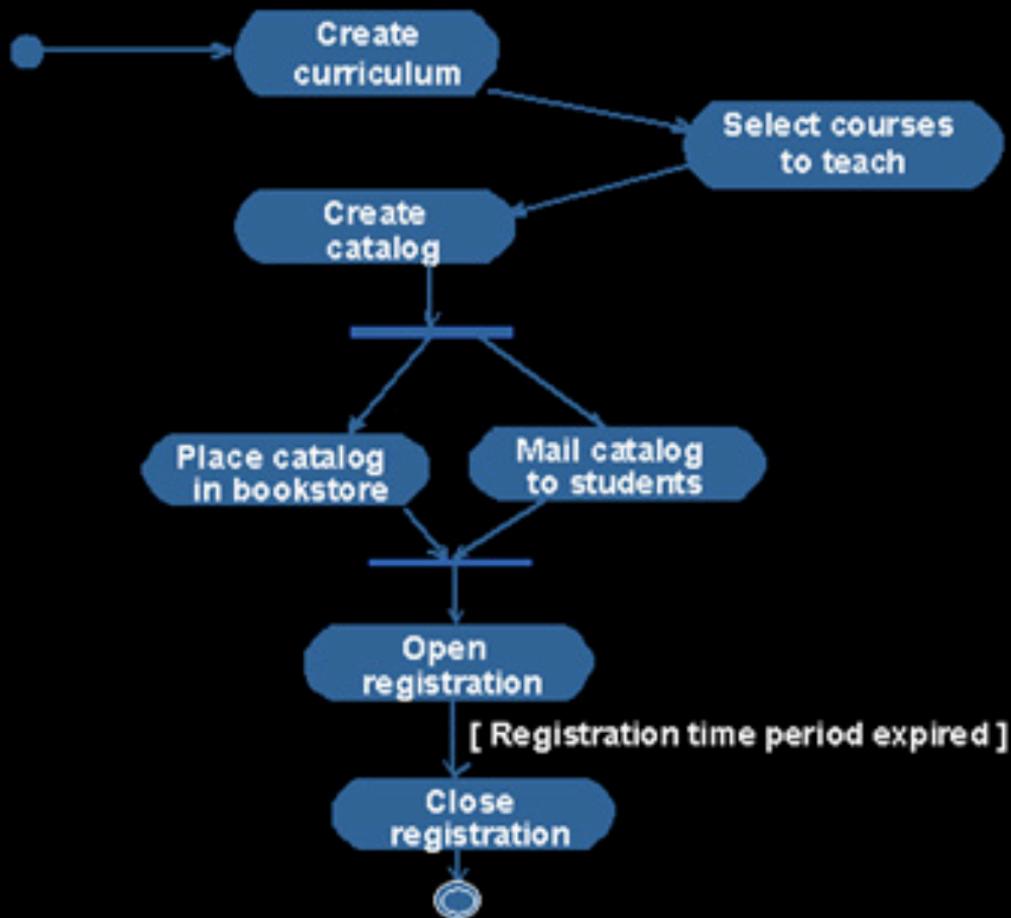


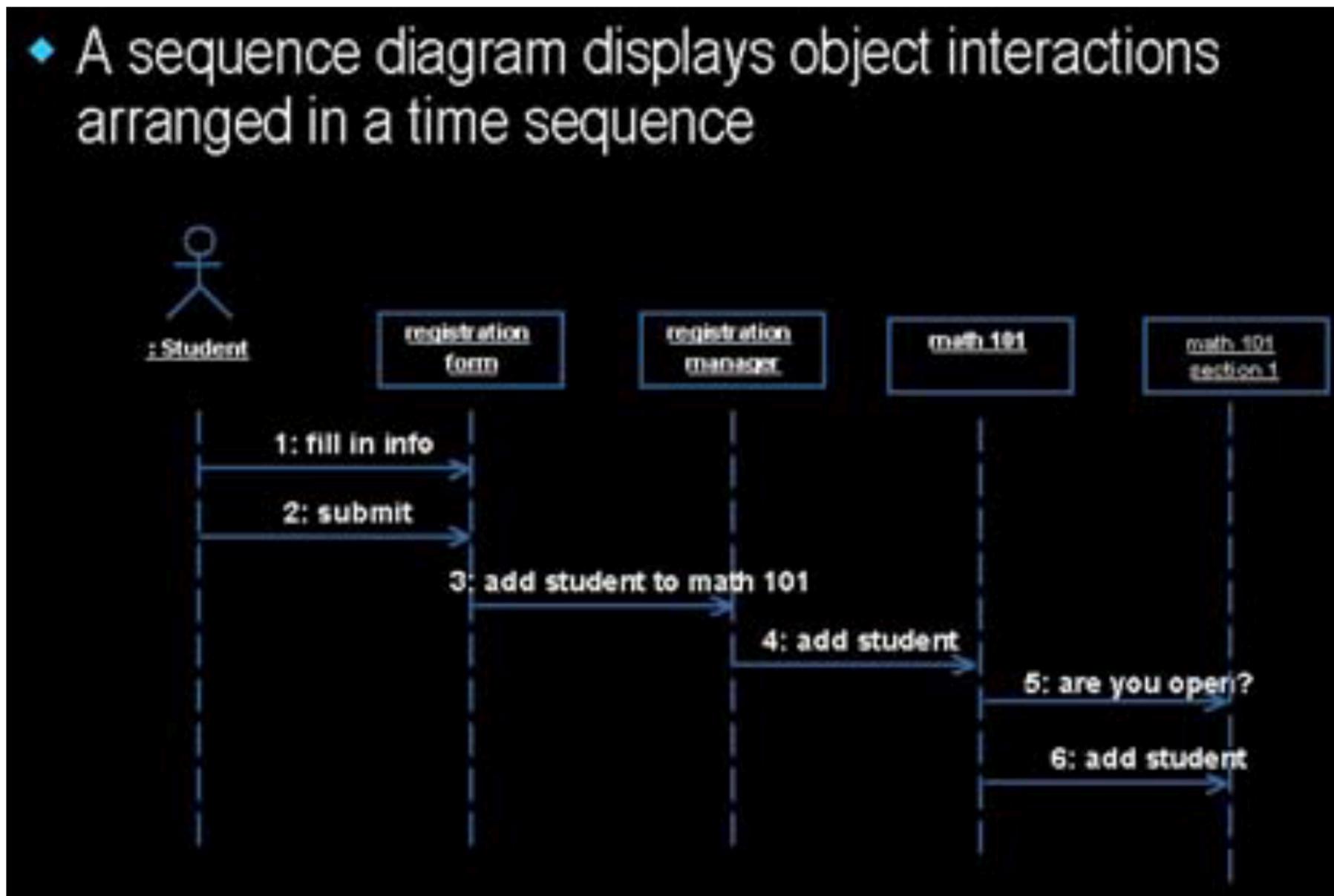
Activity Diagrams

- ◆ Activity diagrams show flow of control



Sequence Diagrams

- ◆ A sequence diagram displays object interactions arranged in a time sequence

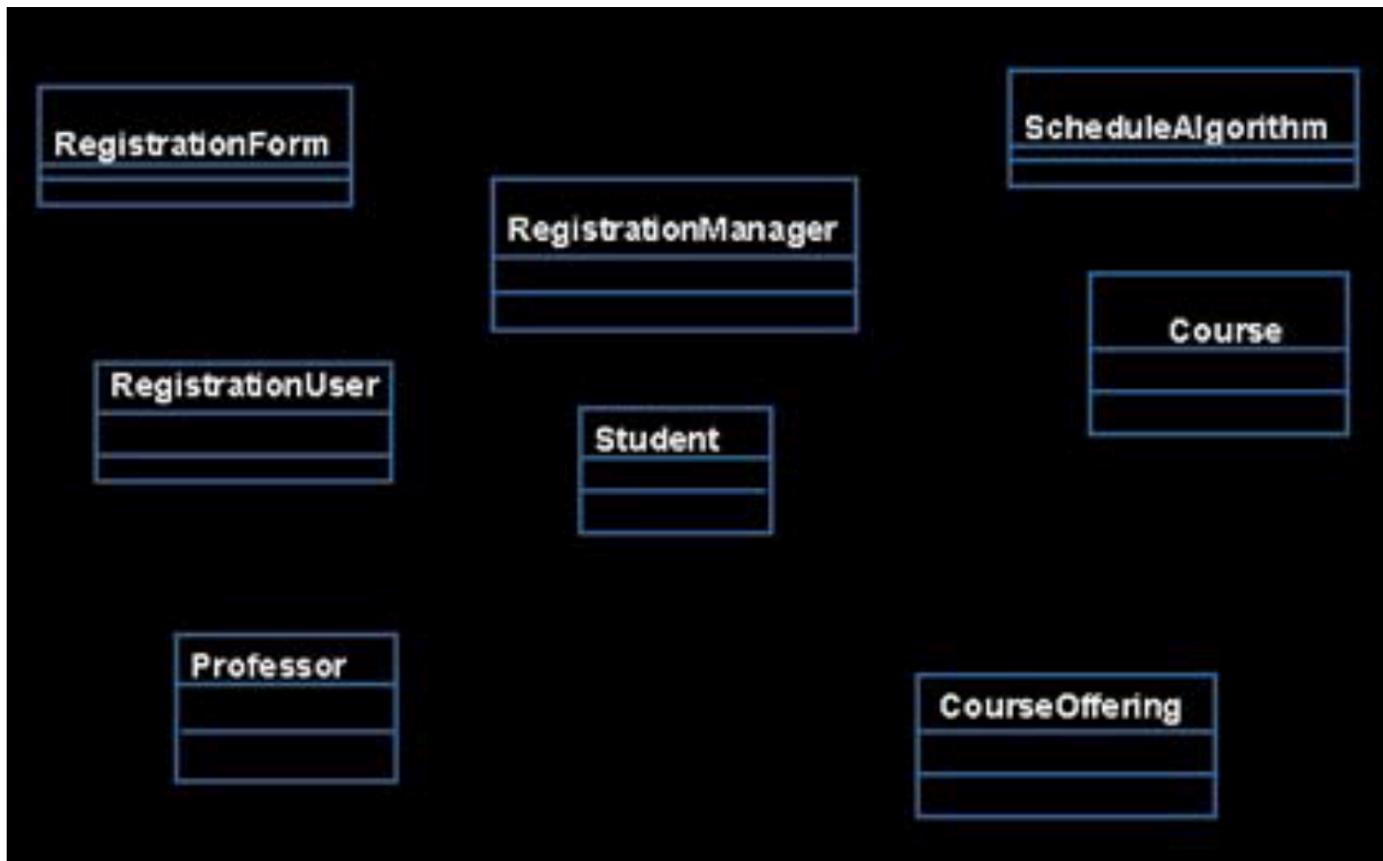


Class Diagrams

The UML modeling elements found in class diagrams include:

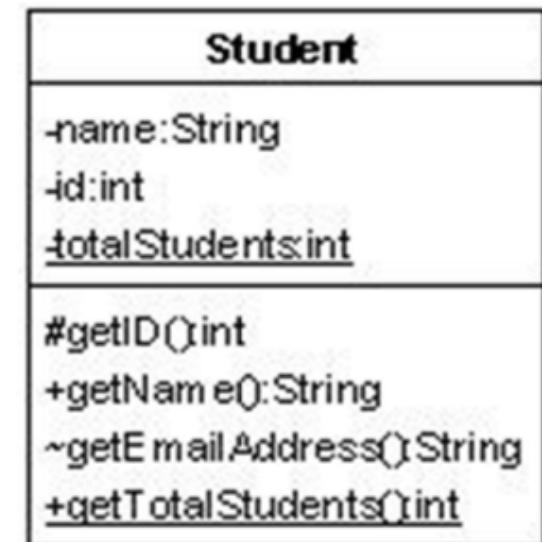
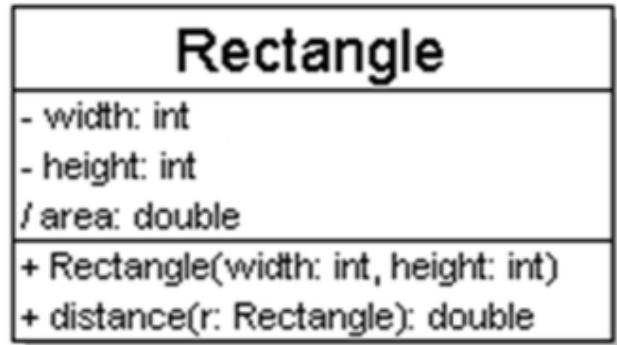
- Classes and their structure and behavior.
- Association, aggregation, dependency, and inheritance relationships.
- Multiplicity and navigation indicators
- Role names.

Class Diagrams

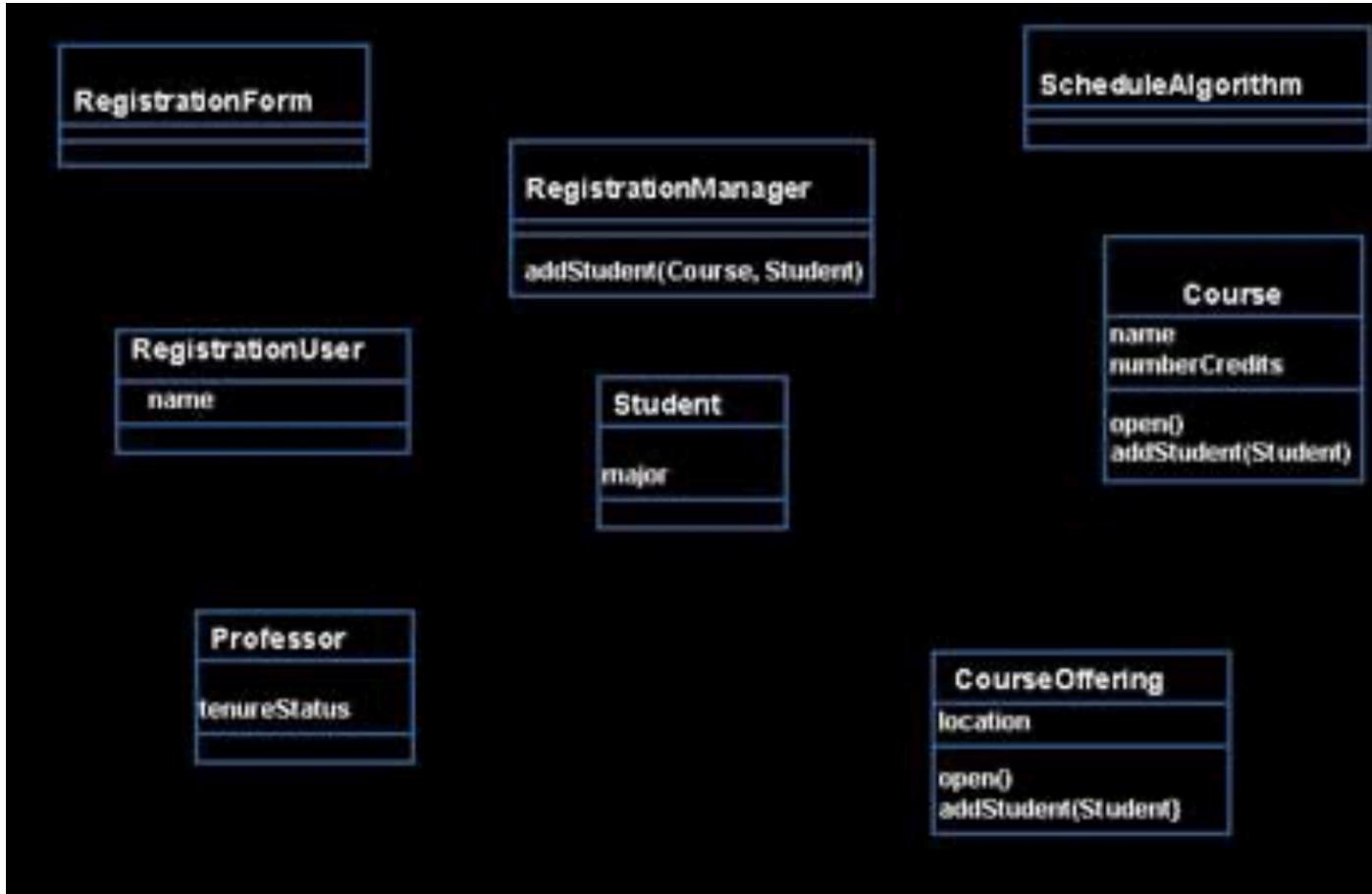


Class Diagrams

- Class name in top of box
 - write <>interface<> on top of interfaces' names
 - use italics for an abstract class name
- Attributes (optional)
should include all fields of the object
- Operations / methods (optional)
 - may omit trivial (get/set) methods
 - but don't omit any methods from an interface!
 - should not include inherited methods



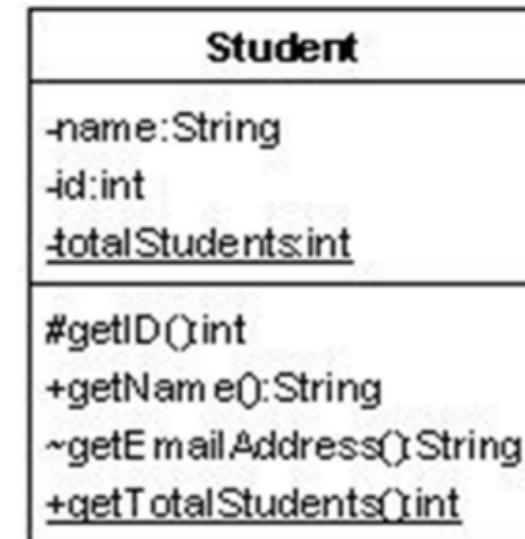
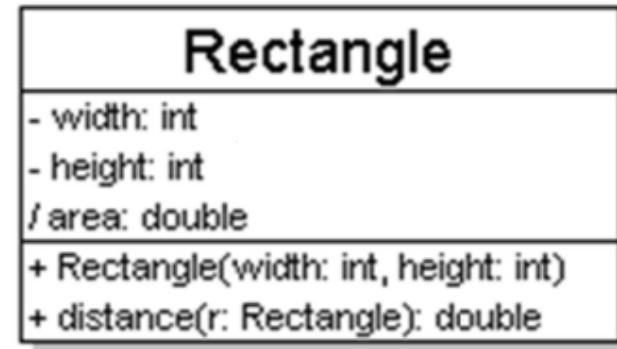
Class Diagrams -- attributes



Class Attributes

attributes (fields, instance variables)

- visibility name : *type [count]* = *default_value*
- visibility:
 - + public
 - # protected
 - - private
 - ~ package (default)
 - / derived
- underline static attributes
- **derived attribute:** not stored, but can be computed from other attribute values
- attribute example:
- balance : double = 0.00



Class Methods

Operation/ Method

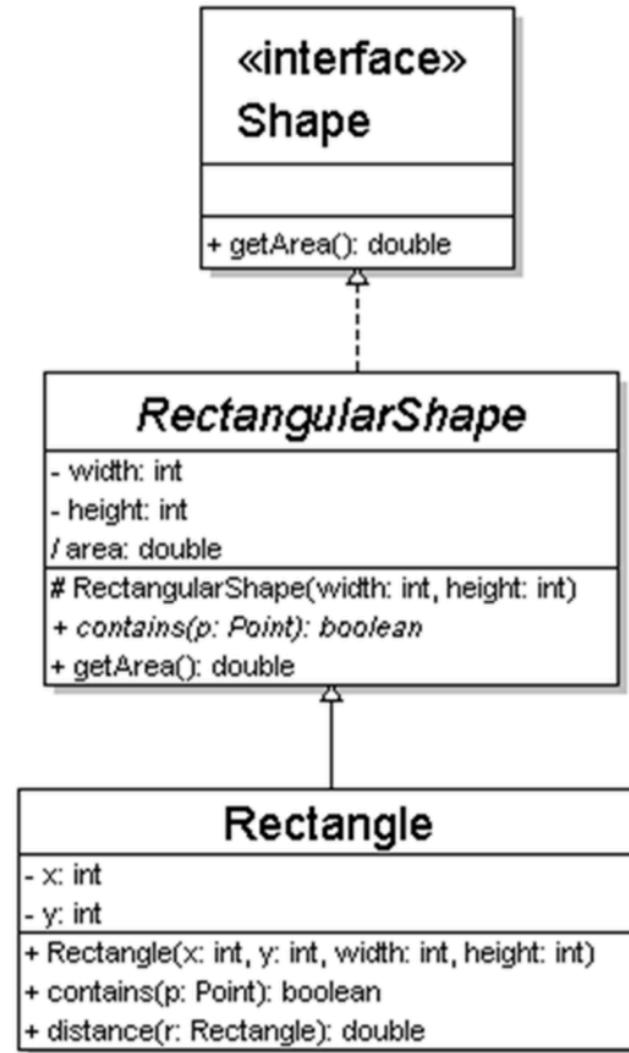
- visibility name [*parameter*] : *return type*
- visibility:
 - + public
 - # protected
 - - private
 - ~ package (default)
 - / derived
- underline static methods
- parameter types listed as (name: type)
- omit return_type on constructors and when return type is void
- Method example:
+ distance(p1: Point, p2: Point): double

Relationship between Classes

- **generalization:** an inheritance relationship
 - inheritance between classes
 - interface implementation
- **association:** a usage relationship
 - dependency
 - aggregation
 - composition

Generalization (inheritance) relationships

- hierarchies drawn top-down
- arrows point upward to parent
- line/arrow styles indicate whether parent is a(n):
 - class:
solid line, black arrow
 - abstract class:
solid line, white arrow
 - interface:
dashed line, white arrow
- often omit trivial / obvious generalization relationships, such as drawing the Object class as a parent



Associational relationships

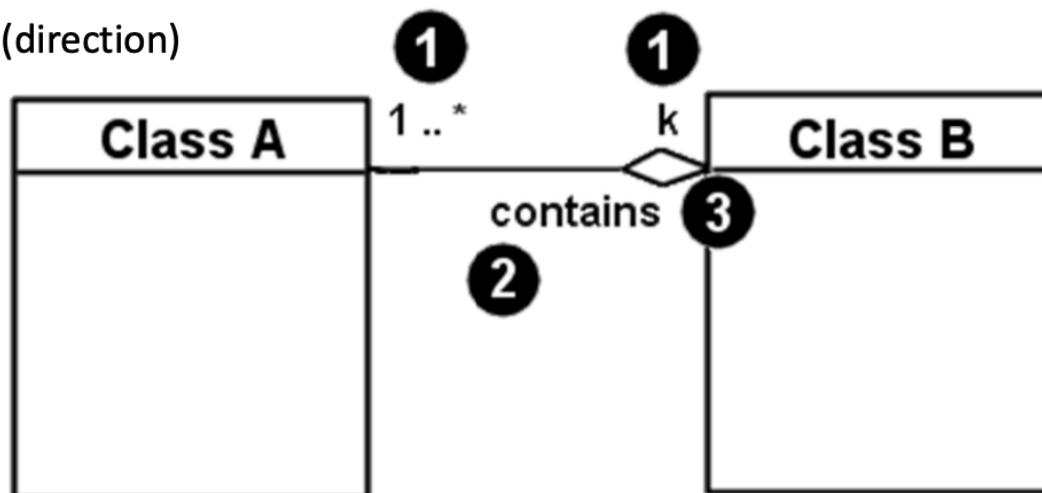
- associational (usage) relationships

1. multiplicity (how many are used)

- * \Rightarrow 0, 1, or more
- 1 \Rightarrow 1 exactly
- 2..4 \Rightarrow between 2 and 4, inclusive
- 3..* \Rightarrow 3 or more (also written as “3..”)

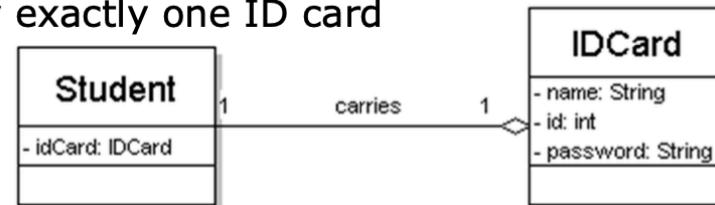
2. name (what relationship the objects have)

3. navigability (direction)

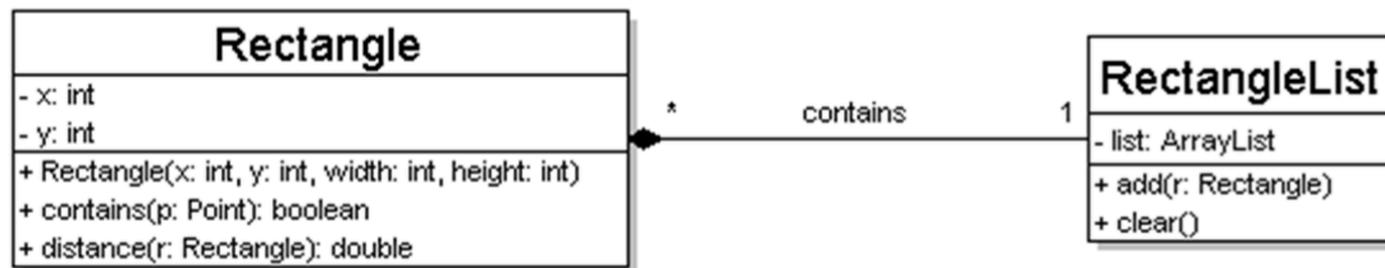


Multiplicity of associations

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

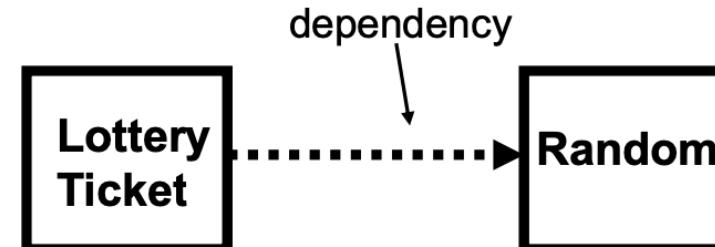
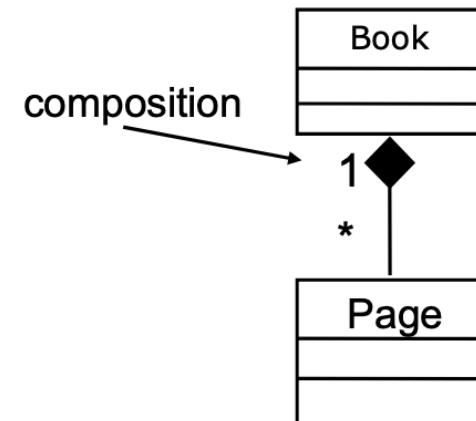
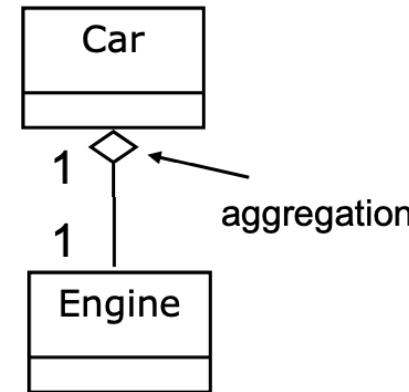


- one-to-many
 - one rectangle list can contain many rectangles

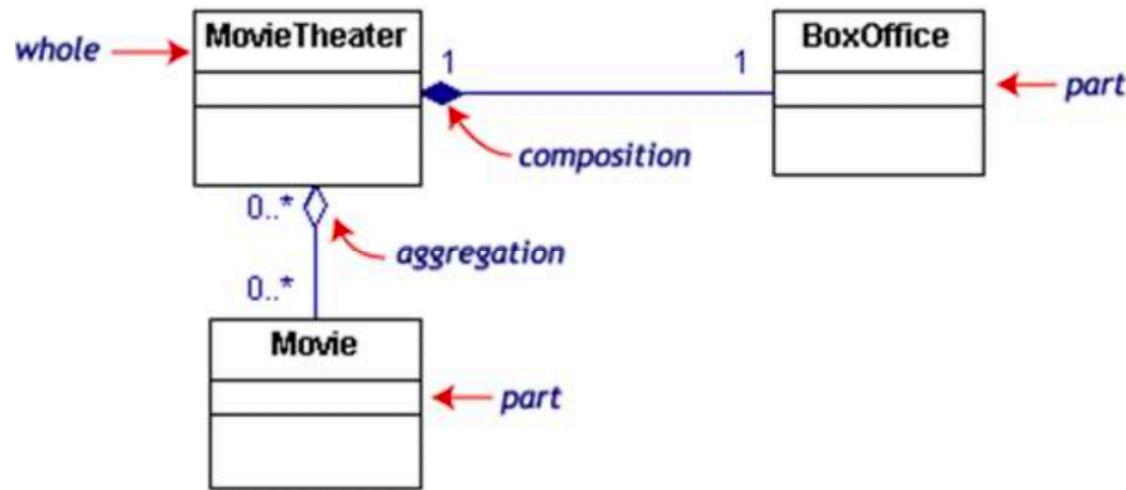


Association types

- **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
- **dependency:** “uses temporarily”
 - symbolized by dotted line
 - often is an implementation detail, not an intrinsic part of that object's state

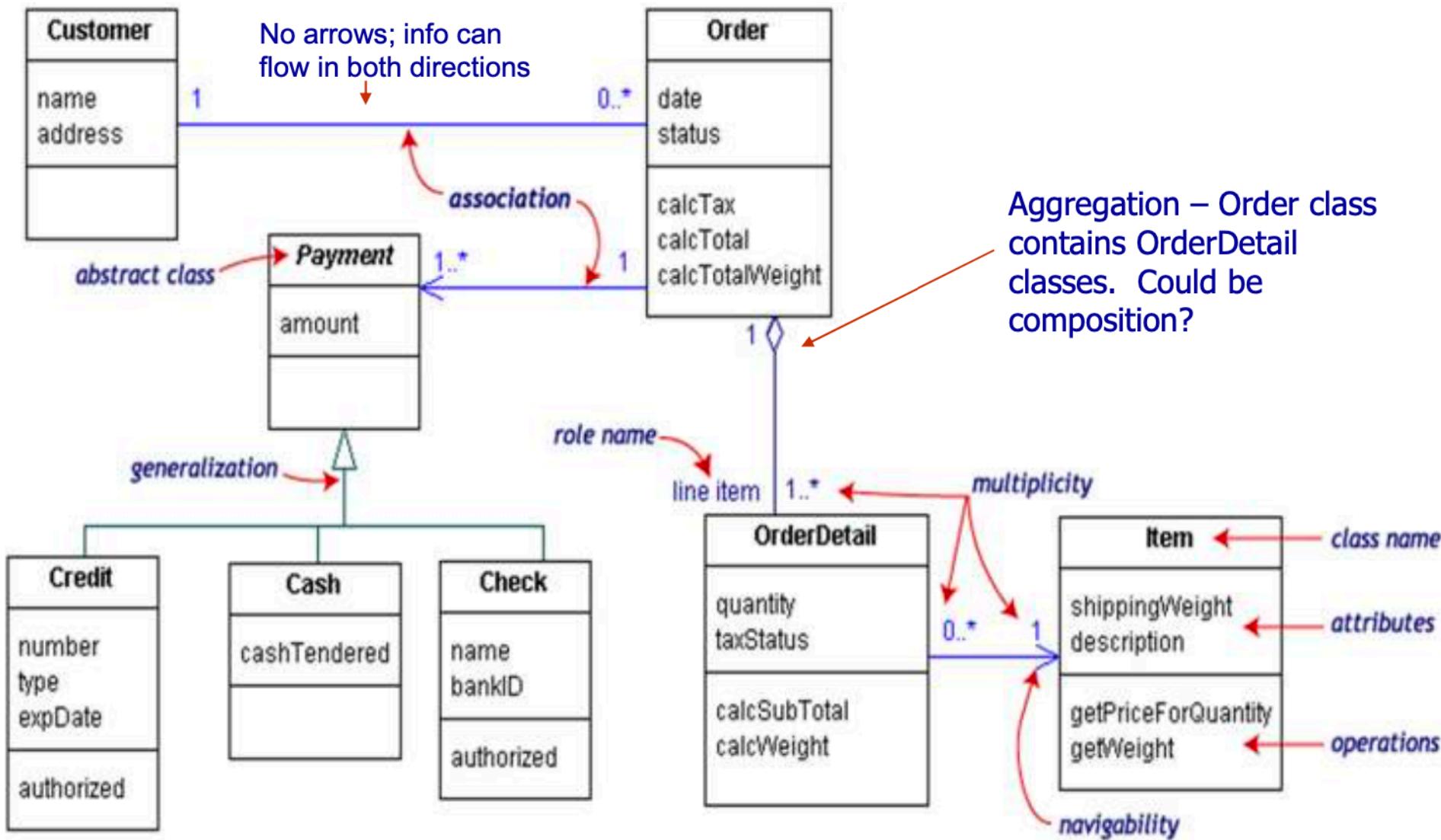


Aggregation Example

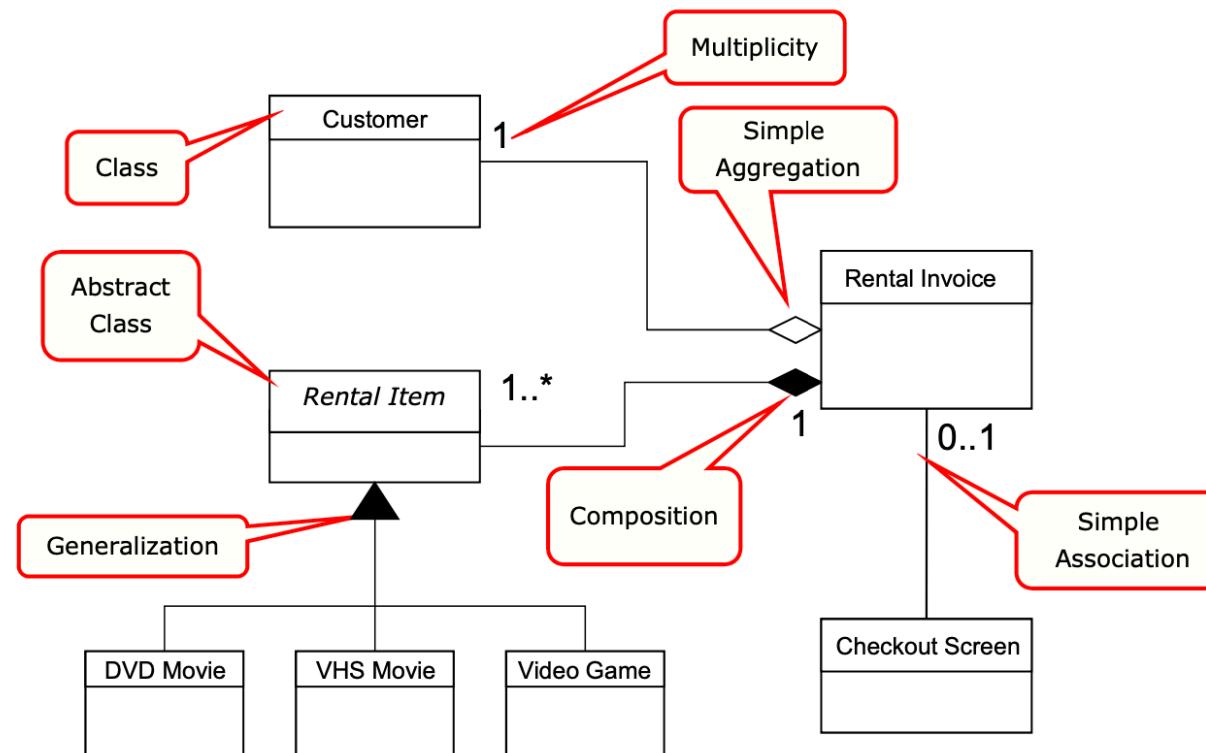


If the movie theater goes away
so does the box office => composition
but movies may still exist => aggregation

Class Diagram Example



Video Store Example



Example

- A professor has a name, address, phone number, email address, and salary. A student has also a name, etc., but no salary (sorry). A student, however, has an average mark (of the final marks of his or her seminars). A seminar has a name and a number. When a student is enrolled in a seminar, the marks for this enrollment are recorded and the current average as well as the final mark (if there is one) can be obtained from the enrollment. From a student one can obtain a list of seminars he or she is enrolled in. Professors teach seminars. Each seminar has at least one and at most three teachers. There are two types of seminar: bachelor and master. From a bachelor seminar students can not withdraw. From a master seminar they can.

Online Shopping

- Each customer has unique id and is linked to exactly one **account**. Account owns shopping cart and orders. Customer could register as a web user to be able to buy items online. Customer is not required to be a web user because purchases could also be made by phone or by ordering from catalogues.
- Web user has login name which also serves as unique id. Web user could be in several states - new, active, temporary blocked, or banned, and be linked to a **shopping cart**. Shopping cart belongs to account.
- Account owns customer orders. Customer may have no orders. Customer orders are sorted and unique. Each order could refer to several **payments**, possibly none. Every payment has unique id and is related to exactly one account.
- Each order has current order status. Both order and shopping cart have **line items** linked to a specific product. Each line item is related to exactly one product. A product could be associated to many line items or no item at all.

Hospital Management System

- *Person* could be associated with different *Hospitals*, and a Hospital could employ or serve multiple Persons.
- Hospitals have departments that have staff that can be broken down into operations (Doctors, nurses), administrative(receptionist), and technical (technologists and technicians) .