Object Oriented Programming using Python (I)

Lecture(8)

UML (Unified Modeling Language)

Prepared by: ***Dr. Rula Amjed Hamid***
University of Information Technology and Communications
College of Business Informatics

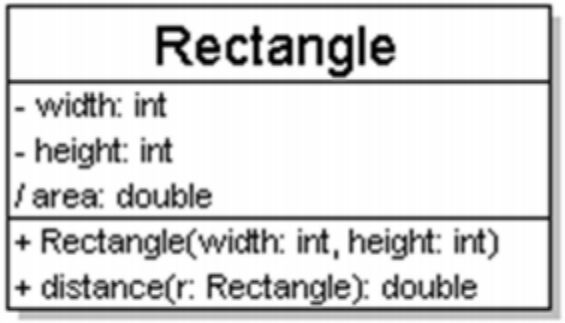
What is UML?

A **UML diagram** is a **diagram** based on the **UML** (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.

UML class diagram

UML class diagram: a picture of – the classes in an OO system and their attributes and methods

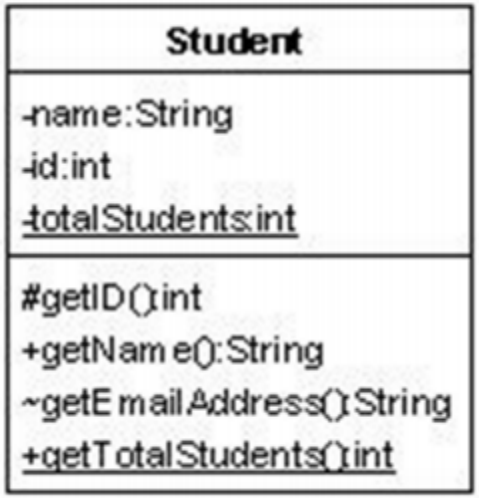
- connections between the classes
- that interact or inherit from each other



Description :

class name in top of box

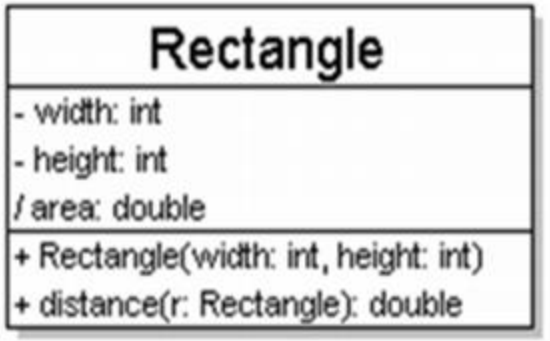
- attributes – should include all fields of the object
- operations / methods ()

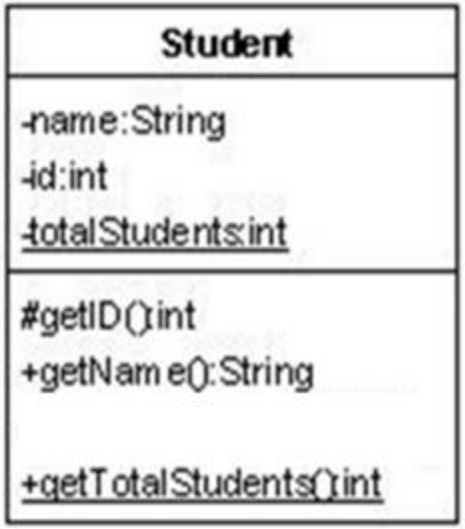


Class attributes (= fields)

- attributes (fields, instance variables)
  - *visibility name* : *type* [*count*] = *default_value*
  - visibility: + public
    # protected
    - private

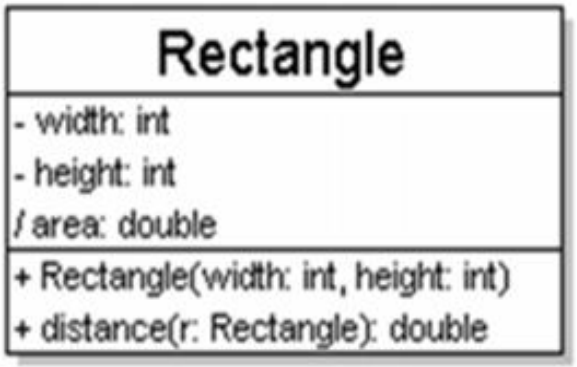
    / derived
  - underline static attributes
  - **derived attribute**: not stored, but can be computed from other attribute values
  - attribute example:
    - balance : double = 0.00

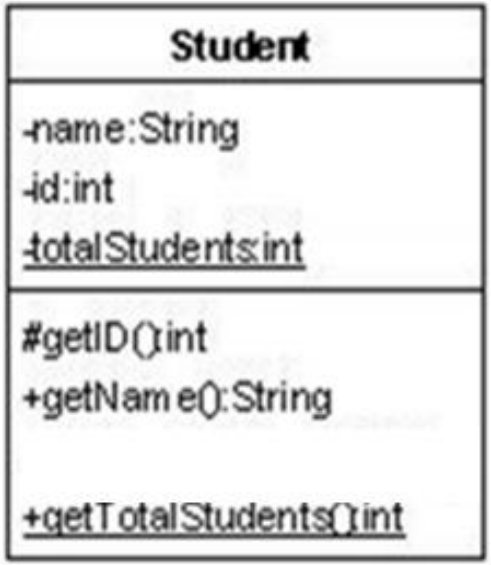




Class operations / methods

- operations / methods
  - *visibility name* (*parameters*) : *return_type*
  - visibility: + public
    # protected
    - private
  - underline static methods
  - parameter types listed as (name: type)
  - method example:
    + distance(p1: Point, p2: Point): double





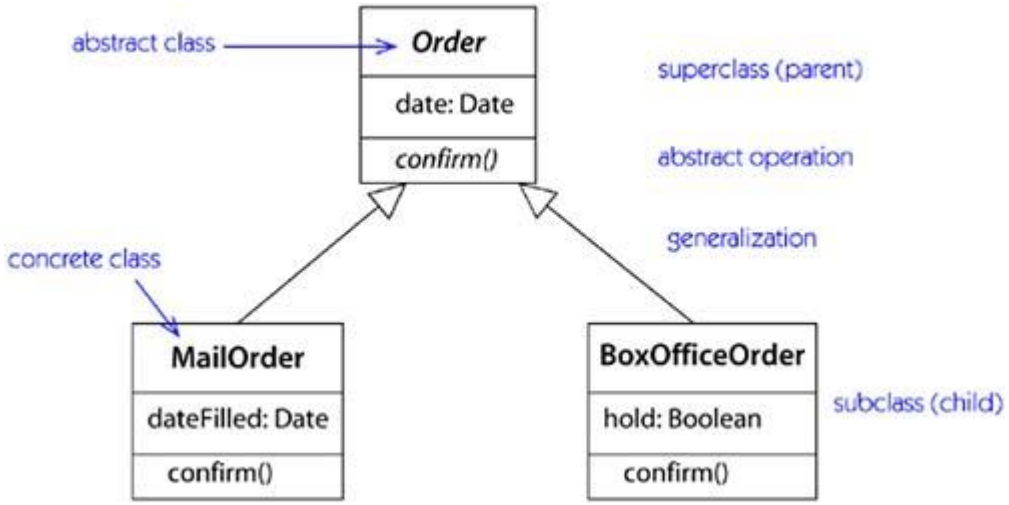
Relationships between classes

- **generalization**: an inheritance relationship
  - inheritance between classes
  - interface implementation

- **association**: a usage relationship
  - dependency
  - aggregation
  - composition

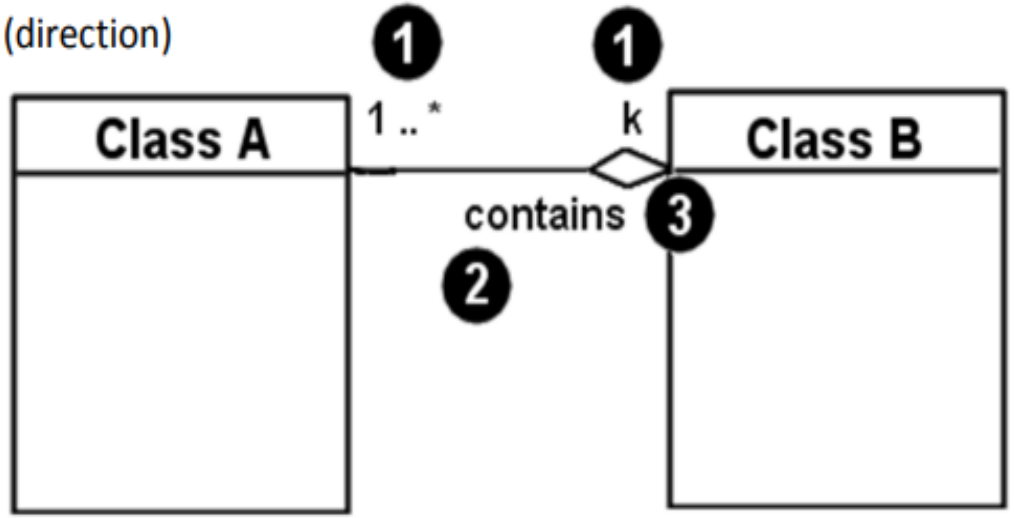
1-Generalization (inheritance)relationship

In UML modeling, a generalization relationship is **a relationship that implements the concept of object orientation called inheritance**. The generalization relationship occurs between two entities or objects, such that one entity is the parent, and the other one is the child.



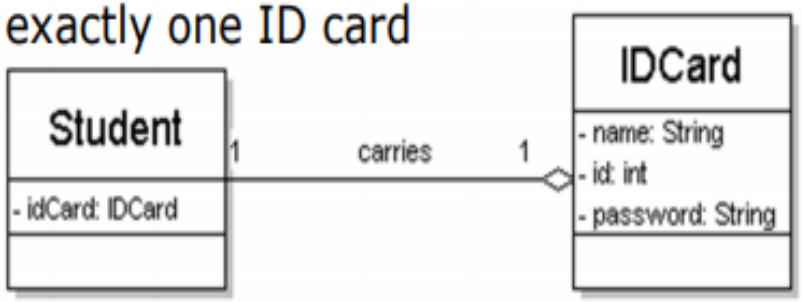
2- Associational (usage) relationships

- associational (usage) relationships
  1. multiplicity (how many are used)
     - * ⇒ 0, 1, or more
     - 1 ⇒ 1 exactly
     - 2..4 ⇒ between 2 and 4, inclusive
     - 3..* ⇒ 3 or more (also written as "3..")
  2. name (what relationship the objects have)
  3. navigability (direction)

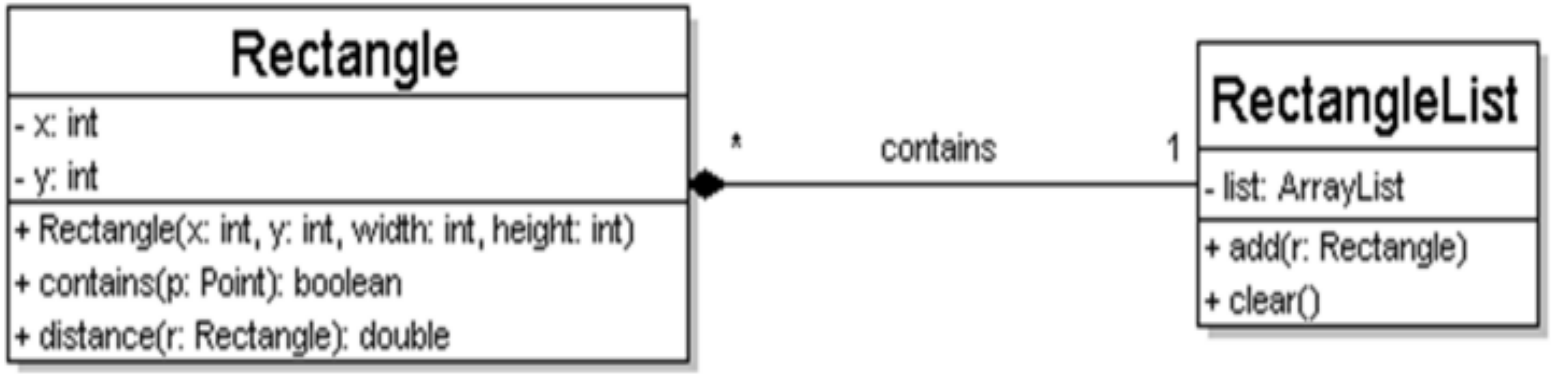


Association relations

- one-to-one
  - each student must carry exactly one ID card

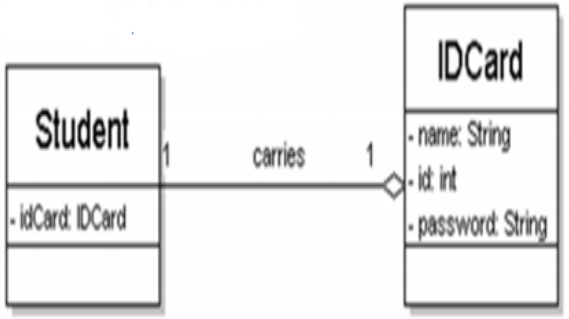


- one-to-many
  - one rectangle list can contain many rectangles

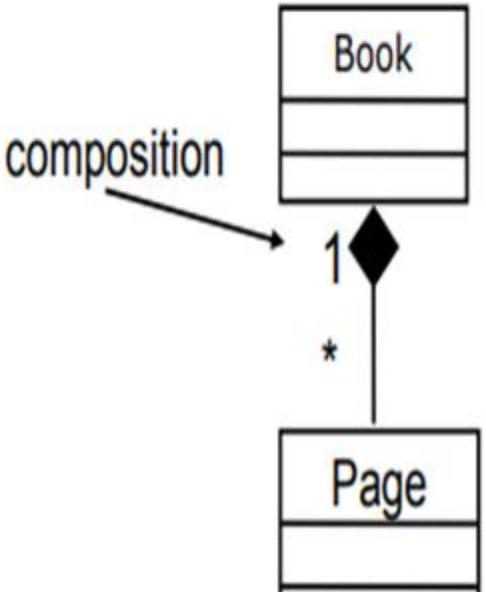


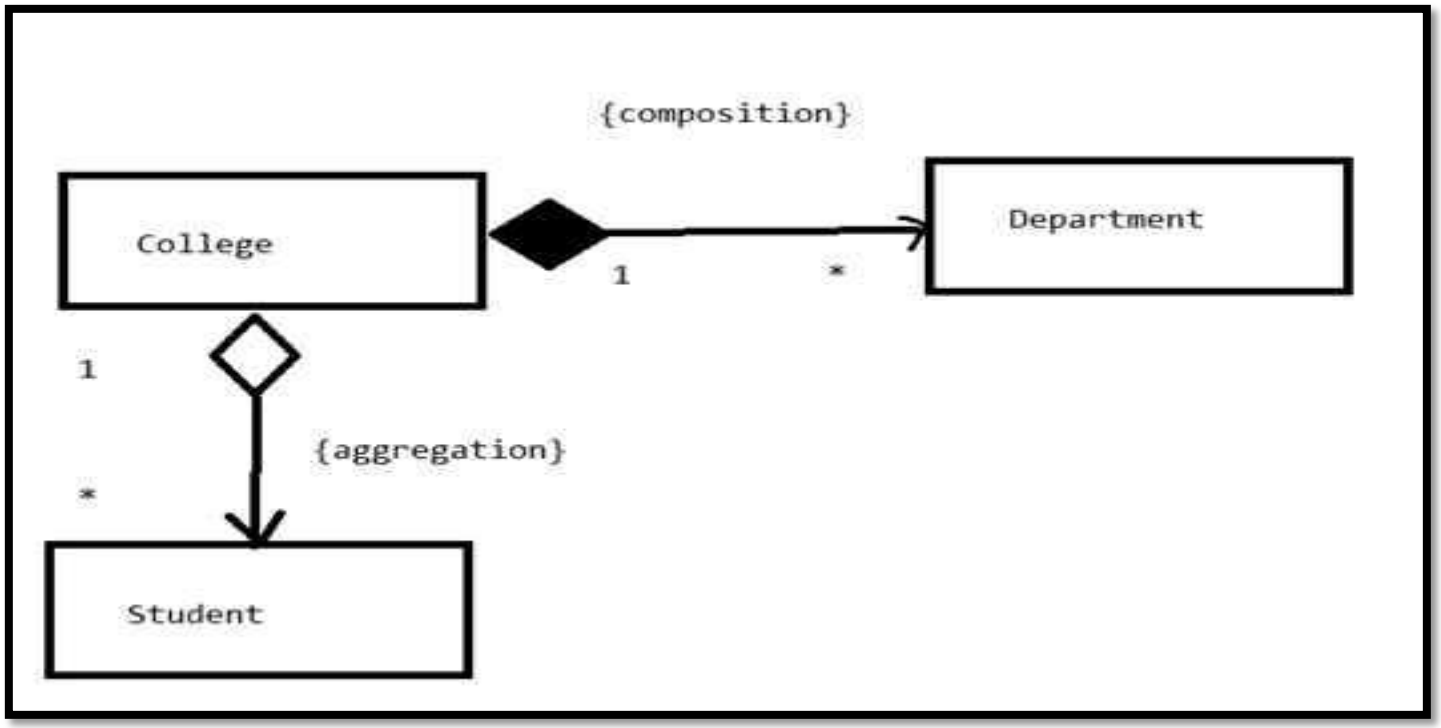
Association types

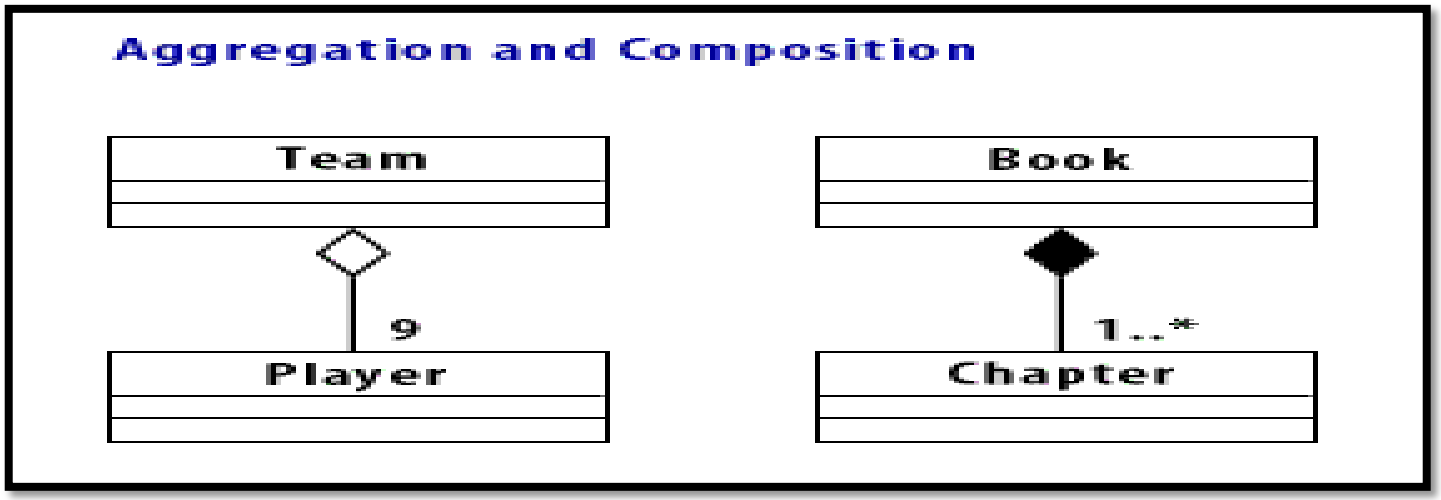
- **aggregation**: "is part of"
  - symbolized by a clear white diamond



- **composition**: "is entirely made of"
  - stronger version of aggregation
  - the parts live and die with the whole
  - symbolized by a black diamond




{composition}
College
Department
1
*
1
{aggregation}
*
Student


Aggregation and Composition
Team
9
Player
Book
1..*
Chapter

Car

Boat

Wheel

Engine

Propeller

Crankshaft

Piston

Class diagram example