Object oriented analysis and design
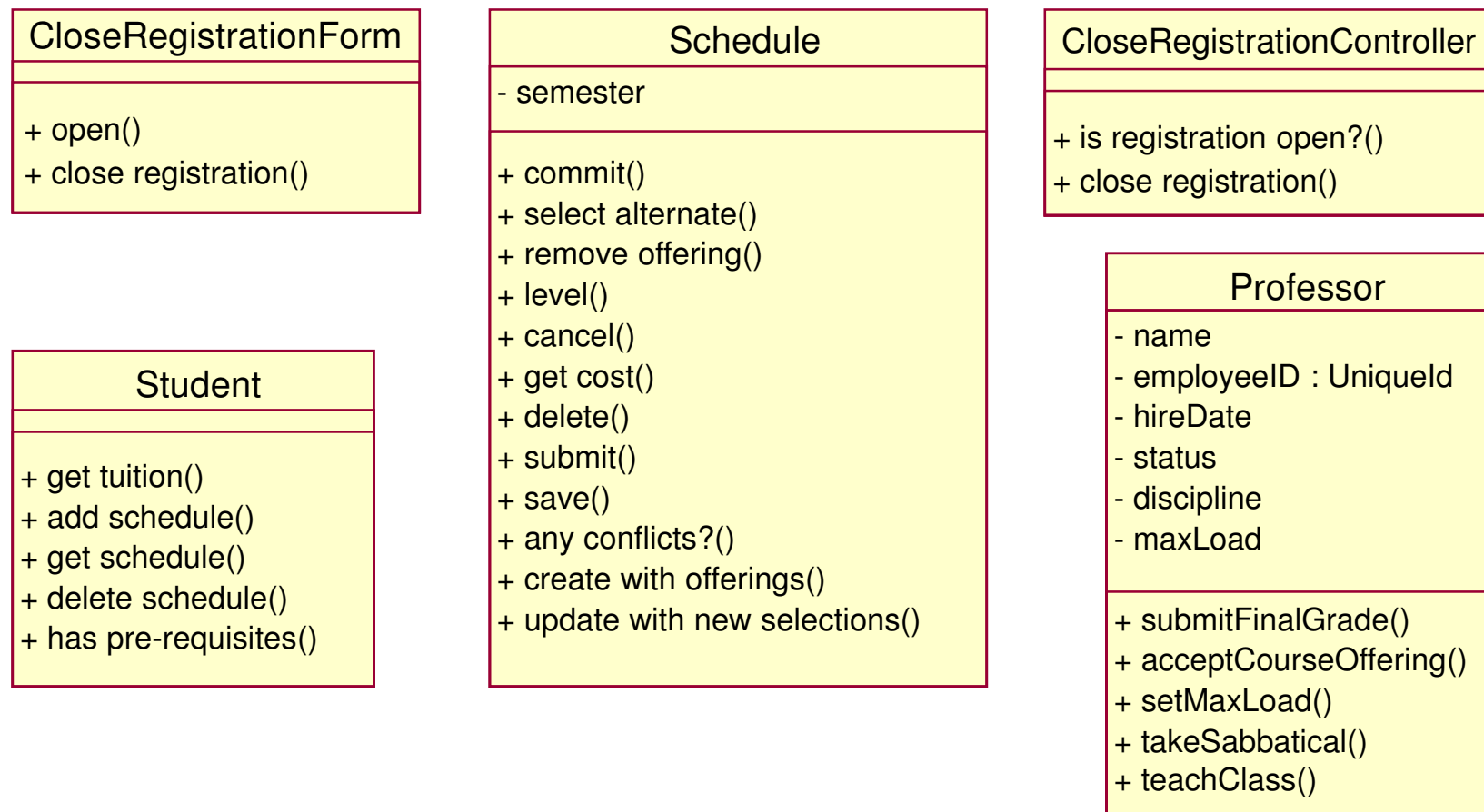
Module 5: Class Diagrams

Objectives

- Describe the static view of the system and show how to capture it in a model.

- Demonstrate how to read and interpret a class diagram.

- Model an association and aggregation and show how to model it in a class diagram.

- Model generalization on a class diagram.

Where Are We?

- **Class diagrams**
- Class relationships
  - Association
  - Aggregation
  - Generalization

What Is a Class Diagram?

- Static view of a system

CloseRegistrationForm
+ open() + close registration()

Schedule
- semester
+ commit() + select alternate() + remove offering() + level() + cancel() + get cost() + delete() + submit() + save() + any conflicts?() + create with offerings() + update with new selections()

CloseRegistrationController
+ is registration open?() + close registration()

Student
+ get tuition() + add schedule() + get schedule() + delete schedule() + has pre-requisites()

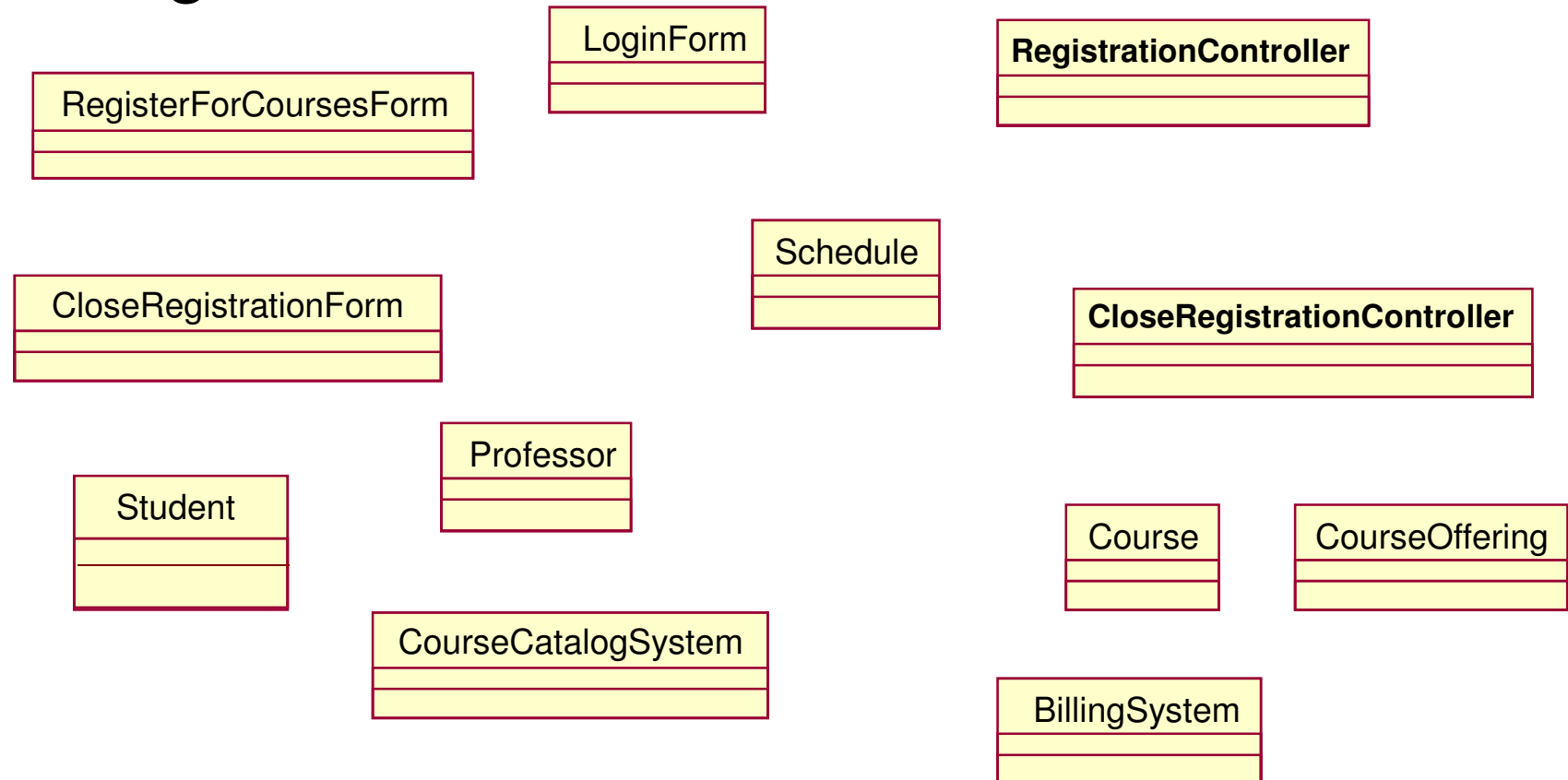| Professor |
|---|
| - name
- employeeID : UniqueId
- hireDate
- status
- discipline
- maxLoad |
| + submitFinalGrade()
+ acceptCourseOffering()
+ setMaxLoad()
+ takeSabbatical()
+ teachClass() |

Class Diagram Usage

- When modeling the static view of a system, class diagrams are typically used in one of three ways, to model:
  - The vocabulary of a system
  - Collaborations
  - A logical database schema

Example: Class Diagram

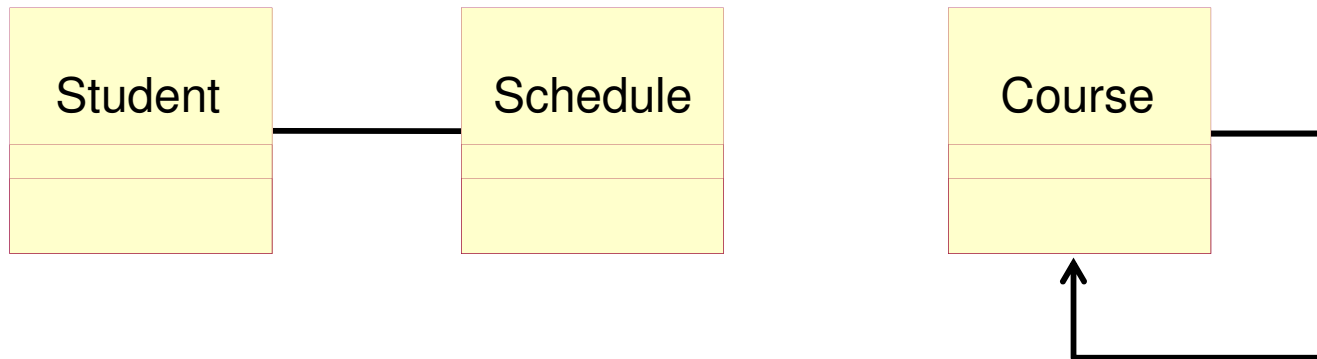- Is there a better way to organize class diagrams?

LoginForm

RegistrationController

RegisterForCoursesForm

Schedule

CloseRegistrationForm

CloseRegistrationController

Professor

Student

Course

CourseOffering

CourseCatalogSystem

BillingSystem

Where Are We?

- Class diagrams
- **Class relationships**
  - Association
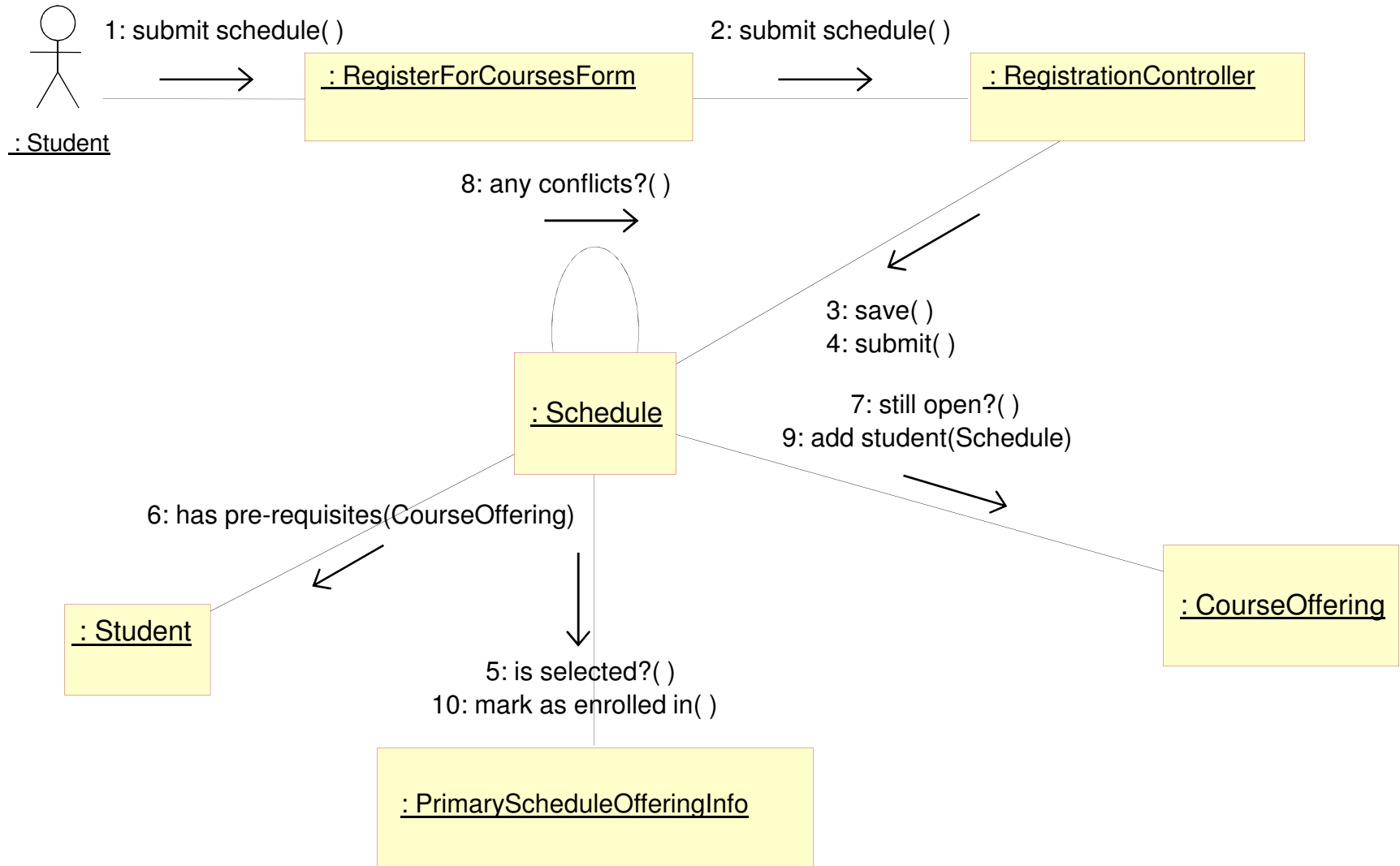 - Aggregation
  - Generalization

What Is an Association?

- The semantic relationship between two or more classifiers that specifies connections among their instances.

- A structural relationship specifying that objects of one thing are connected to objects of another thing.

| Student | | Schedule | | Course |

Example: What Associations Can You Find?

: Student

1: submit schedule( )

: RegisterForCoursesForm

2: submit schedule( )

: RegistrationController

8: any conflicts?( )

: Schedule

3: save( )
4: submit( )

7: still open?( )
9: add student(Schedule)

6: has pre-requisites(CourseOffering)

: Student

: CourseOffering

5: is selected?( )
10: mark as enrolled in( )

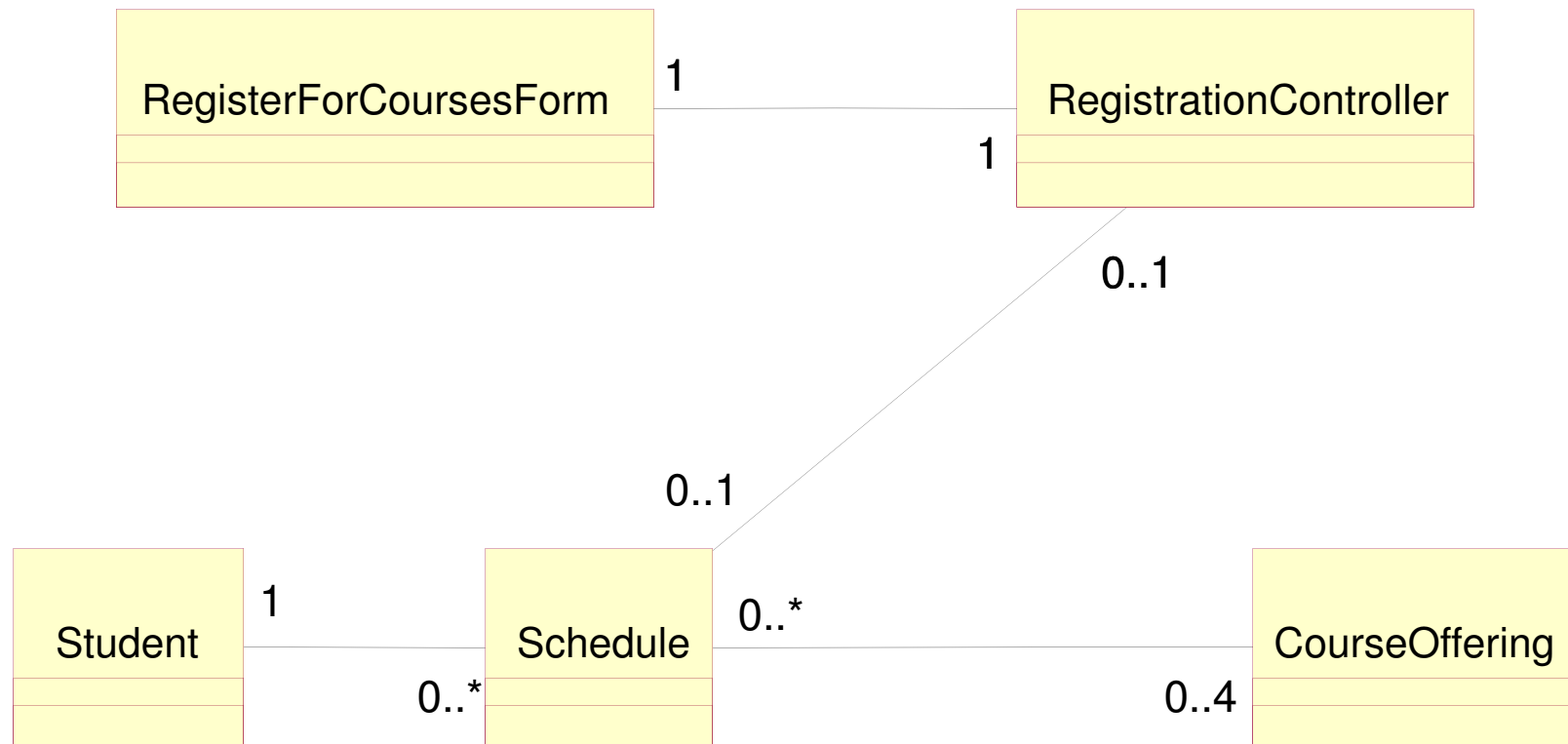: PrimaryScheduleOfferingInfo

What Is Multiplicity?

- Multiplicity is the number of instances one class relates to ONE instance of another class.

- For each association, there are two multiplicity decisions to make, one for each end of the association.

  - For each instance of Professor, many Course Offerings may be taught.

  - For each instance of Course Offering, there may be either one or zero Professor as the instructor.

Professor	instructor	CourseOffering
	0..1 0..*	
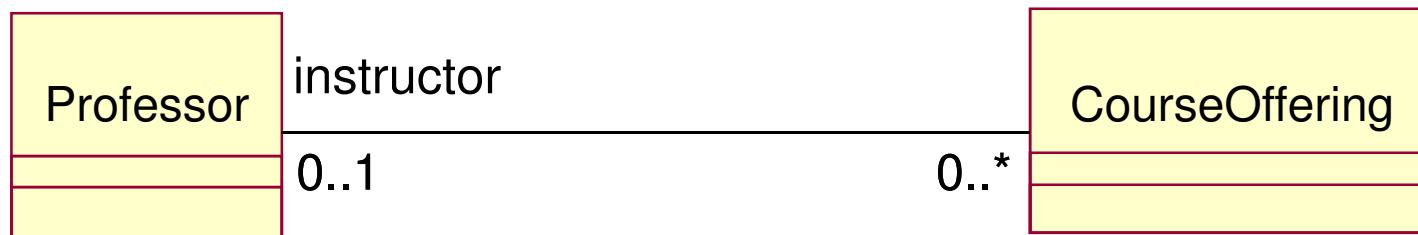
Multiplicity Indicators

Unspecified	
Exactly One	1
Zero or More	0..*
Zero or More	*
One or More	1..*
Zero or One (optional value)	0..1
Specified Range	2..4
Multiple, Disjoint Ranges	2, 4..6

Example: Multiplicity

RegisterForCoursesForm — 1 ——— 1 — RegistrationController

0..1

0..1

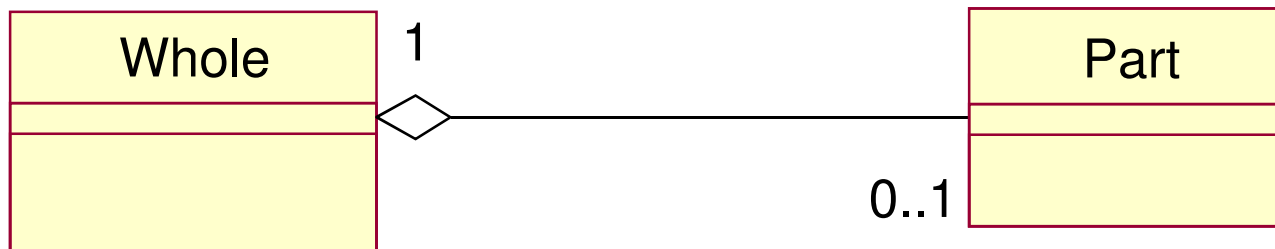Student — 1 ——— 0..* — Schedule — 0..* ——— 0..4 — CourseOffering
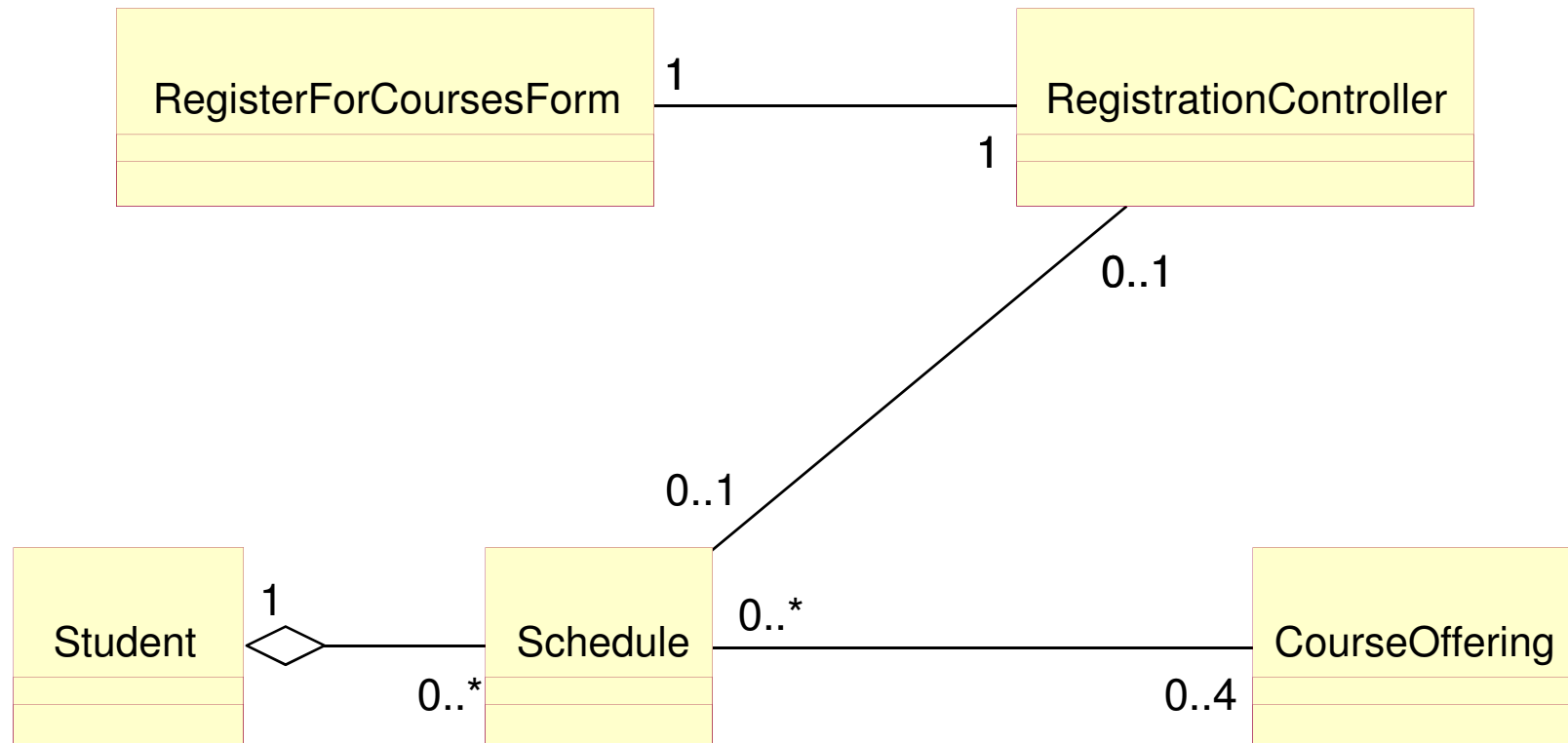
Where Are We?

- Class diagrams
- **Class relationships**
  - Association
  - Aggregation
  - Generalization

What Is an Aggregation?

- A special form of association that models a whole-part relationship between the aggregate (the whole) and its parts.
  - An aggregation is an "is a part-of" relationship.
- Multiplicity is represented like other associations.

Example: Aggregation

Where Are We?

- Class diagrams
- Class relationships
    - Association
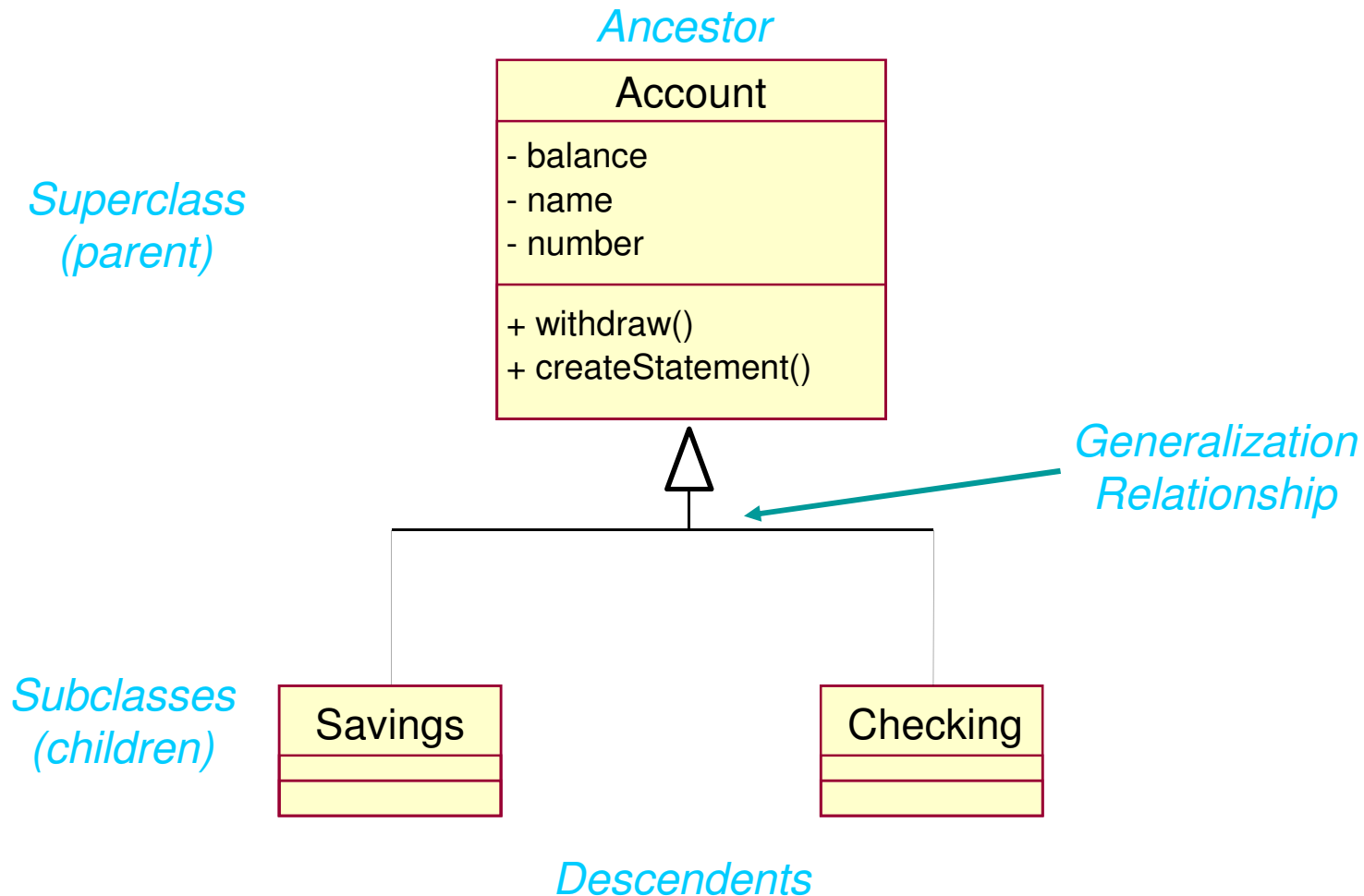   - Aggregation
    - Generalization

Review: What Is Generalization?

- A relationship among classes where one class shares the structure and/or behavior of one or more classes.

- Defines a hierarchy of abstractions where a subclass inherits from one or more superclasses.
  - Single inheritance
  - Multiple inheritance

- Is an "is a kind of" relationship.

Example: Single Inheritance

- One class inherits from another.

Ancestor

*Superclass
(parent)*

Account
- balance - name - number
+ withdraw() + createStatement()

*Generalization
Relationship*

*Subclasses
(children)*

Savings

Checking

Descendents

Review

- What does a class diagram represent?
- What benefits do packages provide to the model?
- Define association, aggregation, and generalization.
- How do you find associations?
- What is multiplicity? What information does multiplicity provide the modeler?