MAHARISHI UNIVERSITY of MANAGEMENT

Engaging the Managing Intelligence of Nature

Computer Science Department

CS401 Modern Programming Practices (MPP)

Professor Joe Bruen

Lecture 3: Associations

Modeling Relationships with UML

Wholeness Statement

In the real world, objects have relationships. These manifested relationships appear in many different ways. In this lecture, we explore the range of relationships that can be modeled in UML depending on how the objects' relate to each other.

At the most fundamental level every object is made out of the same essence – and is therefore (in a way) related to everything. An intellectual analysis or model of all these relationships is generally not practical.

A direct experience of the underlying reality of all of manifest creation and our relationship with all of nature is a result of our practice of Transcendental Meditation.

Association Analysis

Finding the relationships between concepts

Associations

- Define how classes interrelate to each other.
- Look at the problem statement after defining the data dictionary and identify verbs (verb phrases)
  - Chose structural relationship.
  - Ignore actions and behaviors (transient).
- Methods
  - Verb phrase analysis
  - Create association matrix

Associations between Classes

- Associations
- Aggregation / Composition
- Inheritance
- Use verb phrases in requirements to help identify associations, ignoring those that represent transient actions or behaviors

The Student Registration System

We have been asked to develop an automated Student Registration System (SRS) for the university. This system will enable students to register online for courses each semester, as well as track their progress toward completion of their degree.

When a student first enrolls at the university, he/she uses the SRS to set forth a plan of study as to which courses he/she plans on taking to satisfy a particular degree program, and chooses a faculty advisor. The SRS will verify whether or not the proposed plan of study satisfies the requirements of the degree that the student is seeking.

Once a plan of study has been established, then, during the registration period preceding each semester, students are able to view the schedule of classes online, and choose whichever classes they wish to attend, indicating the preferred section (day of the week and time of day) if the class is offered by more than one professor. The SRS will verify whether or not the student has satisfied the necessary prerequisites for each requested course by referring to the student's online transcript of courses completed and grades received (the student may review his/her transcript online at any time).

Assuming that (a) the prerequisites for the requested course(s) are satisfied, (b) the course(s) meet(s) one of the student's plan of study requirements, and (c) there is room available in each of the class(es), the student is enrolled in the class(es).

If (a) and (b) are satisfied, but (c) is not, the student is placed on a first-come, first-served wait list. If a class/section that he/she was previously waitlisted for becomes available (either because some other student has dropped the class or because the seating capacity for the class has been increased), the student is automatically enrolled in the waitlisted class, and an email message to that effect is sent to the student. It is the student's responsibility to drop the class if it is no longer desired; otherwise, he/she will be billed for the course.

Students may drop a class up to the end of the first week of the semester in which the class is being taught.

Association Matrix

	Section	Course	Plan of Study	Professor	Student	Transcript
Section						
Course						
Plan of Study						
Professor						
Student						
Transcript						

Associations– In class Exercise

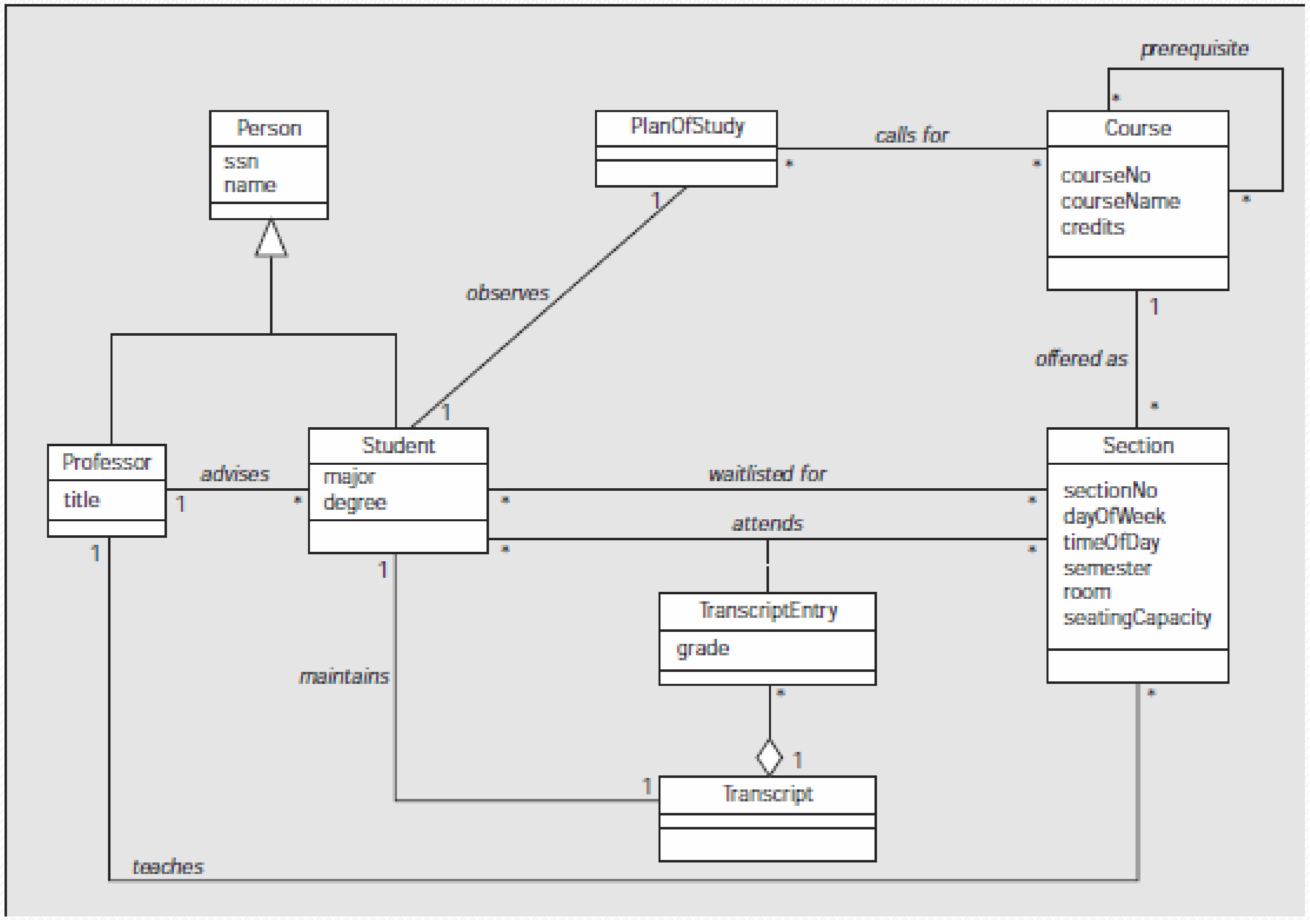
In your small groups refer to our problem description and fill in the Association Matrix for our classes.

	Section	Course	Plan of Study	Professor	Student	Transcript
Section		instance of		is taught by		included in
Course		prerequisite for	is called for by			
Plan of Study		calls for			observed by	
Professor	teaches				advises; teaches	
Student	registered for; waitlisted for; has previously taken	plans to take	observes	is advised by; studies under		owns
Transcript	Includes?	Includes?			belongs to	

Problem Description – In class Exercise

In your small group now try to create a diagram with all the classes and their labeled associations

Student Registration System



Main Point 1

The whole is greater than the sum of the parts.

It is not just about the different classes that make up a system, but also about how they relate to each other, what their associations are.

If we don’t understand how they relate it is easy to lose track of what we are doing, and why we are doing it. The interactions bring life into the system.

During concept analysis we decomposed the different parts, and now we re-unite them using associations.

Modeling Associations

Connecting the parts into a wholeness

Topics:

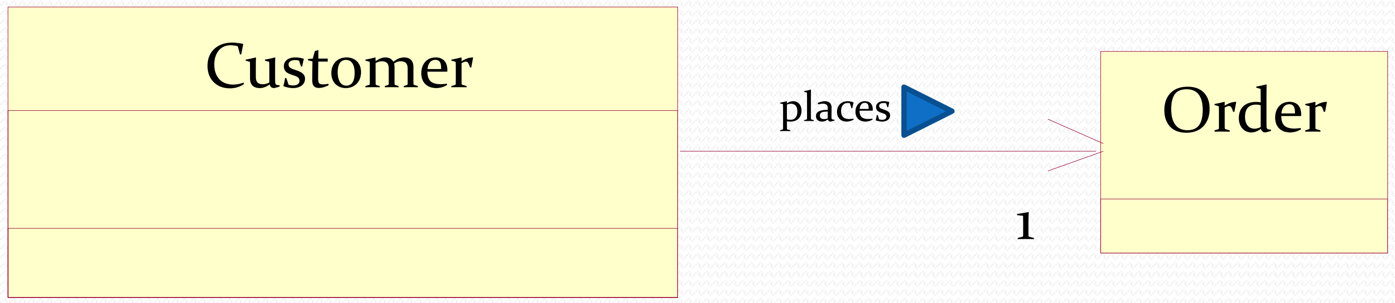
- Association
- Multiplicity
- Aggregation
- Composition
- Inheritance
- Navigation
- Roles
- Association Classes
- Reflexive Associations

Describing Relationships

- Relationships define how objects interact:
  - Student *takes* Sections
  - Professor *teaches* Sections
  - Client *pays* Vendor
  - Customer *places* Order
- Relationships are often two-way streets:
  - Student *takes* Section
  - Section *is taken by* Student

Association

- Unidirectional
  - Objects of a class have a reference to an object of another class.
  - Associations as attributes.
  - Name that describes the association



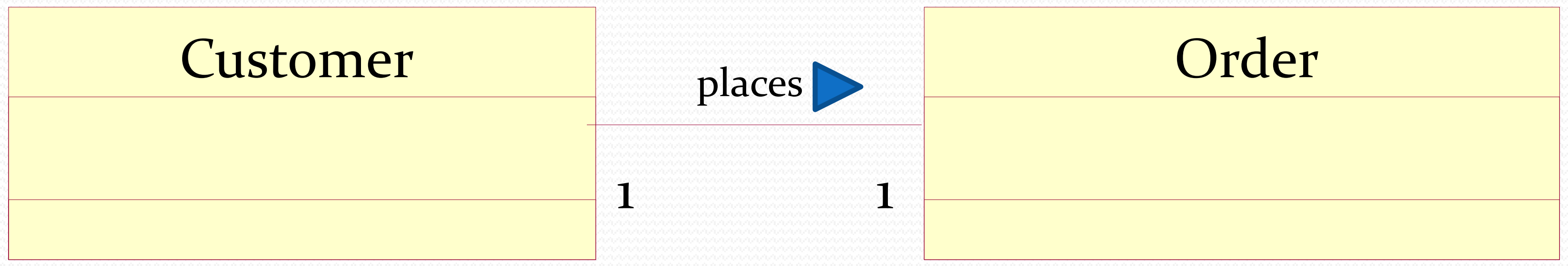
```
public class Customer {
    private Order order;
}
```

```
public class Order {

}
//does this look right
to you?
```

Association

- Bi-directional
  - Description direction is more important (still optional)

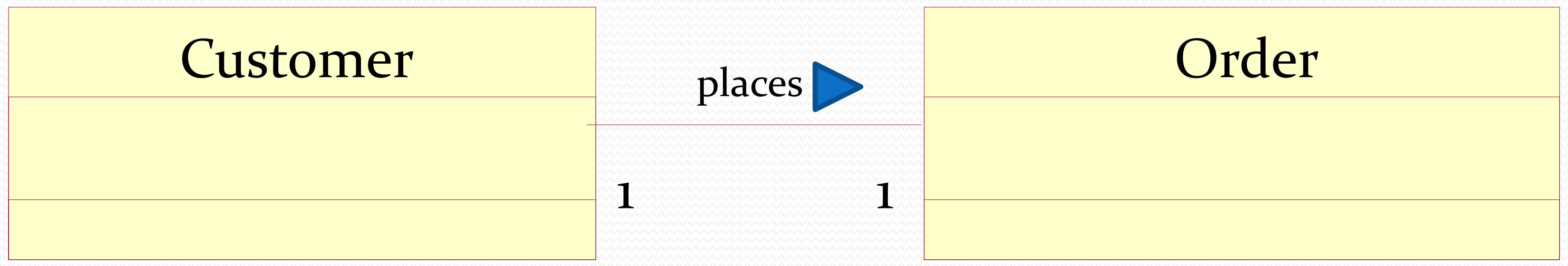


```
public class Customer {
    private Order order;
}
//now does this look right?
```

```
public class Order {
    private Customer customer;
}
```

Association

- A customer should be able to have more than one order outstanding .....



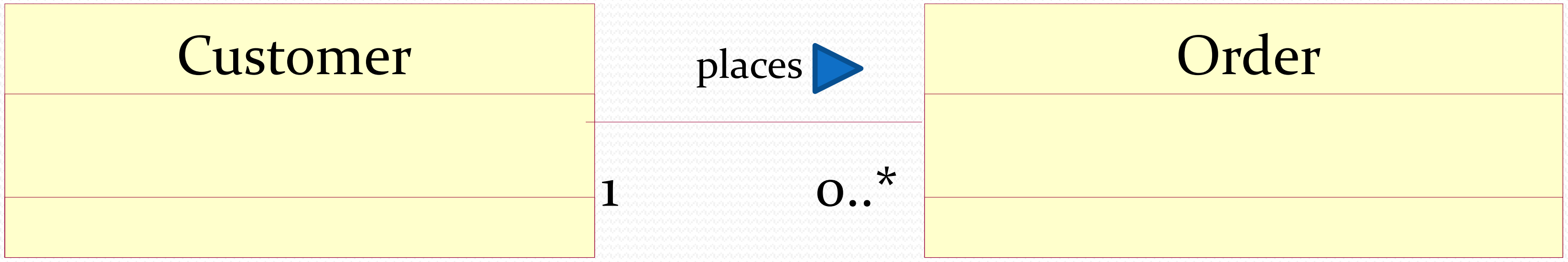
```
public class Customer {
    private Order order;
}
```

```
public class Order {
    private Customer customer;
}
```

Association

1	one (mandatory)
3	three (exactly)
*	many
0..*	zero or more (optional)
1..*	one or more
0..1	zero or one (optional)

- Multiplicity



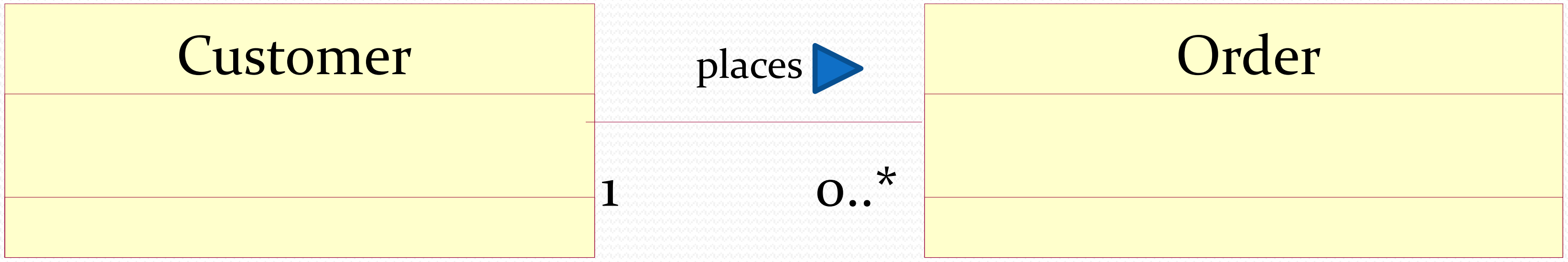
```
public class Customer {
    //private Order order;
    //what will our code for
    // multiple order be?
}
```

```
public class Order {
    private Customer customer;
}
```

Association

1	one (mandatory)
3	three (exactly)
*	many
0..*	zero or more (optional)
1..*	one or more
0..1	zero or one (optional)

- Multiplicity



```
public class Customer {
    private List<Order> orders;
}
```

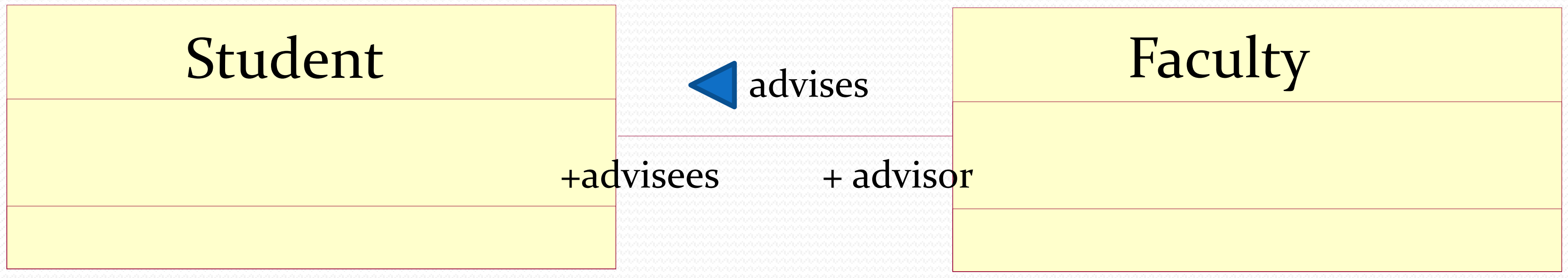
```
public class Order {
    private Customer customer;
}
```

Multiplicity

- How many instances of each object must/may be associated with the other?
- Optionality
  - Is the association required?
- Cardinality
  - How many associations are needed?
- UML combines both ideas in the Multiplicity concept.

Association roles

- Add a description to one, or both sides of the association to indicate the role(s) each object plays in the relationship.



```
public class Student {
    private Faculty advisor;
}
```

```
public class Faculty {
    private List<Student> advisees;
}
```

Main Point 2

Associations model the relationships that can exist between concepts. Simple associations are modeled with a line.

The line should have a name for ease of reading, and additional symbols for directionality and multiplicity.

The end of a line can also specify an association role, which is a different name for the connecting concept that is used in the context of this relationship.

The simplest state of awareness can also be modeled with a straight line.

Mid-Term Prep Exercise (3-1)

For the following problem statement work independently for 10 minutes to create the class diagram. Try to show the associations you need.

Then review your diagrams for 10 minutes with your small group.

In Class Mid-Term Practice

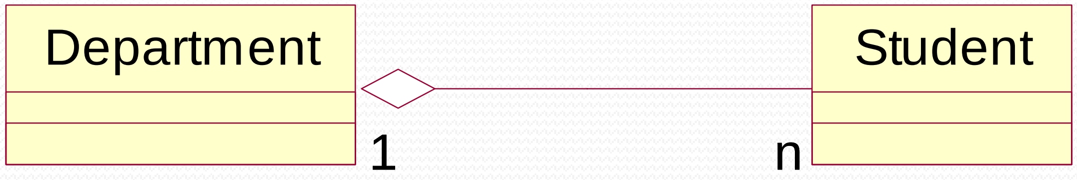
A Human Resource (HR) department keeps track of employees for several companies. Each company has a name and may consist of many departments. A department has a name and a location. Each department has one or more positions. A position may be vacant. Otherwise, an employee is assigned to it. The HR personnel enter an employee's first name, middle initial, last name, birth date and Social Security Number (SSN). An employee also has a unique employee Id and a salary. A position has a title and a short description.

Aggregation

- Represents a ‘whole-part’ relationship
  - ‘contain’
  - ‘is part of’

⟵ Association name is implied

- Code looks the same as an association.

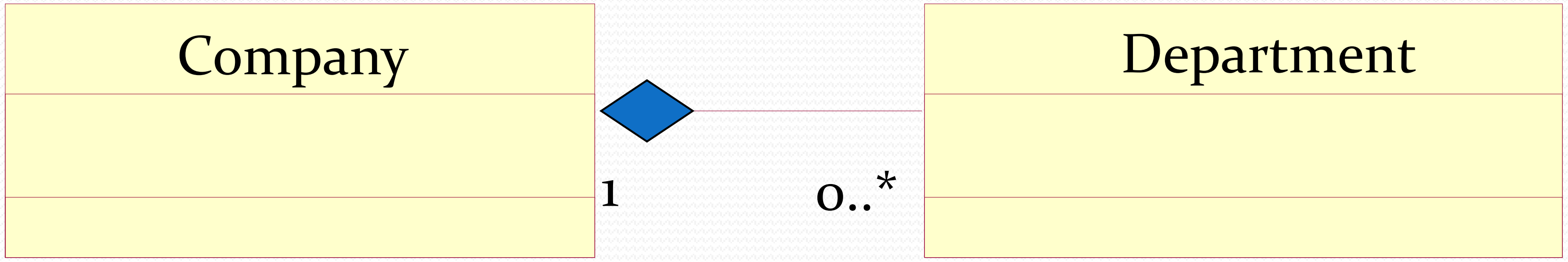


```
public class Department {
    private List<Student> students;
}
```

```
public class Student {
    private Department department;
}
```

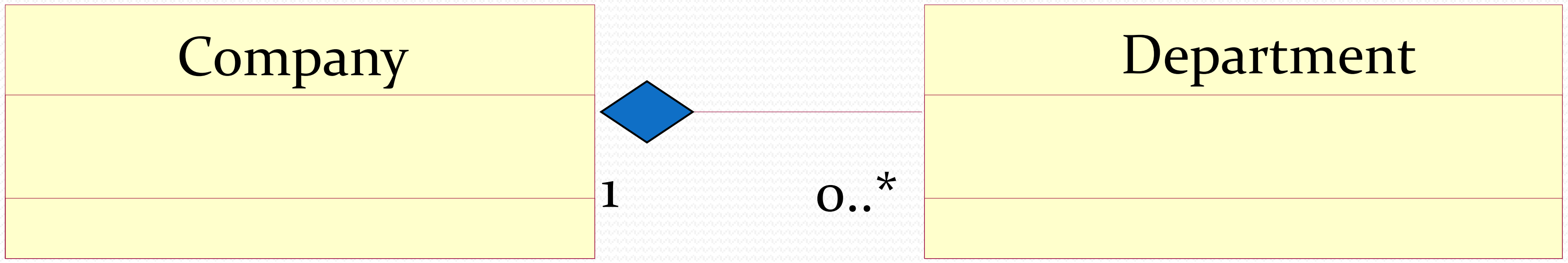
Composition

- Strong aggregation
  - ‘whole-part’ relationship
  - Parts ‘live-and-die’ with the whole.
  - Code impacts?



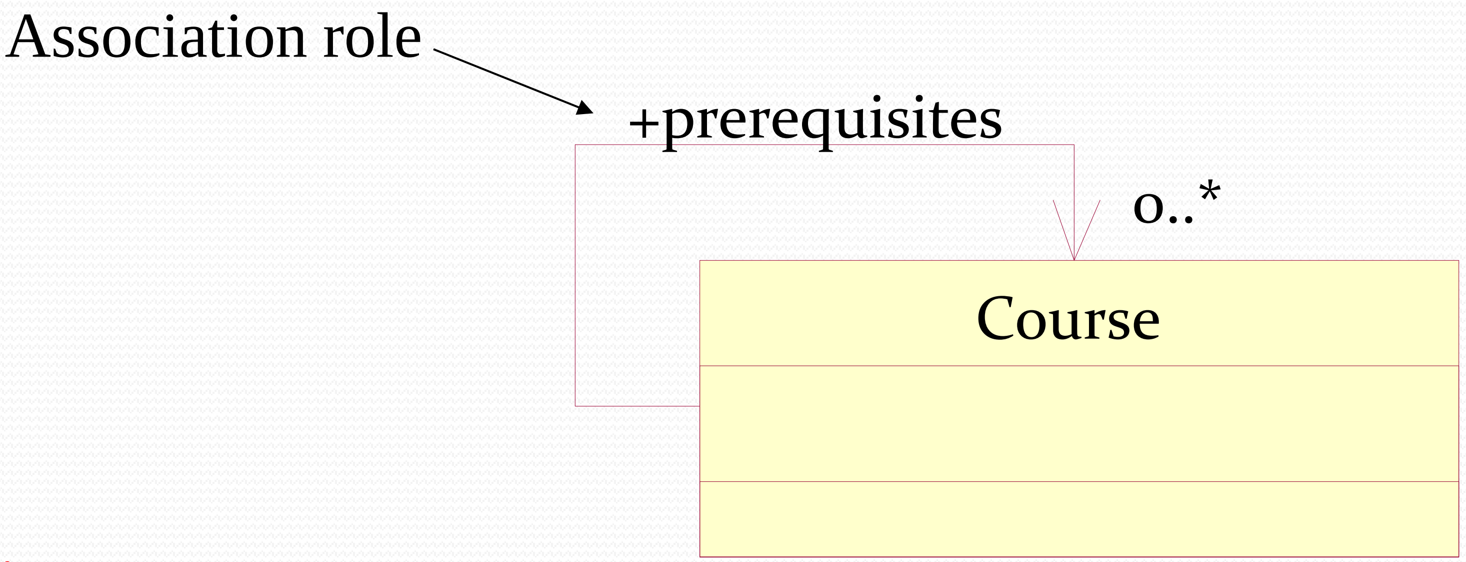
Composition

- Code impacts? *Will likely be in the logic for how we construct a department and what happens when a company is deleted.*



Reflexive Association

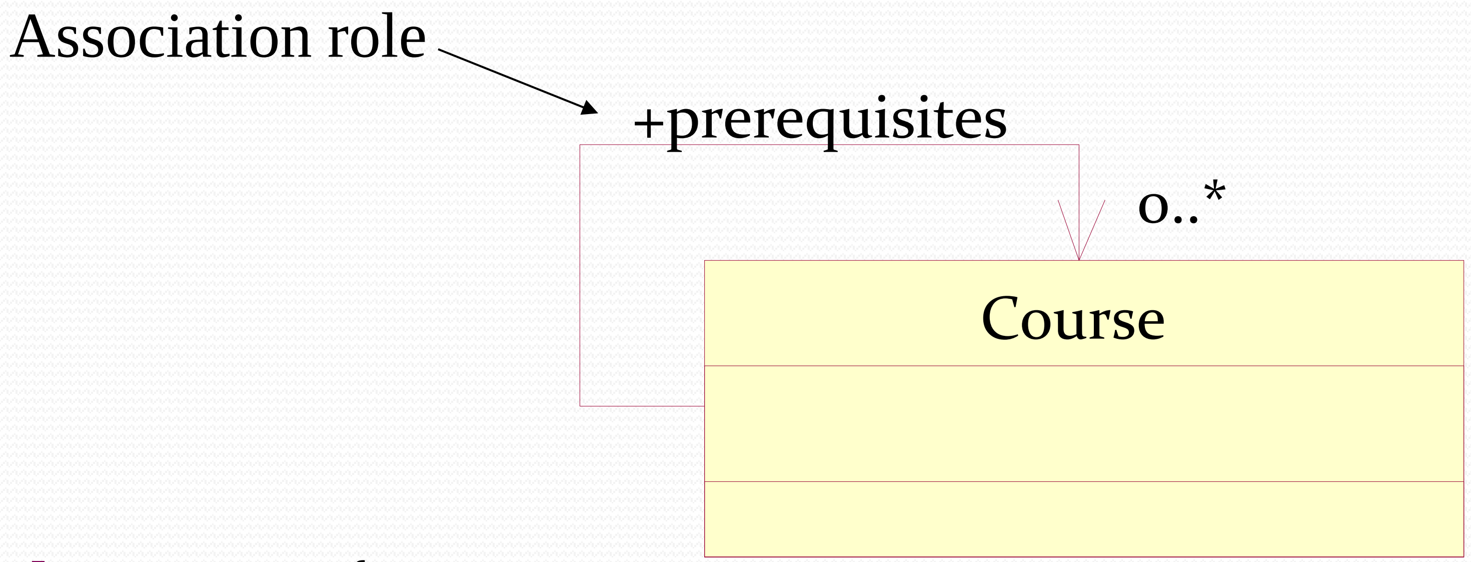
- Relationship between two or more objects of the same class.



What will the code be?

Reflexive Association

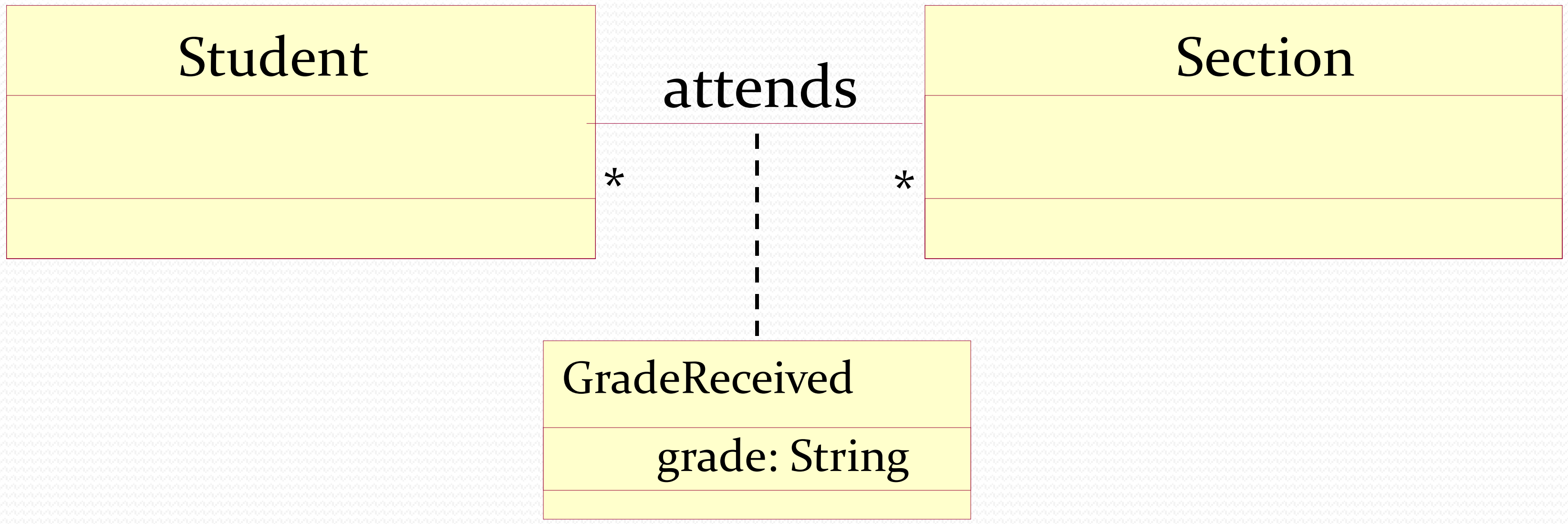
- Relationship between two or more objects of the same class.



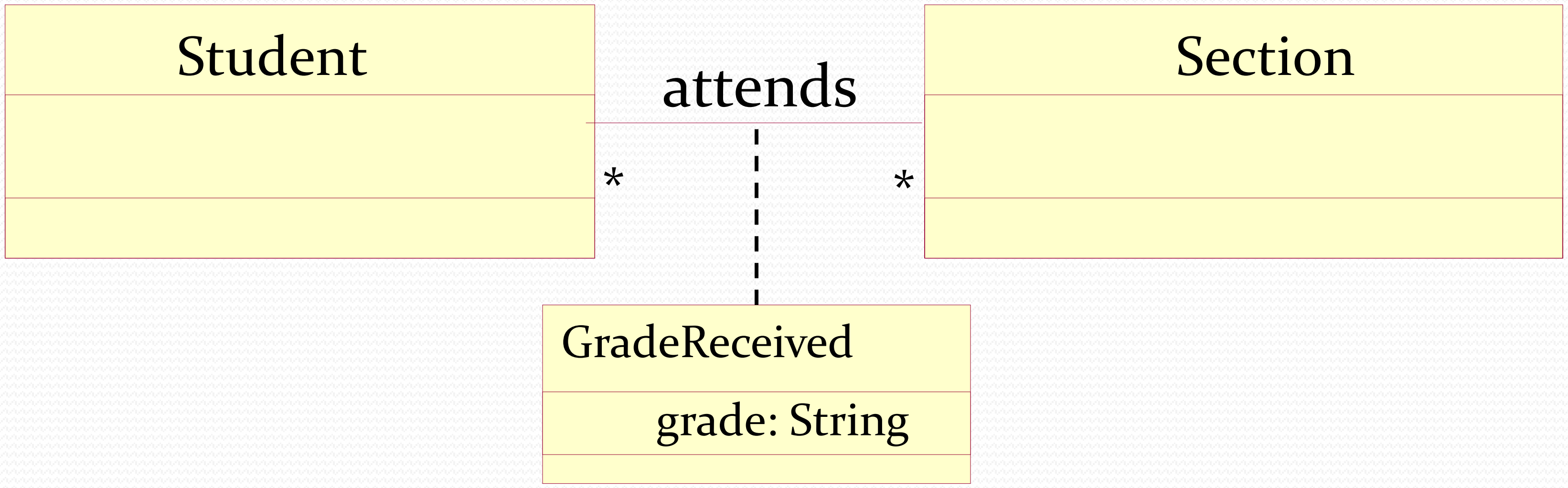
```
public class Course {
    private List<Course> prerequisites;
}
```

Association Classes

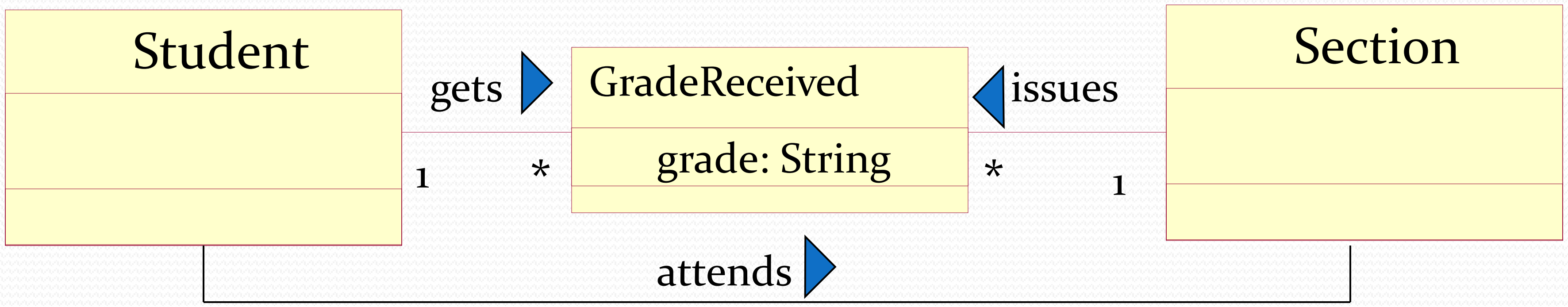
- Association Classes are useful to contain attributes of the link between objects.
  - These are often modeled this way during analysis



Association Class - reworked



- During design, association classes are often re-worked to reflect the code instead of the concepts



Main Point 3

There are several special forms of association, such as reflexive associations, aggregation, composition, and association classes.

Although most of these have their own symbols, you could still model these relationships without them.

The use of the symbols is to (easily) communicate additional information about the relationship.

Nevertheless, even these additional symbols are still based on a line, diversity on the basis of unity.

Mid-Term Prep Exercise (3-2)

- Objective: understand reflexive associations
- Task: Draw a UML class diagram for each of the following problem statements.

1. Doubly-Linked List:

   A LinkedList consists of zero or more ListItems. However, the LinkedList class only knows about the first ListItem. Each ListItem knows its previous and its next ListItem, if any.

2. Position Hierarchy:

   A position may or may not be a managerial position. If it is a managerial position, then other positions report to this one.

3. Course Prerequisites:

   A university course may or may not require the student to have taken other courses first before taking this one. Any course can be a prerequisite course for any other course, except for itself.

Summary

Today we looked at modeling associations:

- We can use an association matrix to analyze what the relations are between concepts
- Associations are modeled with a line, a name describing the association, numbers on each side to indicate multiplicity, and optionally an arrow for directionality
- Reflexive associations, aggregation, composition, and association classes are further model refinements of associations.

Connecting the Parts of Knowledge With the Wholeness of Knowledge

1. Class diagrams are defined in terms of concepts (classes) and their relationships (associations)

2. Although there are various special association forms (composition, aggregation, etc.), the code always consists of references to other objects

---

3. **Transcendental consciouness** by its very nature, is self-referral – the Self being aware of the Self – its fundamental association.

4. **Wholeness moving within itself**: In Unity Consciousness one feels intimately associated with all other things in creation as a result of perceiving all things in terms of one's Self