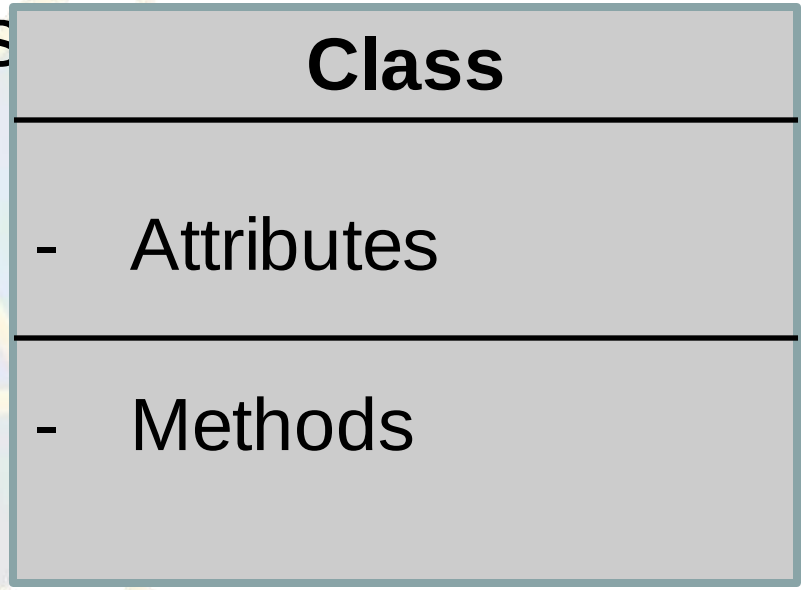
Source: http://www.uml-diagrams.org/class-diagrams-overview.html

Class Diagram

- Used for describing structure and behavior in the use cases
- Provides a conceptual model of the system in terms of entities and their relationships
- Used for requirement capture, end-user interaction
- Detailed class diagrams are used by developers

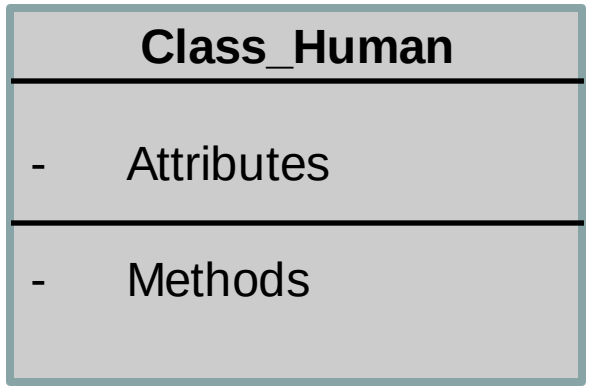
Class Diagram

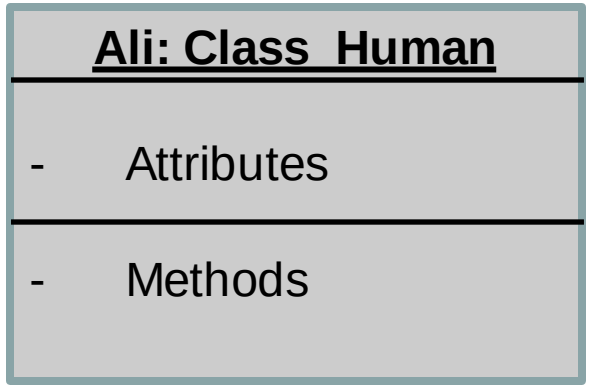
- **Class diagram** models the static structure of a system. It shows relationships between classes, objects, attributes, and operations



Presenting Objects and Classes

- Class
- Object of a class

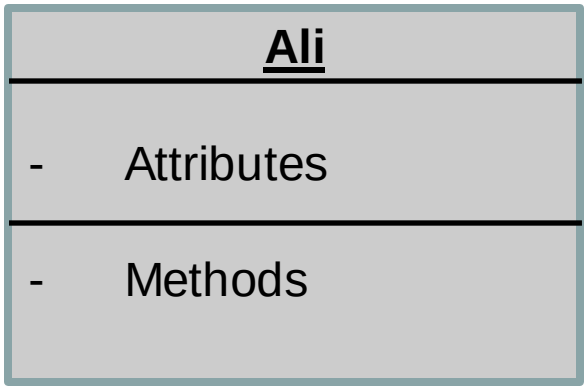


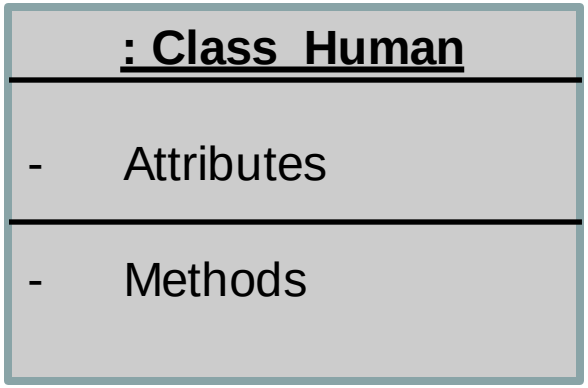


Presenting Objects and Classes

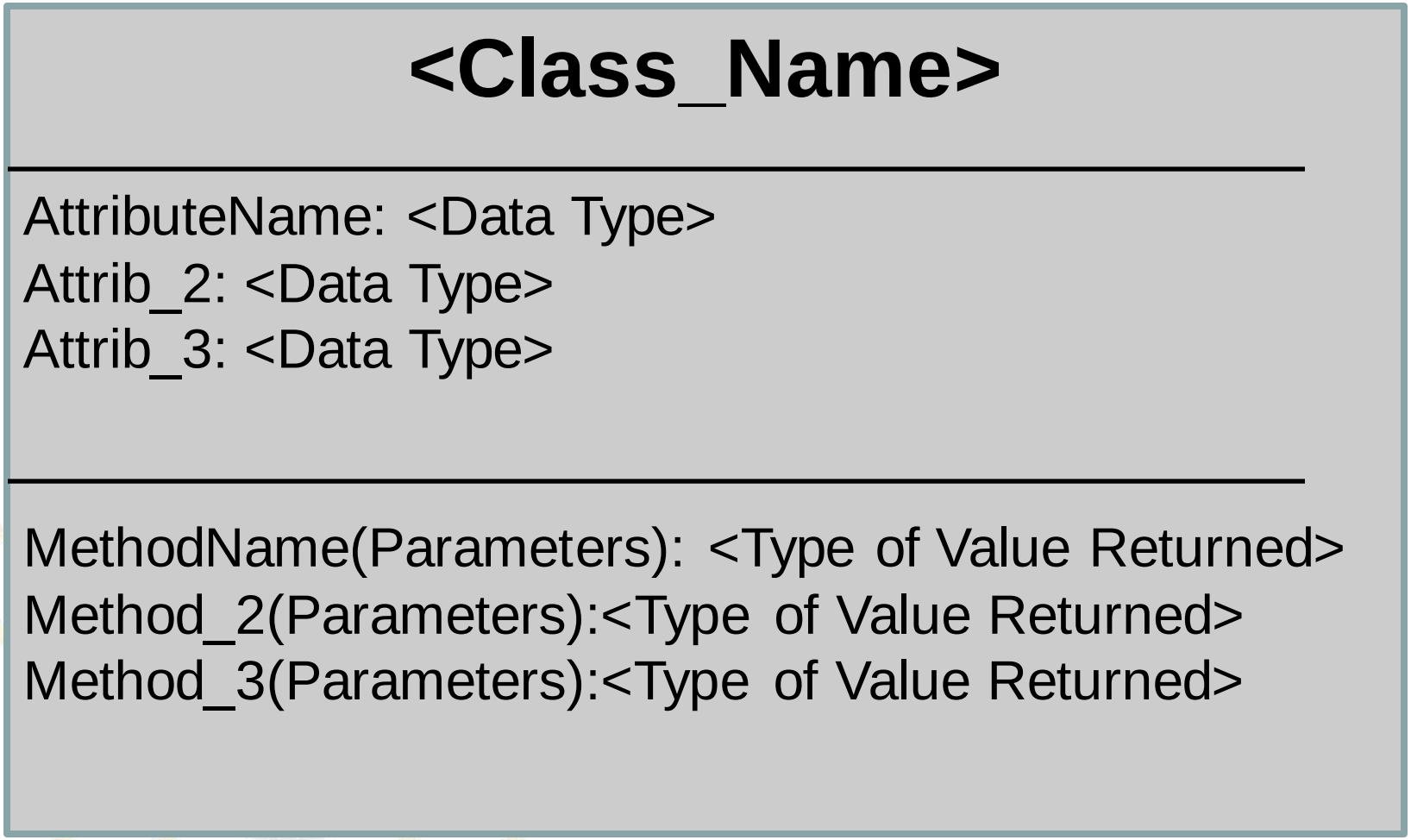
- Object

- Anonymous Object of a class

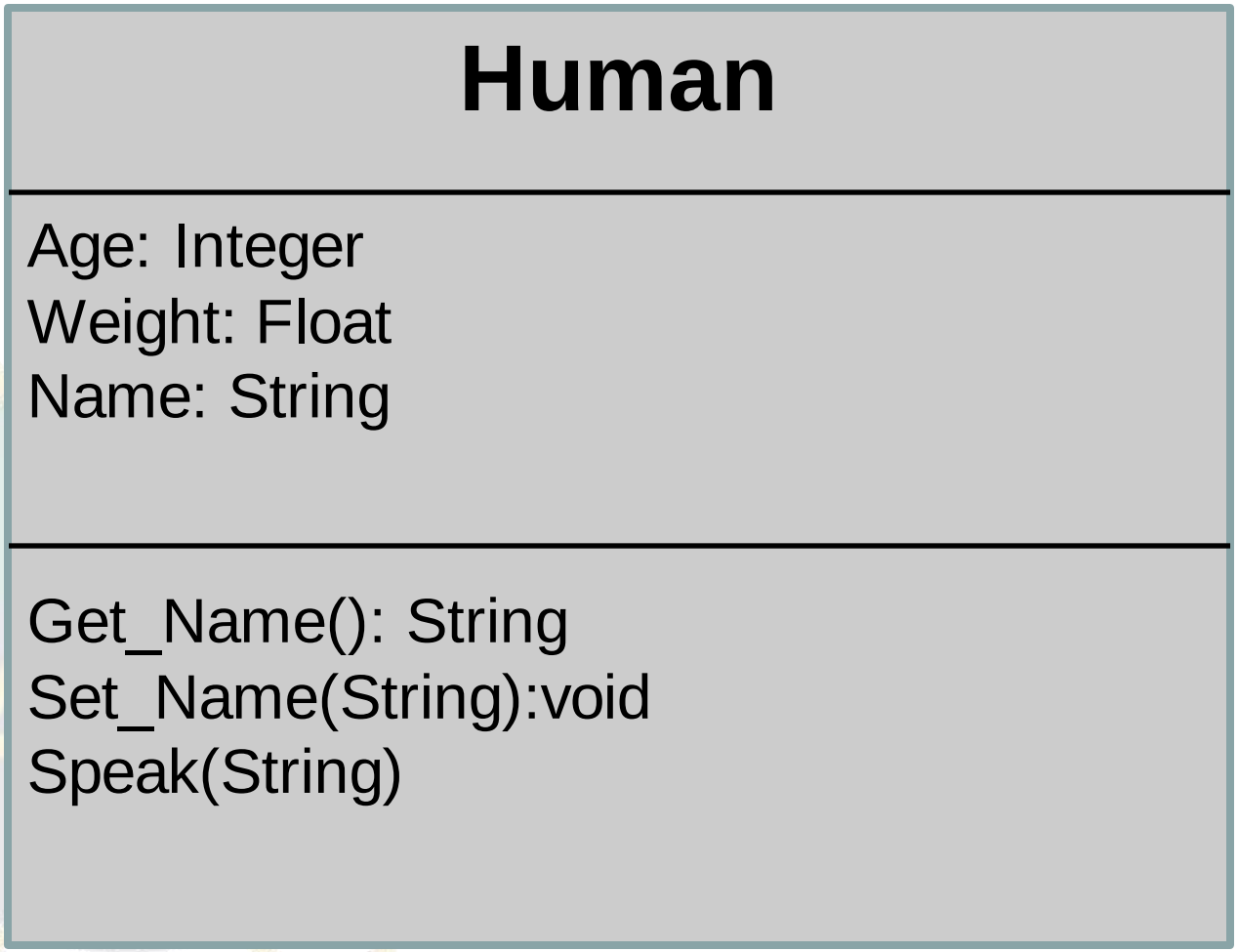




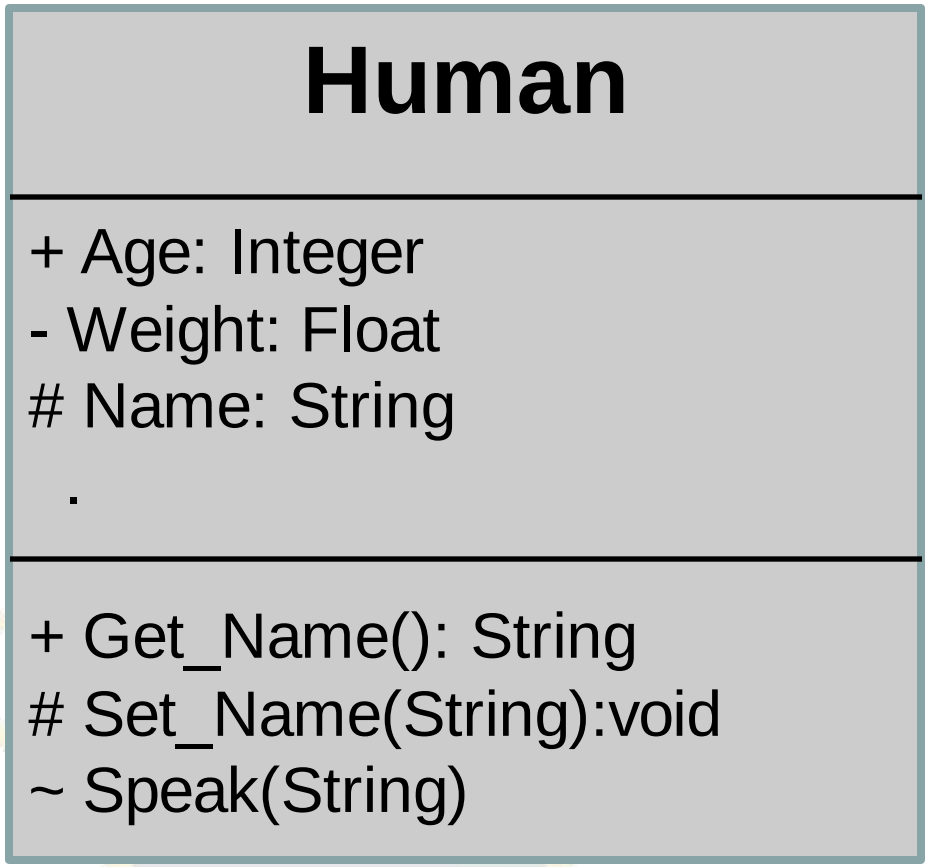
Attributes and Methods



Attributes and Methods



Attributes and Methods Visibility



Sym	Means	Seen by
+	Public	Any
-	Private	Own Class
#	Protected	Own and Child_class
~	Package	Classes in the same Package

Attributes are generally hidden and methods are visible

OO Relationships

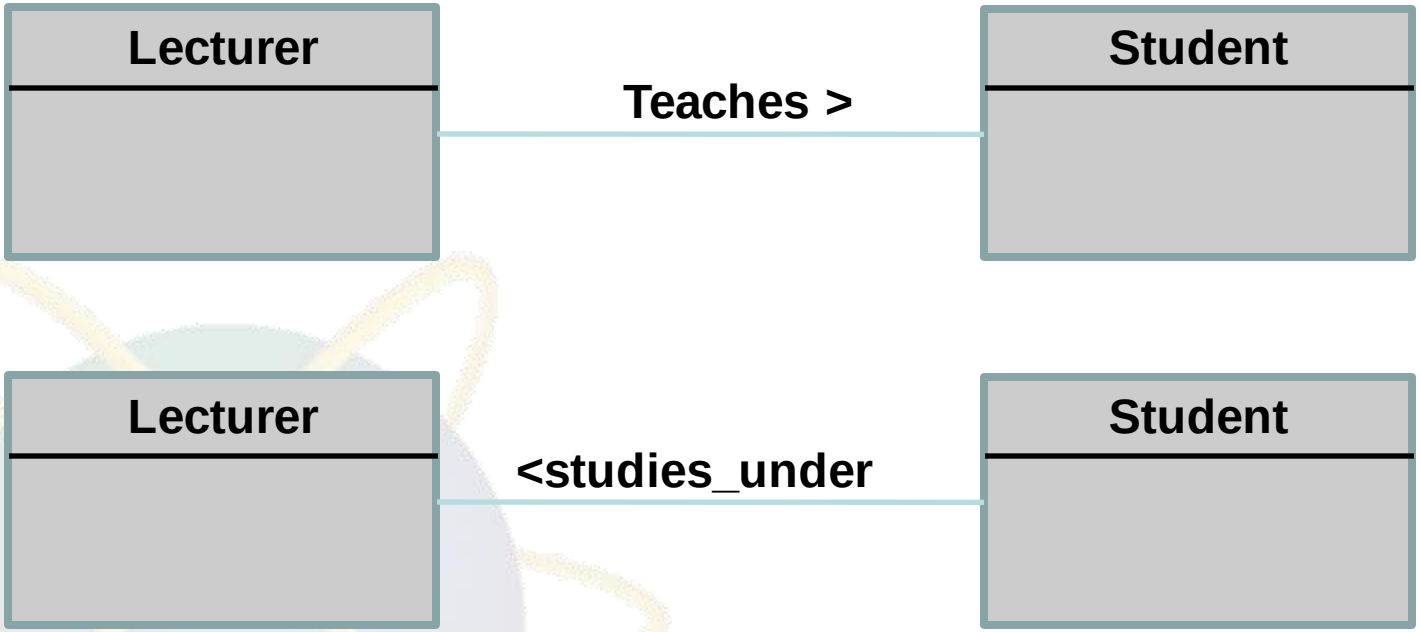
- There are two kinds of relationships:
  - Generalization (parent-child relationship)
  - Association (student enrols in course)

- Associations can be further classified as:
  - Aggregation
  - Composition

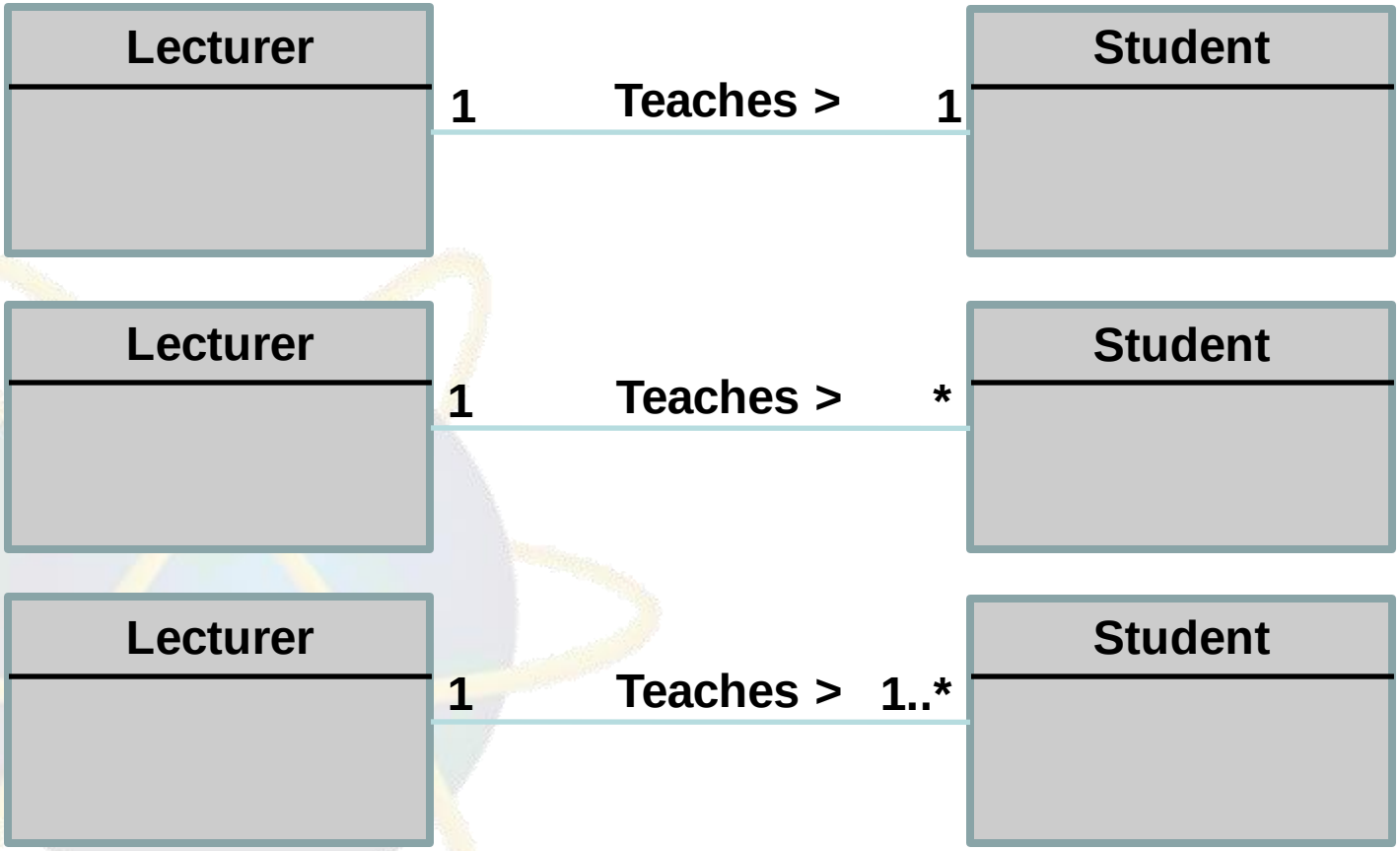
OO Relationships: Association

- Represent relationship between instances of classes
  - ❑ Student enrols in a course
  - ❑ Courses have students
  - ❑ Courses have exams
- Association has two ends
  - ❑ Role names (e.g. enrols)
  - ❑ Multiplicity (e.g. One course can have many students)
  - ❑ Navigability (unidirectional, bidirectional)

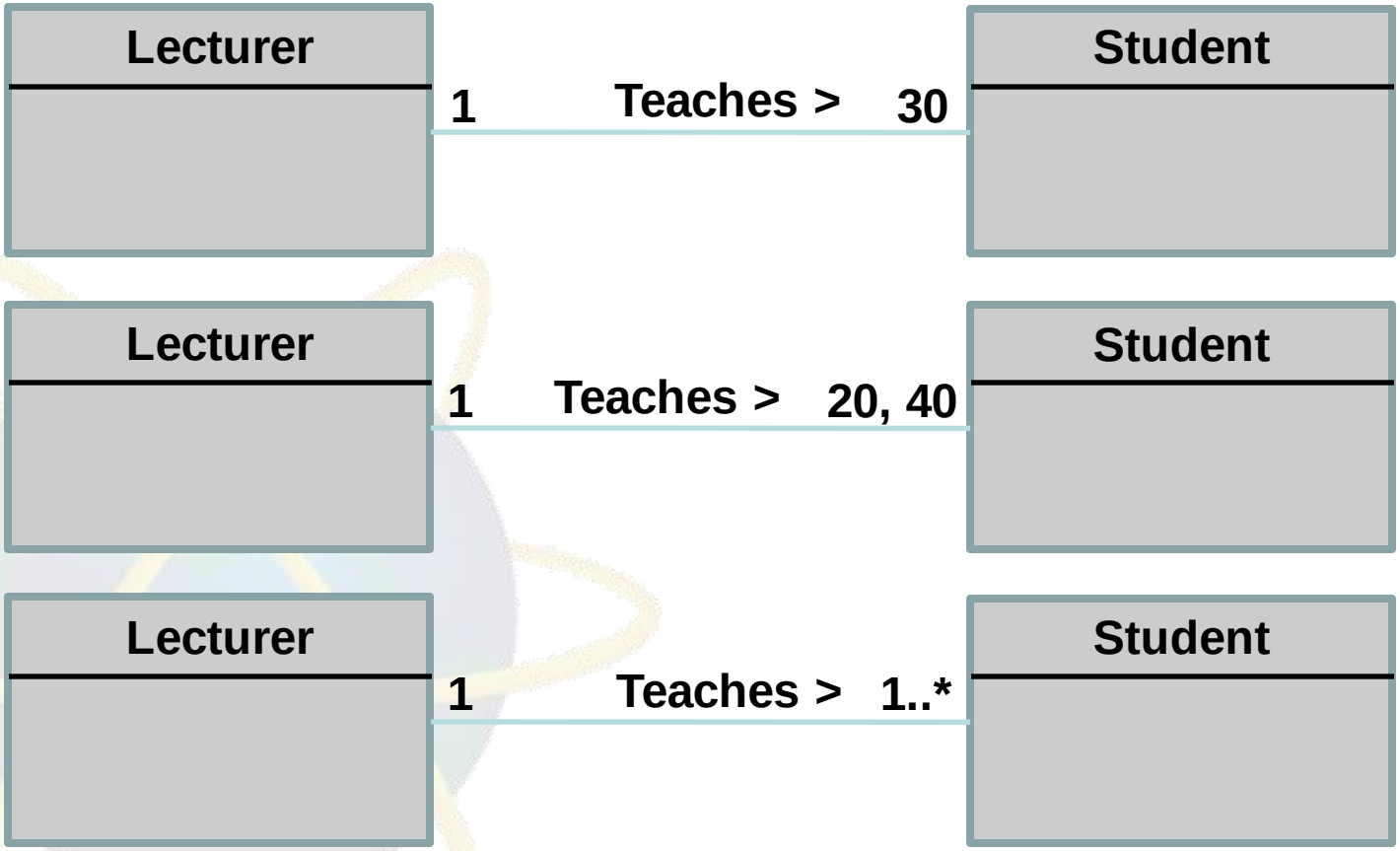
Association



Association

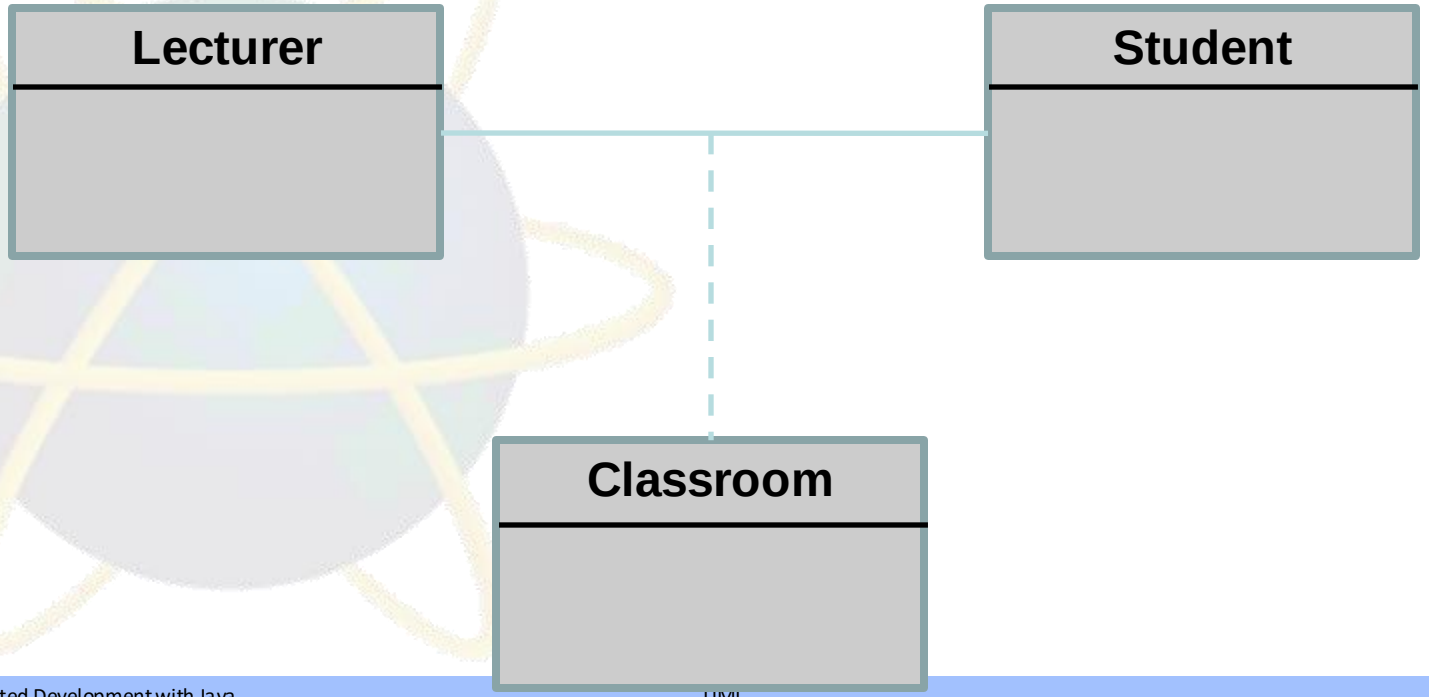


Association



Association Class

- Just like a class, an association can have its own attributes and methods called "Association Class"
- can associate with other classes

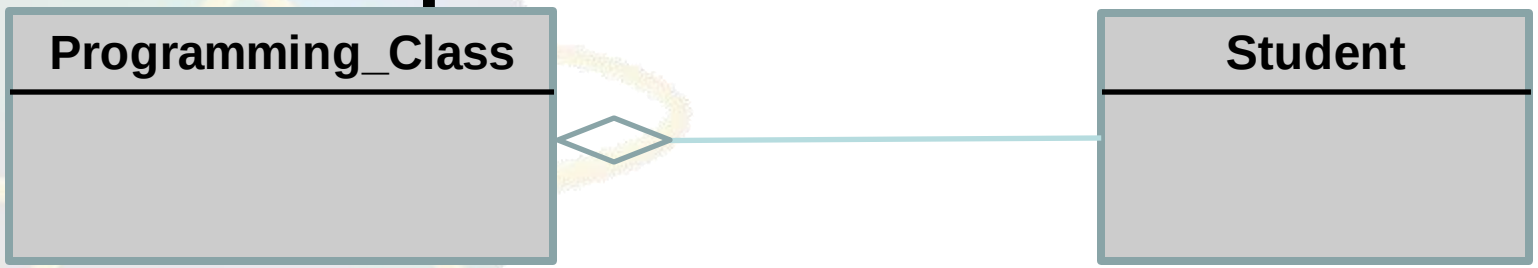


Association: Model to Implementation

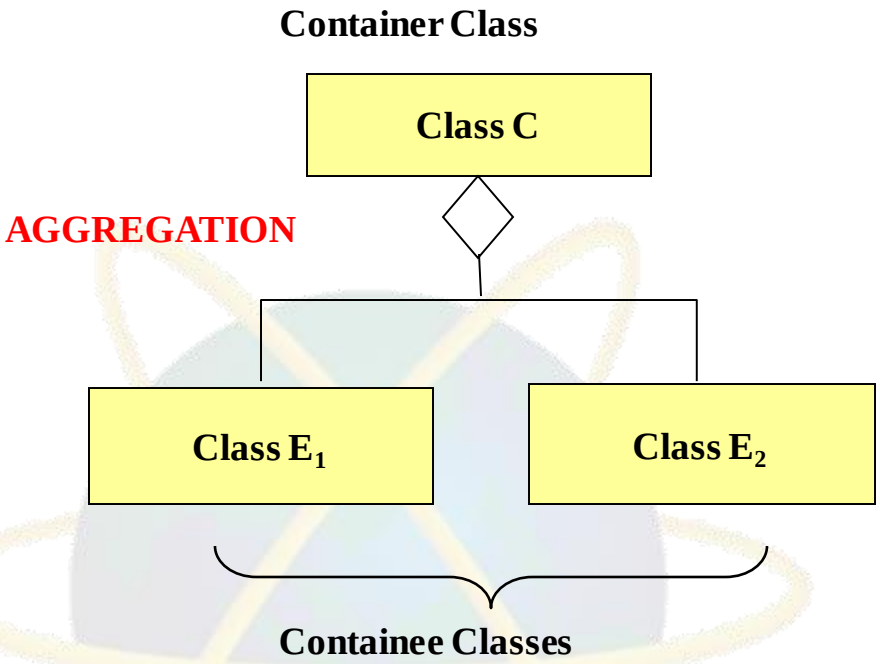

```
Class Student {
    Course enrolls[4];
}

Class Course {
        Student have[];
}
```

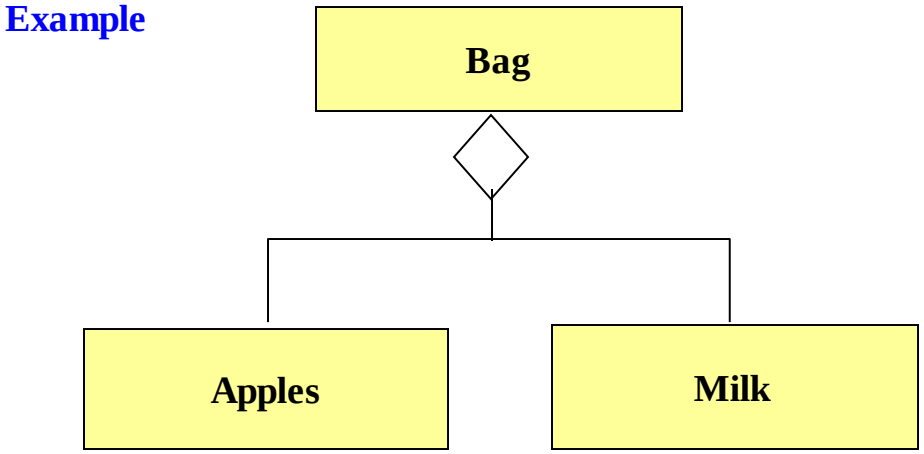
Class Diagram Associations: Aggregation

- Aggregation is a strong type of association that shows that a class CAN BE PART OF another class
- **kind of Container-Containee relationship**

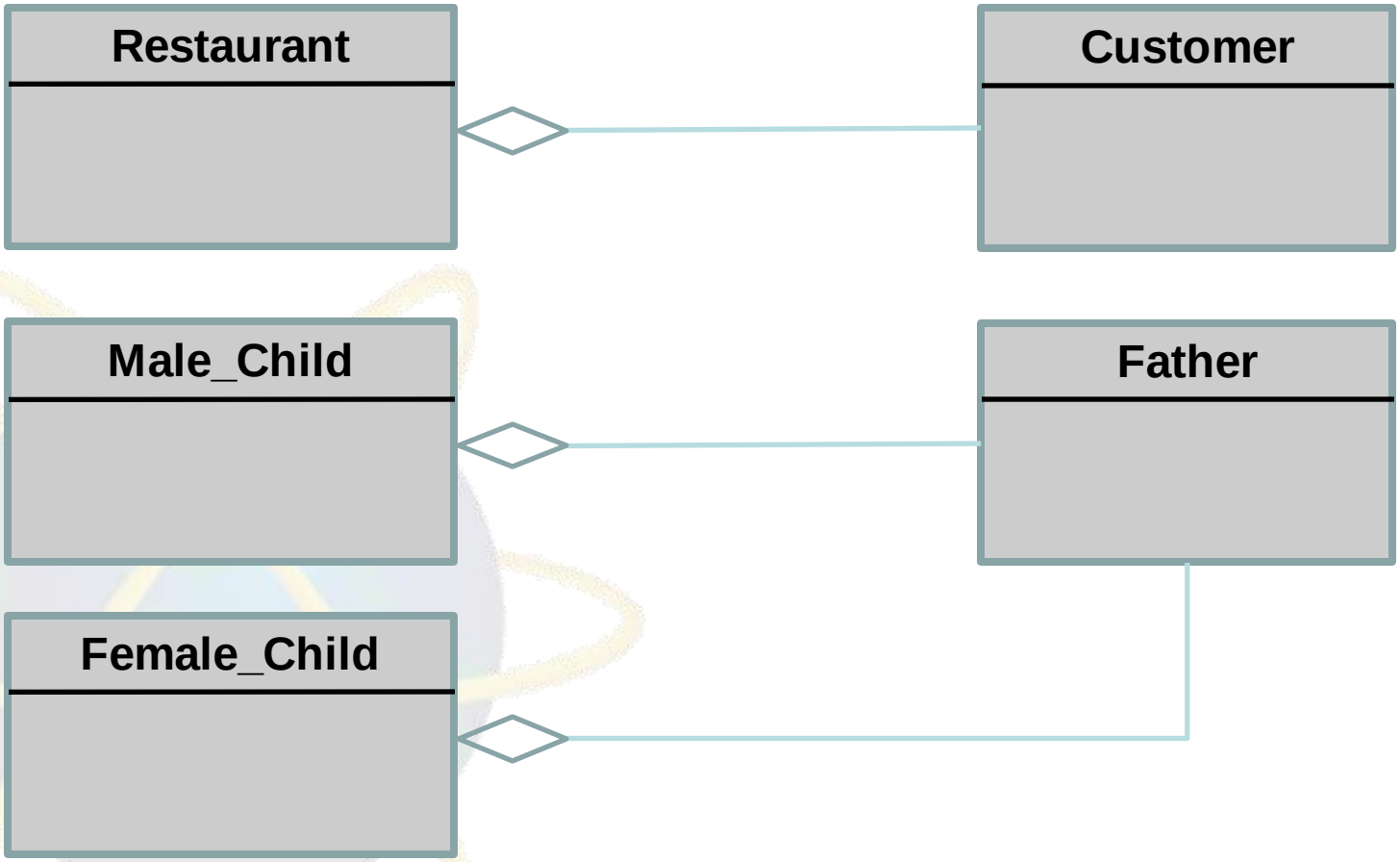


Class Diagram Associations: Aggregation





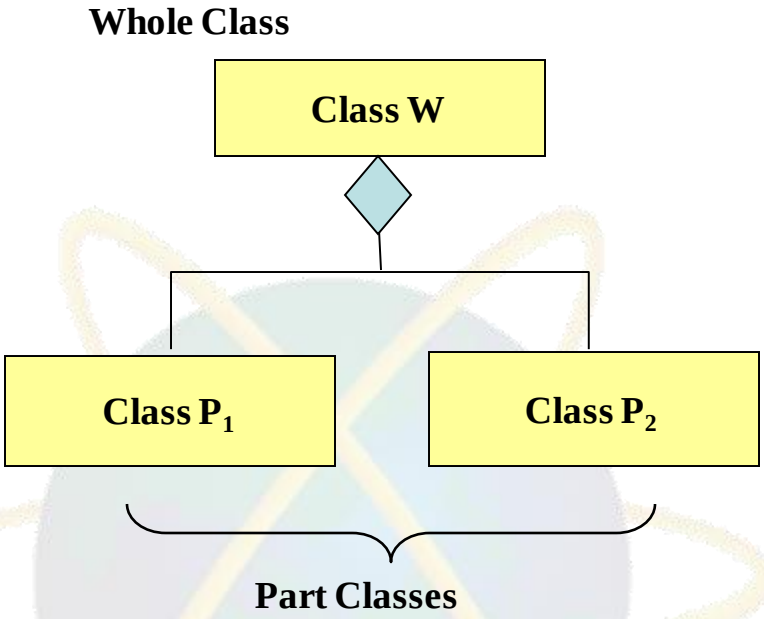
Aggregation Examples

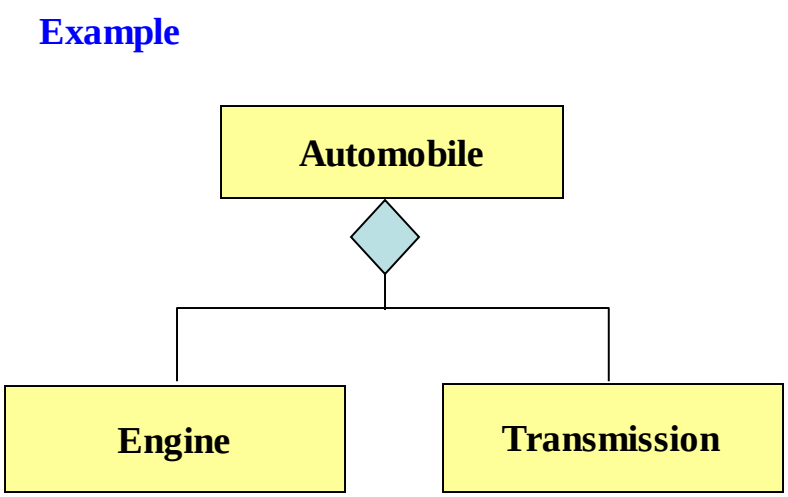


Class Diagram Associations: Composition

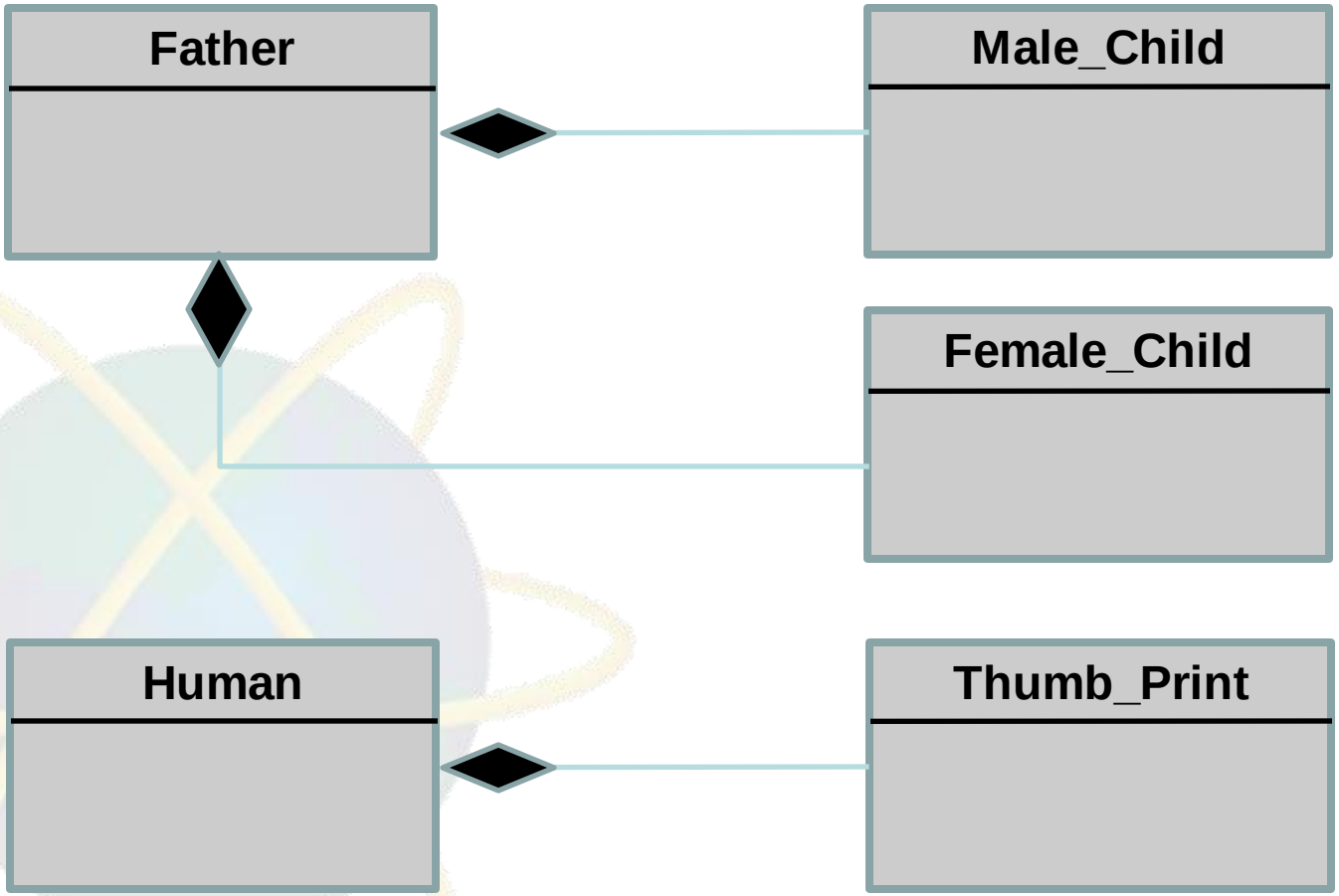
- Composition is a stronger type of association that tells that a class BELONG TO another class.
- kind of Whole-Part relationship

Class Diagram Associations: Composition





Composition Examples

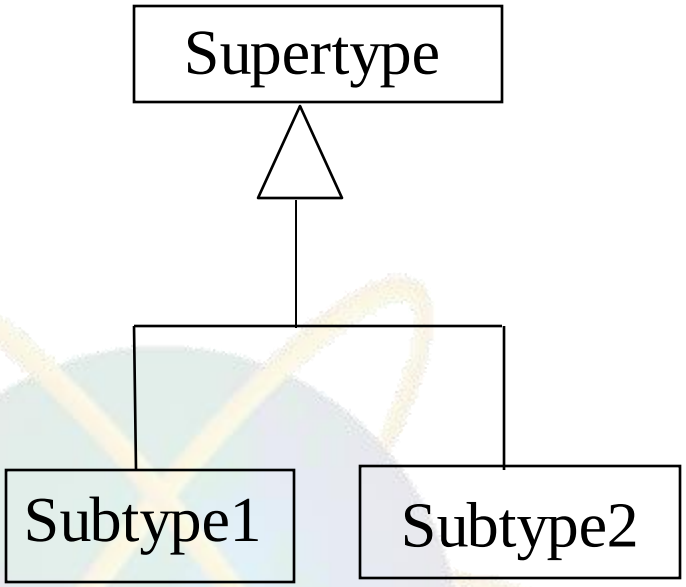


Aggregation vs. Composition

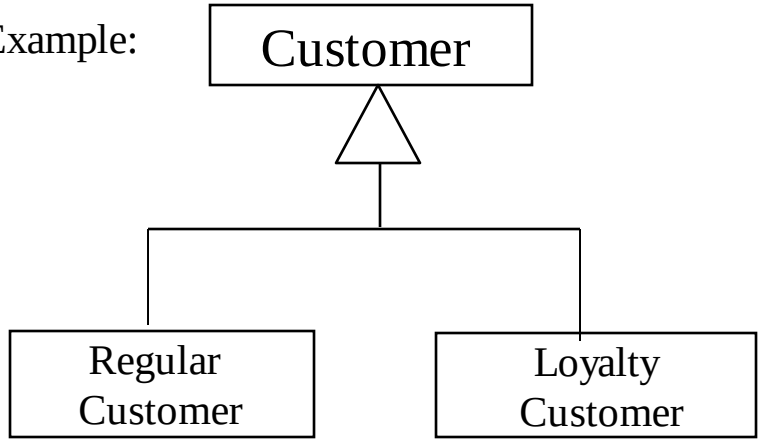
- **Composition** is really a strong form of **aggregation**
  - components have only one owner
  - components cannot exist independent of their owner
  - components live or die with their owner
  - e.g. Each car has an engine that can not be shared with other cars.

- **Aggregations** may form "part of" the aggregate, but may not be essential to it. They may also exist independent of the aggregate.
  - e.g. Apples may exist independent of the bag.

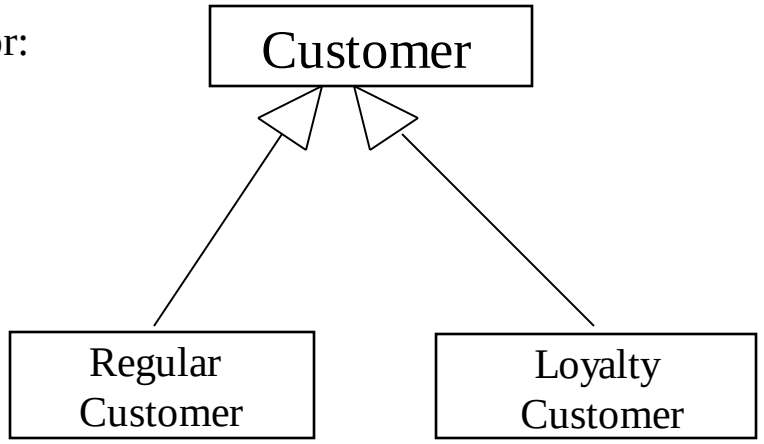
OO Relationships: Generalization



Example:



or:

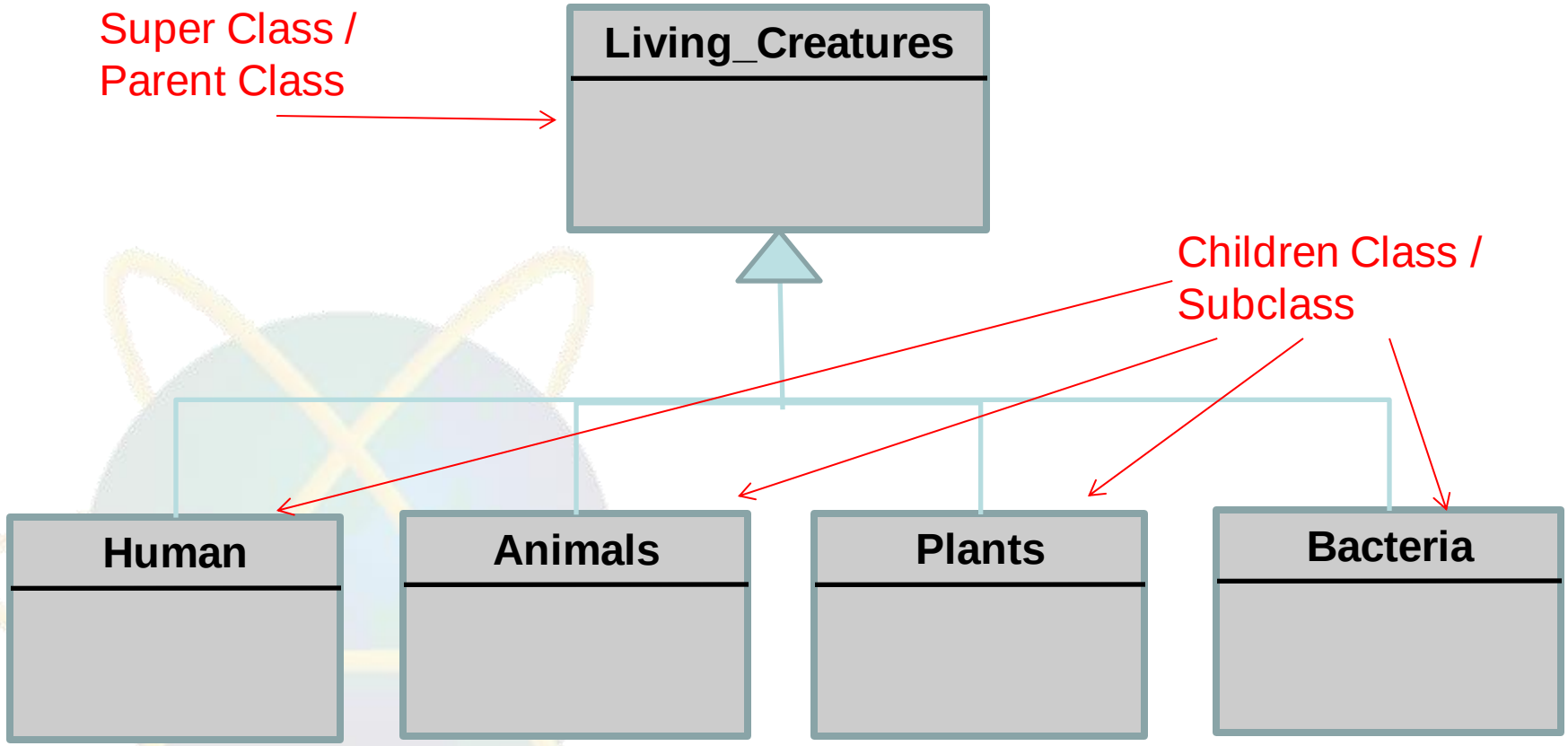


- Generalization expresses a parent/child relationship among related classes.
- Used for abstracting details in several layers

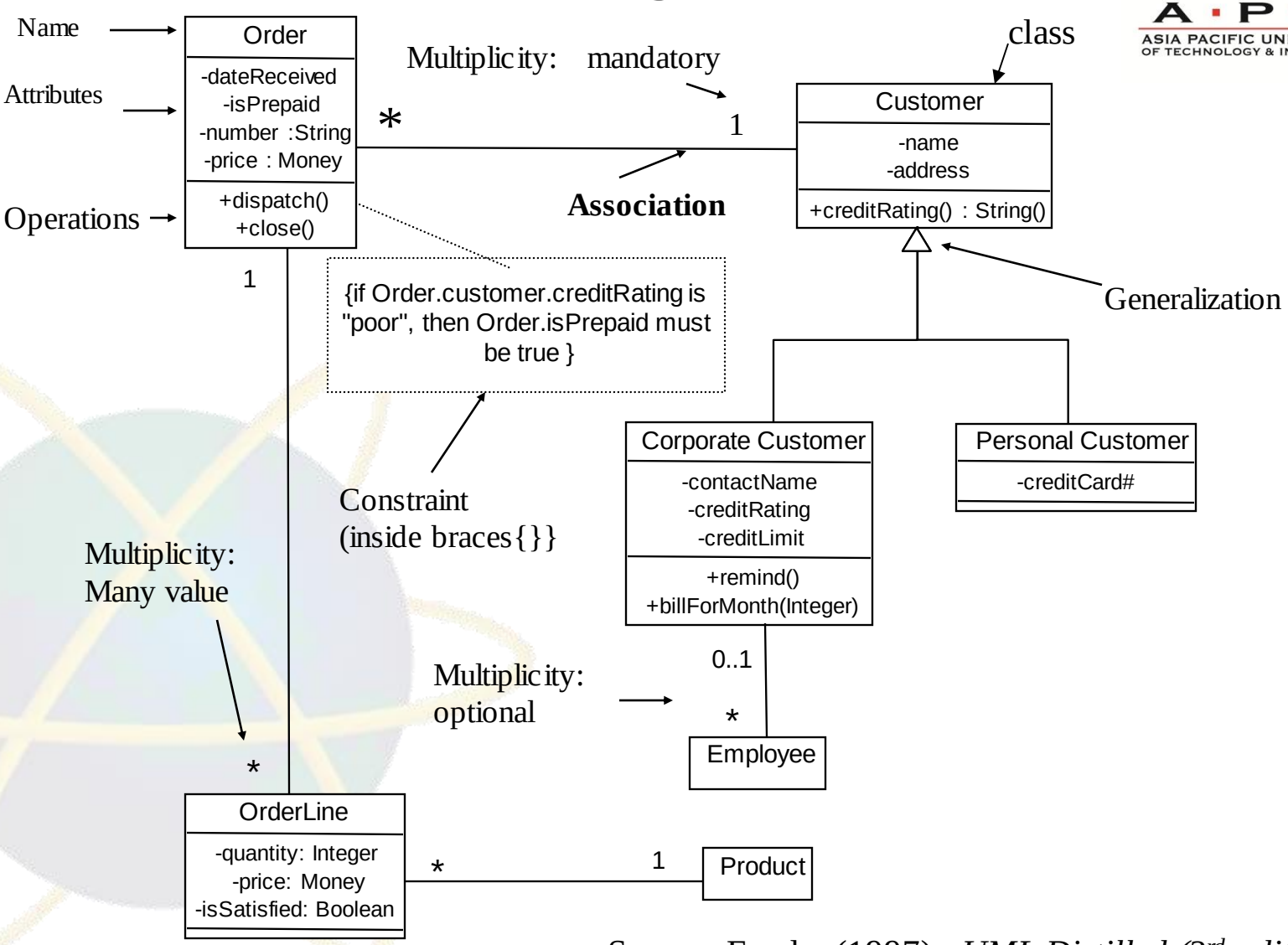
Generalization (Inheritance)

- Child Class CAN inherit attributes and methods of the Parent Class.
- Parent Class CANNOT inherit attributes and methods of the Child Class.
- Child Class inherits the PUBLIC and the PROTECTED attributes and methods of the Parent Class.
- Child Class WILL NOT inherit the PRIVATE attributes and methods of the Parent Class.

Generalization (Inheritance)



Overall Structure of a Class Diagram



Source: Fowler (1997) - *UML Distilled (3rd edition)*

Exercise

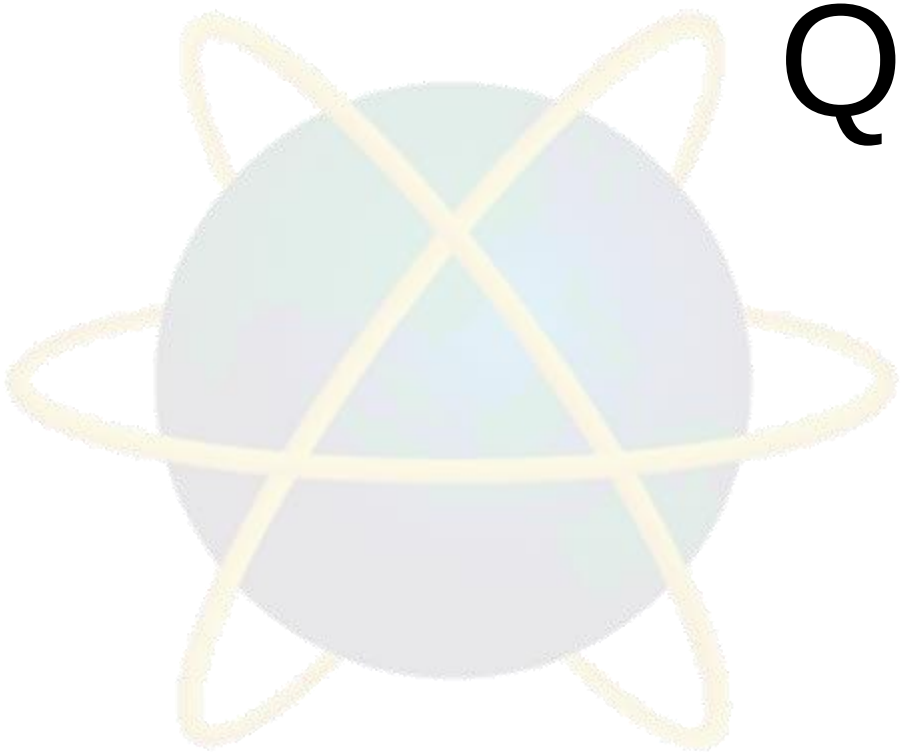
Draw a class diagram for the following scenario:
A car rental company wants to develop an automated system that would handle car reservations, customer billing, and car auctions.
Normally, a customer reserves a car, picks it up, and then returns it after a certain period of time. At the time of pick up, the customer has the option to buy or waive collision insurance on the car. When the car is returned, the customer receives a bill and pays the specified amount. In addition to renting out cars, every six months or so, the auto rental company auctions the cars that have accumulated over 20,000 miles.

Quick Review Questions

- What is a class diagram?
- What are the components of a class diagram?
- Draw the class diagram for borrowing a book from the library

Summary of Main Teaching Points

-Class diagram and its componets



Q & A

Next Session

-Activity diagram