

COMP 3111

SOFTWARE ENGINEERING

LECTURE 6

SYSTEM REQUIREMENTS CAPTURE

DOMAIN MODELING EXERCISE

EXERCISE: ASU COURSE REGISTRATION

At the beginning of each term, students may request a course catalogue containing a list of course offerings needed for the term. Information about each course, such as instructor, department, and prerequisites are included to help students make informed decisions.

The new system will allow students to select four course offerings for the coming term. In addition, each student will indicate two alternative choices in case a course offering becomes filled or is canceled. No course offering will have more than forty students or fewer than ten students. A course offering with fewer than ten students will be canceled. Once the registration process is completed for a student, the registration system sends information to the billing system so the student can be billed for the term.

Instructors must be able to access the online system to indicate which courses they will be teaching, and to see which students signed up for their course offerings.

For each term, there is a period of time that students can change their schedule. Students must be able to access the system during this time to add or drop courses.

EXERCISE: ASU DOMAIN MODEL CLASSES

At the beginning of each term, students may request a course catalogue containing a list of course offerings needed for the term.

classes: ~~Term~~ → attribute (of CourseOffering)

Student → O.K.

~~CourseCatalogue~~ → irrelevant (physical system output)

CourseOffering → O.K.

Information about each course, such as instructor, department, and prerequisites are included to help students make informed decisions.

classes: Course → O.K.

Instructor → O.K.

Department → O.K.

~~Prerequisite~~ → role (of Course)

~~InformedDecision~~ → vague (mental process of students)

EXERCISE: ASU DOMAIN MODEL CLASSES

The new system will allow students to select four course offerings for the coming term.

classes: ~~System~~ → implementation construct

In addition, each student will indicate two alternative choices in case a course offering becomes filled or is canceled.

classes: ~~AlternativeChoice~~ → redundant (same as CourseOffering)

No course offering will have more than forty students or fewer than ten students.

classes: *no new classes*

A course offering with fewer than ten students will be canceled.

classes: *no new classes*

EXERCISE: ASU DOMAIN MODEL CLASSES

Once the registration process is completed for a student, the registration system sends information to the billing system so the student can be billed for the term.

classes: ~~RegistrationProcess~~ → **operation** (activity of using the system)

~~Information~~ → **vague** (need to specify more clearly)

~~BillingSystem~~ → **irrelevant** (external system)

Instructors must be able to access the online system to indicate which courses they will be teaching, and to see which students signed up for their course offerings.

classes: *no new classes*

For each term, there is a period of time that students can change their schedule.

classes: ~~PeriodOfTime~~ → **vague** (related to processing constraints)

~~Schedule~~ → **redundant** (same as CourseOffering)

EXERCISE: ASU DOMAIN MODEL CLASSES

Students must be able to access the system during this time to add or drop courses.

classes: *no new classes*

EXERCISE:

ASU DOMAIN MODEL INITIAL CLASS DIAGRAM

Classes

Student

CourseOffering

Course

Instructor

Department

Student

Instructor

Department

*Course
Offering*

Course

EXERCISE: ASU DOMAIN MODEL ASSOCIATIONS

At the beginning of each term, **students** may request a course catalogue containing a list of **course offerings** needed for the term.

associations: None (possible classes eliminated)

Information about each **course**, such as **instructor**, **department**, and prerequisites are included to help **students** make informed decisions.

associations: Course *IsPrerequisiteFor* Course → **O.K.**
Department *Offers* Course → *inferred* → **O.K.**
Instructor *Teaches* CourseOffering → *inferred* → **O.K.**

The new system will allow **students** to select four **course offerings** for the coming term.

associations: ~~Student *Selects* CourseOffering~~ → **O.K.** Student *EnrollsIn* CourseOffering

multiplicity: max-card(Student, *EnrollsIn*) = 4

EXERCISE: ASU DOMAIN MODEL ASSOCIATIONS

In addition, each **student** will indicate two **course offering** alternative choices in case a **course offering** becomes filled or is canceled.

associations: ~~Student Indicates CourseOffering~~ → **O.K.** Student *HasAlternative* CourseOffering
~~Student Fills CourseOffering~~ → operation
~~? Cancels CourseOffering~~ → operation

multiplicity: $\text{max-card}(\text{Student}, \text{HasAlternative}) = 2$

No **course offering** will have more than forty **students** or fewer than ten **students**.

multiplicity: $\text{max-card}(\text{CourseOffering}, \text{EnrollsIn}) = 40$

A **course offering** with fewer than ten **students** will be canceled.

associations: *no new associations*

EXERCISE: ASU DOMAIN MODEL ASSOCIATIONS

Once the registration process is completed for a **student**, the registration system sends information to the billing system so the **student** can be billed for the term.

associations: None (possible classes eliminated)

Instructors must be able to access the online system to indicate which **course offering** they will be teaching, and to see which **students** signed up for their **course offerings**.

associations: ~~Instructor Indicates CourseOffering~~ → operation

Instructor *Teaches* CourseOffering → **O.K.** (already captured)

~~Instructor Sees Student~~ → irrelevant

~~Student SignsUpFor CourseOffering~~ → operation

EXERCISE: ASU DOMAIN MODEL ASSOCIATIONS

For each term, there is a period of time that **students** can change their **course offering** schedule.

associations: ~~Student Changes CourseOffering~~ → operation

Students must be able to access the system during this time to add or drop courses. **course offering**

associations: ~~Student Adds CourseOffering~~ → operation
~~Student Drops CourseOffering~~ → operation

EXERCISE:

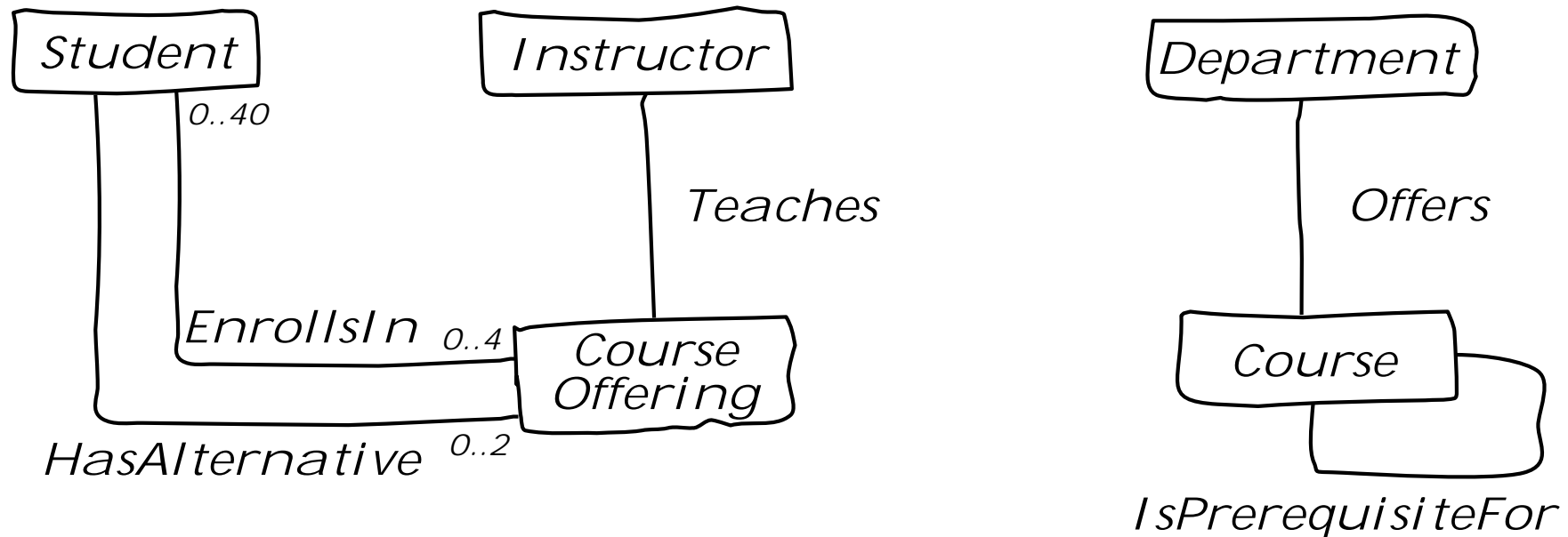
ASU DOMAIN MODEL INITIAL CLASS DIAGRAM

Classes

Student
CourseOffering
Course
Instructor
Department

Associations

Instructor *Teaches* CourseOffering
Department *Offers* Course
Course *IsPrerequisiteFor* Course
Student *EnrollsIn* CourseOffering
Student *HasAlternative* CourseOffering



EXERCISE:

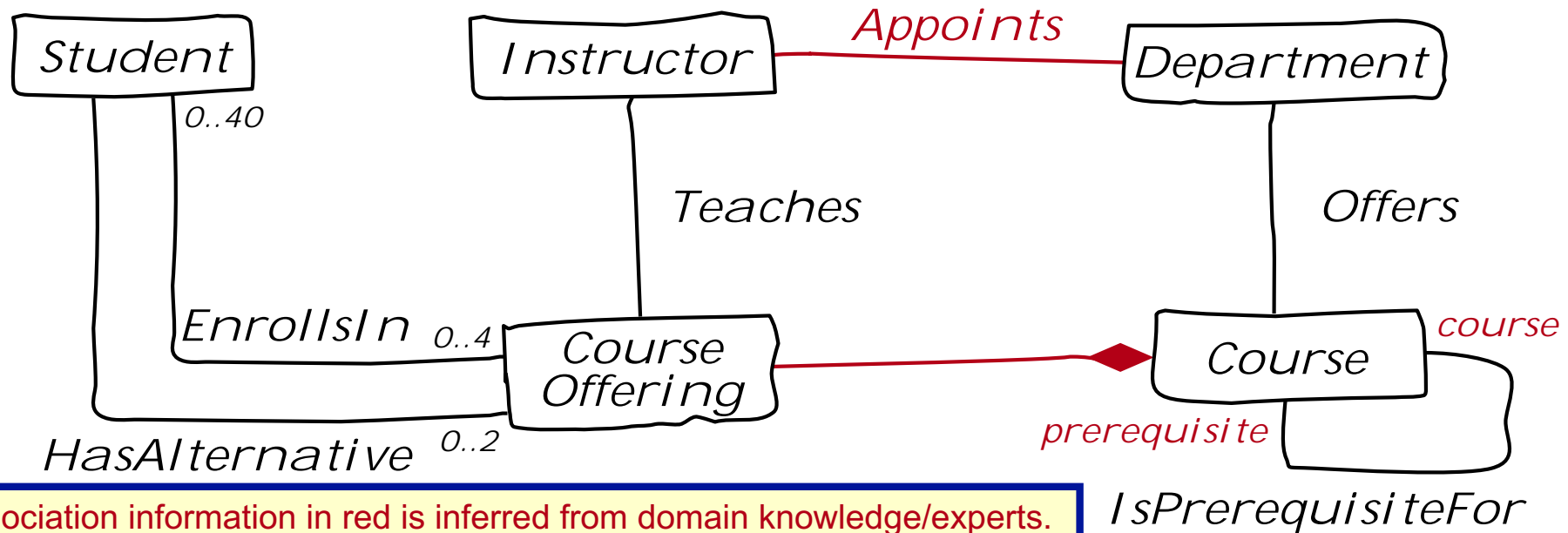
ASU DOMAIN MODEL FIRST REFINEMENT

Classes

Student
CourseOffering
Course
Instructor
Department

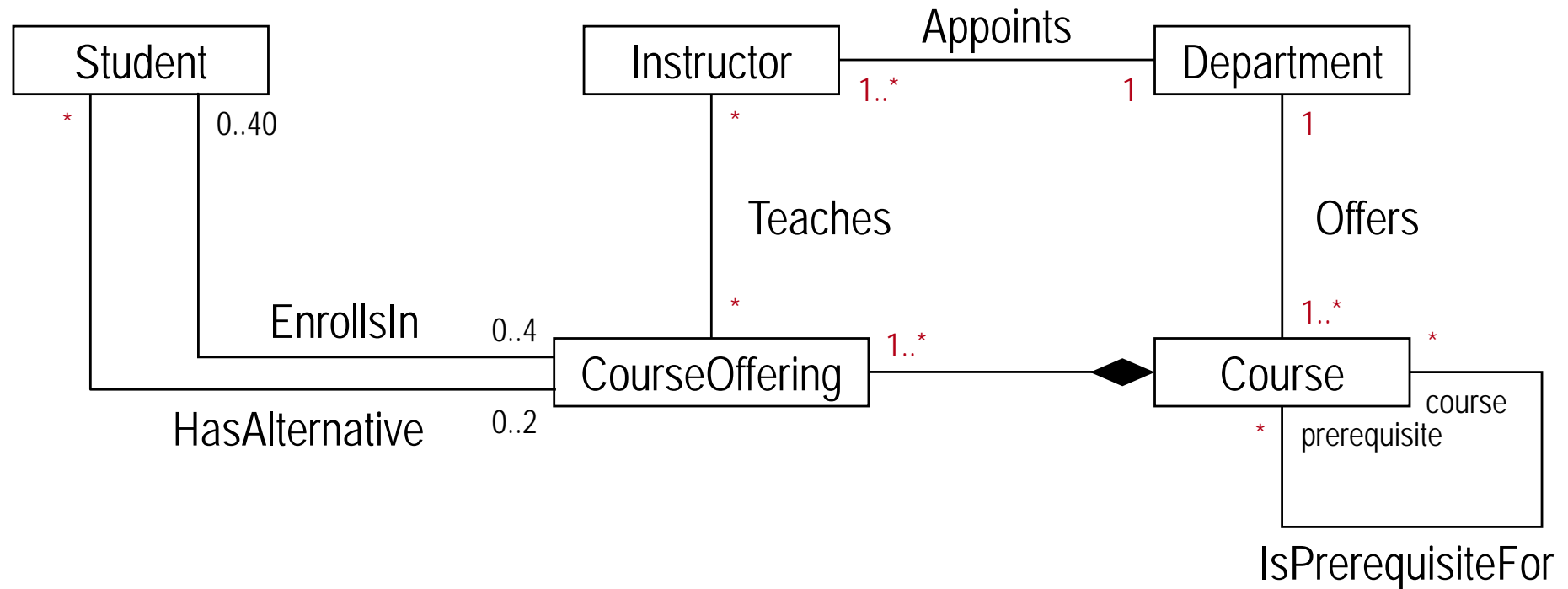
Associations

Instructor *Teaches* CourseOffering
Department *Offers* Course
Course *IsPrerequisiteFor* Course
Student *EnrollsIn* CourseOffering
Student *HasAlternative* CourseOffering



EXERCISE:

ASU DOMAIN MODEL ASSOCIATION MULTIPLICITY



Adding other known association multiplicities.

EXERCISE:

ASU DOMAIN MODEL ASSOCIATION MULTIPLICITY

- Is the multiplicity of the CourseOffering end of the EnrollsIn association realistic?

✎ A student can enroll in at most four course offerings *during their entire time at the university.*

- Actually, the constraint should be:

✎ A student can enroll in at most four course offerings *in one semester.*

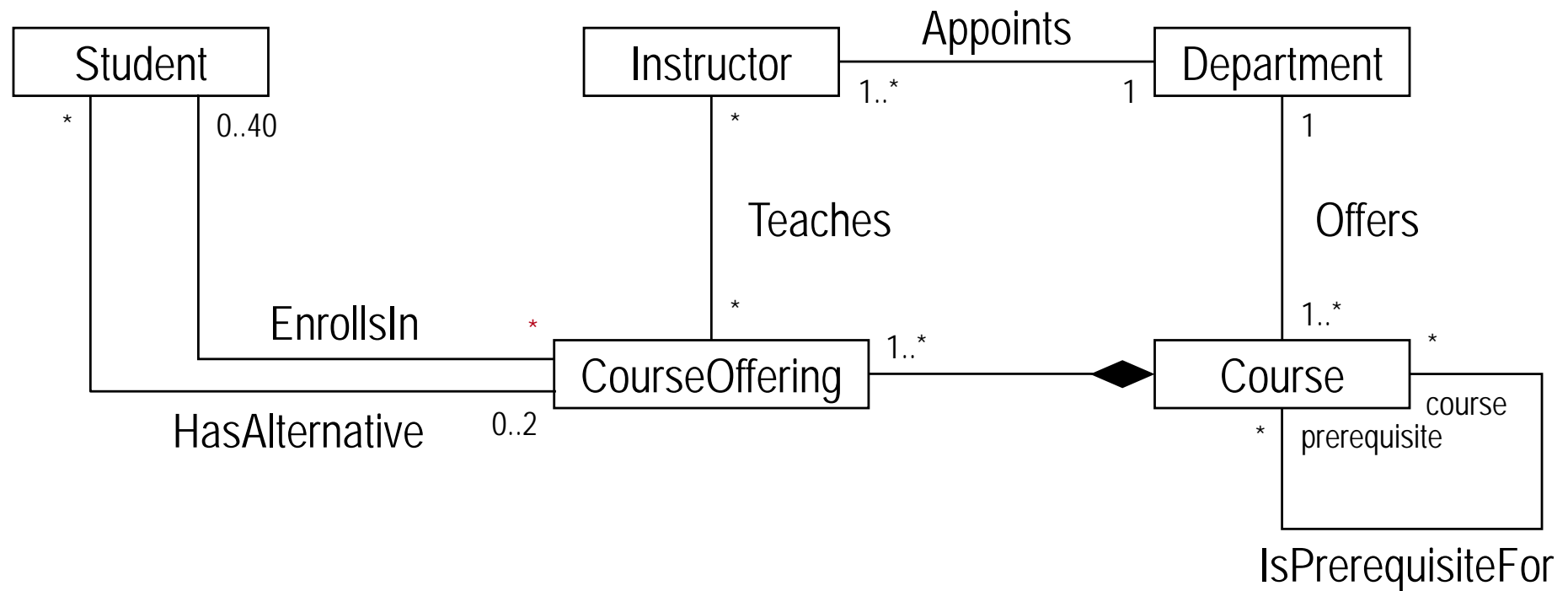
- It is actually unknown the total number of course offerings in which a student will enroll.

✎ The max-card should be *.

- We need to *document outside the domain model* the fact that enrollment in a term should not be more than four course offerings (e.g., by using OCL).

EXERCISE:

ASU DOMAIN MODEL ASSOCIATION MULTIPLICITY



EXERCISE: ASU DOMAIN MODEL ATTRIBUTES

Student
userId :String password : String userGroup : Integer surname : String otherNames : String address : String DOB : Date dateOfAdmission : Date levelOfStudy : String modeOfStudy : String yearOfStudy : Integer

Instructor
userId : String password : String userGroup : Integer surname : String otherNames : String address : String DOB : Date qualification : String dateOfAppointment : String

Department
code: String title : String

Course
code : String title : String description: String credits : Integer

CourseOffering
offeringId : String term : String year : Integer

**Most of these
attributes are
obtained from the
domain experts.**

EXERCISE:

ASU DOMAIN MODEL GENERALIZATION

Student
userId :String password : String userGroup : Integer surname : String otherNames : String address : String DOB : Date dateOfAdmission : Date levelOfStudy : String modeOfStudy : String yearOfStudy : Integer

Instructor
userId : String password : String userGroup : Integer surname : String otherNames : String address : String DOB : Date qualification : String dateOfAppointment : String

Department
code: String title : String

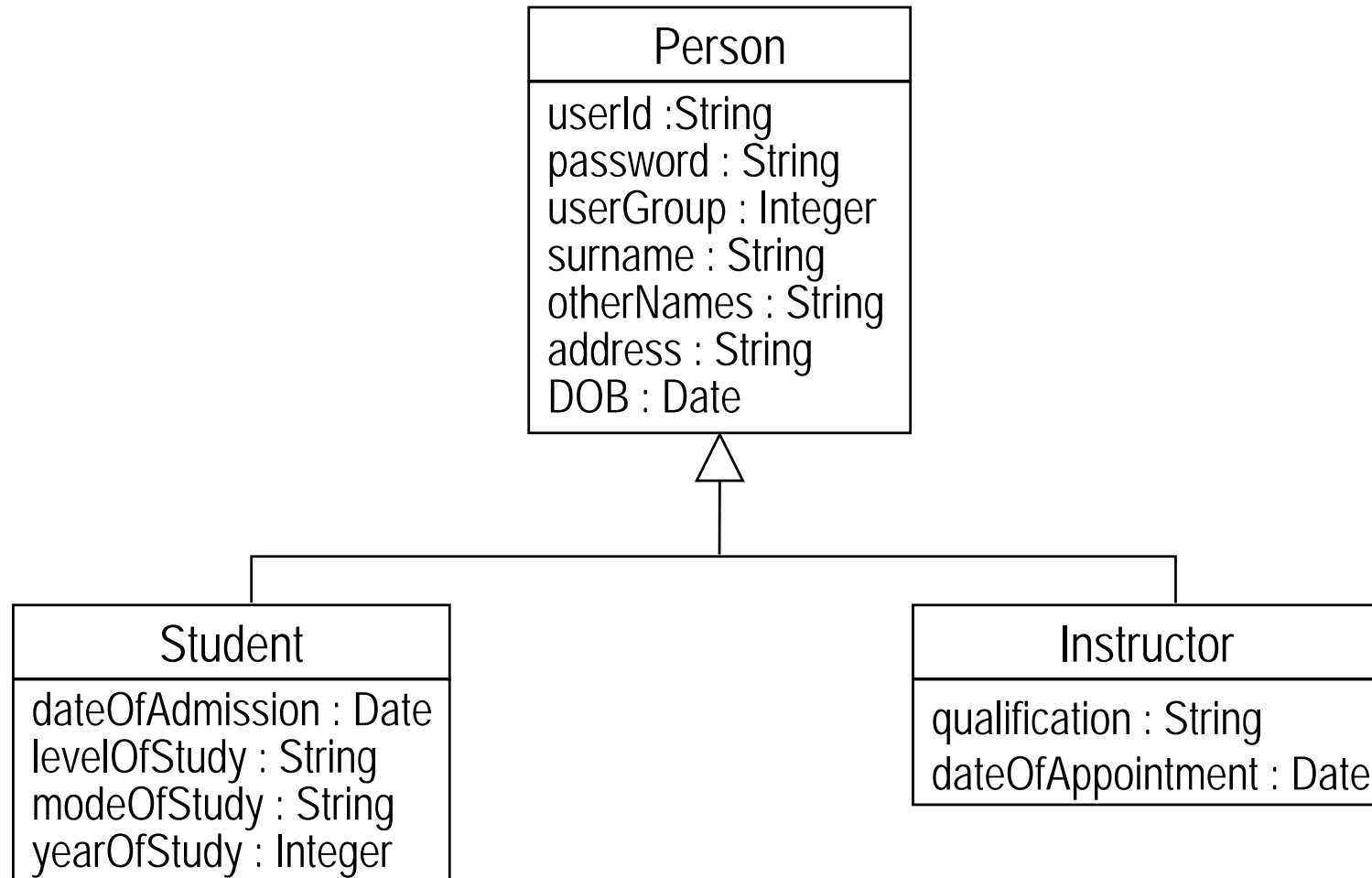
Course
code : String title : String description: String credits : Integer

CourseOffering
offeringId : String term : String year : Integer

Look for common
properties and
semantics among
classes.

EXERCISE:

ASU DOMAIN MODEL GENERALIZATION



EXERCISE:

ASU DOMAIN MODEL GENERALIZATION

