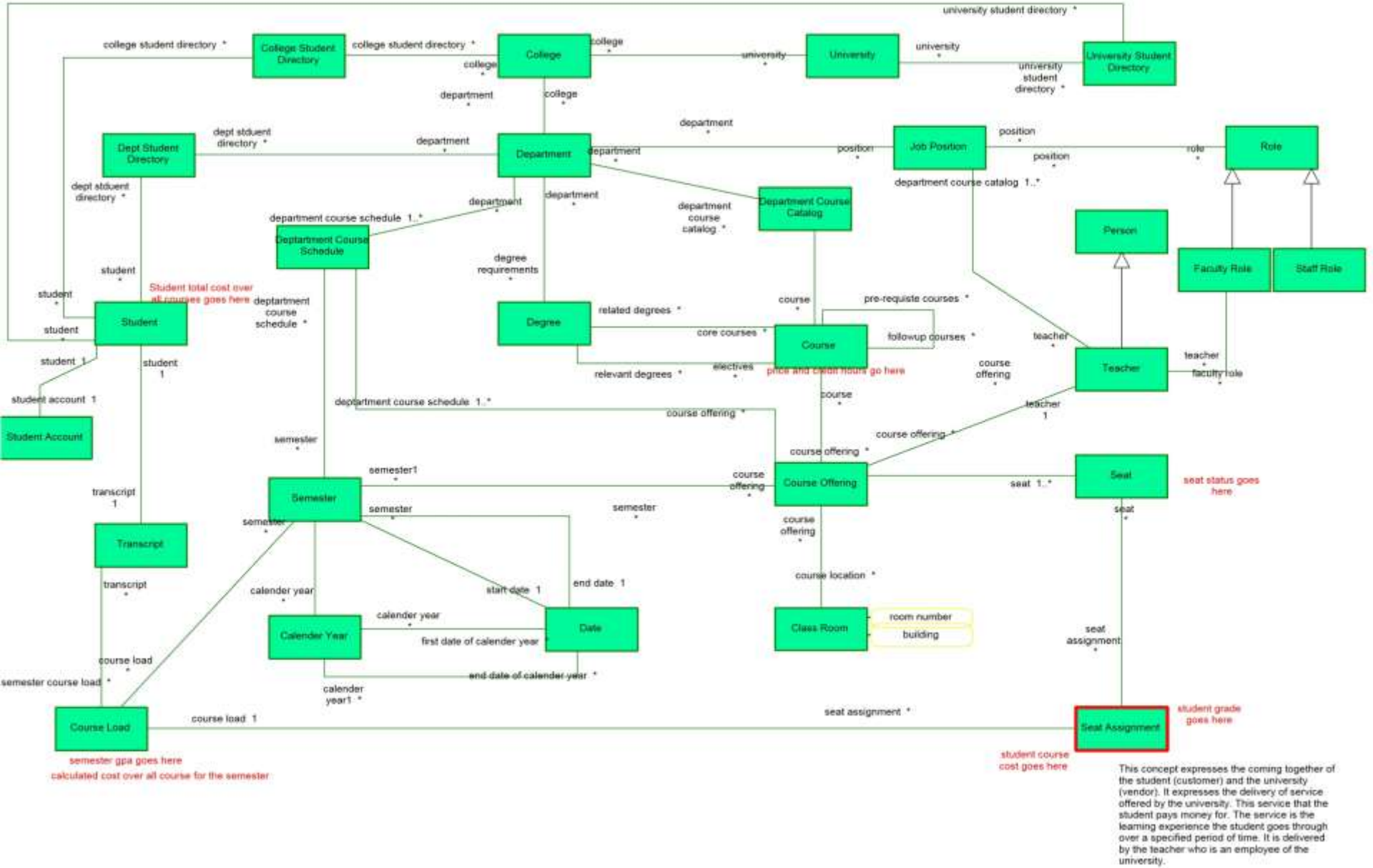
Info 5100: Application Engineering and Development

Business Case Study

Prof Kal Bugrara, PhD.

Objectives

- **How to read a complex domain model**
- **How to abstract a social environment into a model with strong focus on people and subjects and users**
- **Design management and control structure into the model**
- **Understand the full potential of a digital domain model**
  - The whys:
    - Strategic such as
      - how to improve the educational systems (more competitive graduates) through learning and feedback
      - new educational systems by passing existing systems
    - Operational: Improved service delivery


university student directory *
college student directory *
College Student Directory
college student directory *
College
college
university
University
university
University Student Directory
university student directory *
college
college
department
college
department
Dept Student Directory
dept stduent directory *
department
Department
department
position
Job Position
position
position
role
Role
department course catalog 1..*
dept stduent directory *
department
department
department course schedule 1..*
Deptartment Course Schedule
department course catalog *
Department Course Catalog
Person
Faculty Role
Staff Role
student
Student total cost over all courses goes here
degree requirements
student
deptartment course schedule *
Degree
related degrees *
course
pre-requisite courses *
student
core courses *
Course
followup courses *
teacher
student 1
student 1
relevant degrees *
electives *
price and credit hours go here
course offering *
Teacher
teacher
faculty role
studeet account 1
department course schedule 1..*
course
course offering *
teacher 1
Student Account
semester
course offering *
semester1
Semester
course offering
Course Offering
seat 1..*
Seat
seat status goes here
transcript 1
semester
semester
semester
course offering *
seat *
Transcript
transcript *
calender year *
start date 1
end date 1
course location *
calender year
Calender Year
first date of calender year *
Date
Class Room
room number
building
seat assignment *
course load *
semester course load *
end date of calender year *
calender year1 *
seat assignment *
Course Load
course load 1
Seat Assignment
student grade goes here
semester gpa goes here
calculated cost over all course for the semester
student course cost goes here
This concept expresses the coming together of the student (customer) and the university (vendor). It expresses the delivery of service offered by the university. This service that the student pays money for. The service is the learning experience the student goes through over a specified period of time. It is delivered by the teacher who is an employee of the university.

The Problem Statement

Build a university-wide student registration and management system. The system must enable the university to manage its operations at the student, department, and college levels. This will allow the university to devise more effective ways of: measuring unit performance, control costs, forecast enrollments, and implementing marketing plans.

Dept

Responsible for

Academic matters such as students and courses

Business matters such as staff, faculty, etc.

Maintain course catalog and schedule courses

Issues important at this level

What are the courses we teach?

What are the courses we offer at any given semester?

Which courses are core and which ones are electives? What are the course requirements?

What are the degree requirements?

What is our current capacity? How many seats are empty?

What is our faculty/student ratio per class? How do we compare with other depts in the college?

What is the average number of students per class? Largest class? Smallest class?

What is the current student enrollment in our department?

What is the administrative staff to faculty ratio?

What is the ratio of full time faculty vs part-time?

What is the percentage of faculty with Ph.Ds?

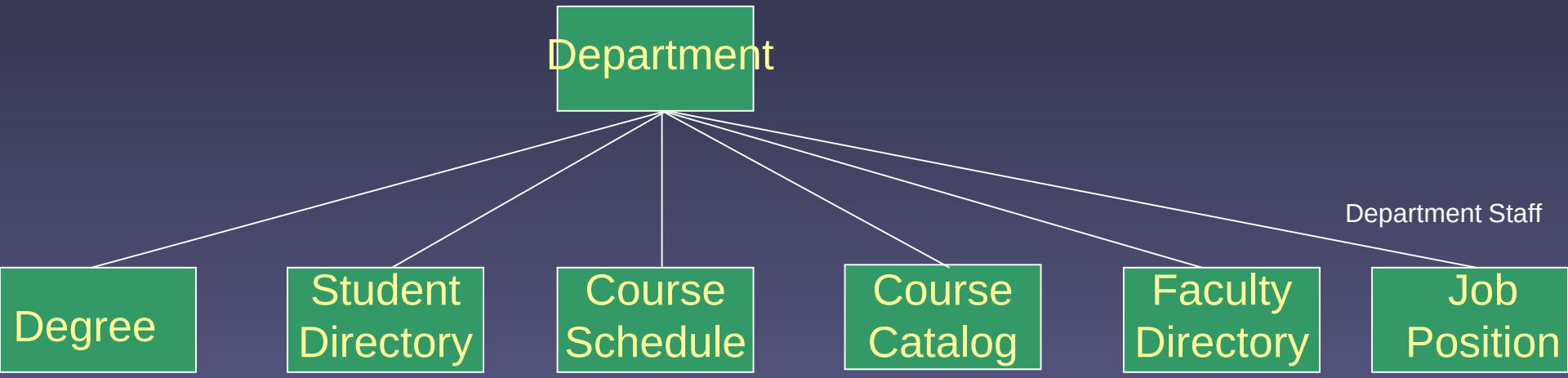
Department Stakeholders and Users

- **Students**
  - Needs: Review courses offered, Register for courses, Review Teacher profiles, Graduation Requirements, Review Academic Status, Departmental student portal, etc
- **Teachers**
  - Student profiles, Grade submission, Contact Info
- **Departmental Staff**
  - Maintain course catalog, Manage course schedule, Student grade submission, course enrollment numbers, student status, Manage Teachers and their profiles
- **College Administrators**
  - Department enrollment numbers, Department performance evaluation
- **Parents**
  - Real-time access to student class standing?

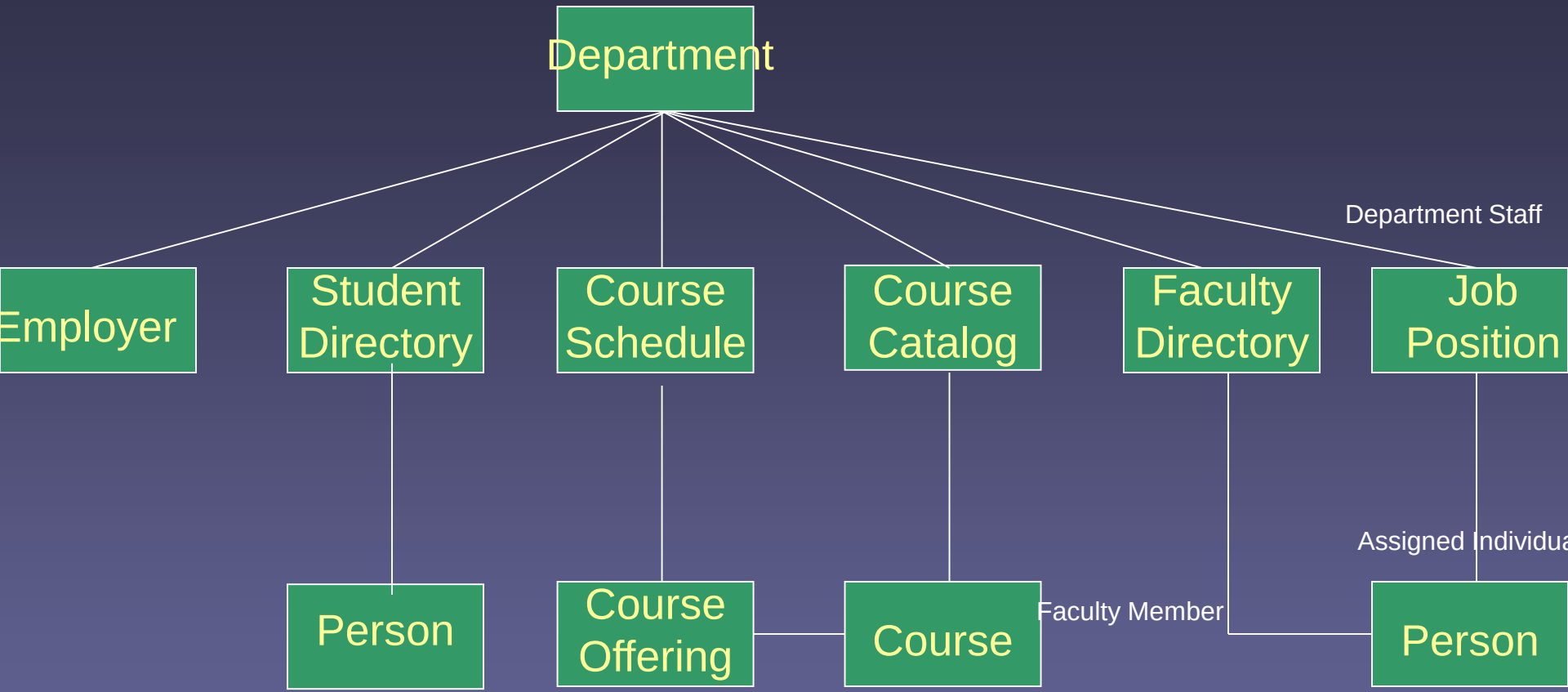
Department

- **Has products and services**
- **Has Demand for its products and services**
  - Customers and course offerings
- **Has a service delivery process of how these products and services will be delivered to its customers.**
  - Instructors, instructors teaching classes, evaluating student performance, assigning grades
- **Has a way of measuring how it is performing through management and feedback**

Department


Department
Department Staff
Degree
Student Directory
Course Schedule
Course Catalog
Faculty Directory
Job Position

An Object Model

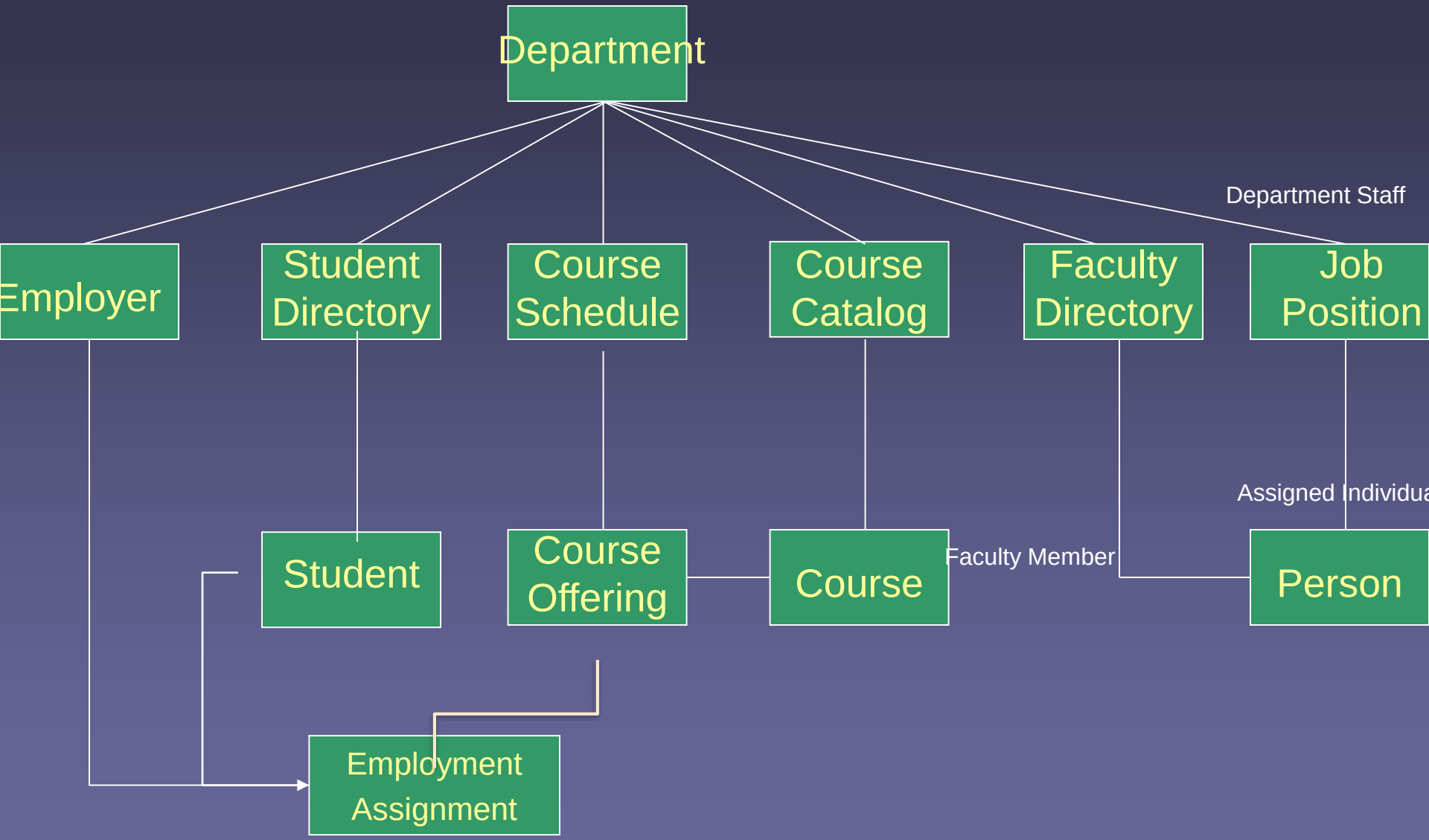


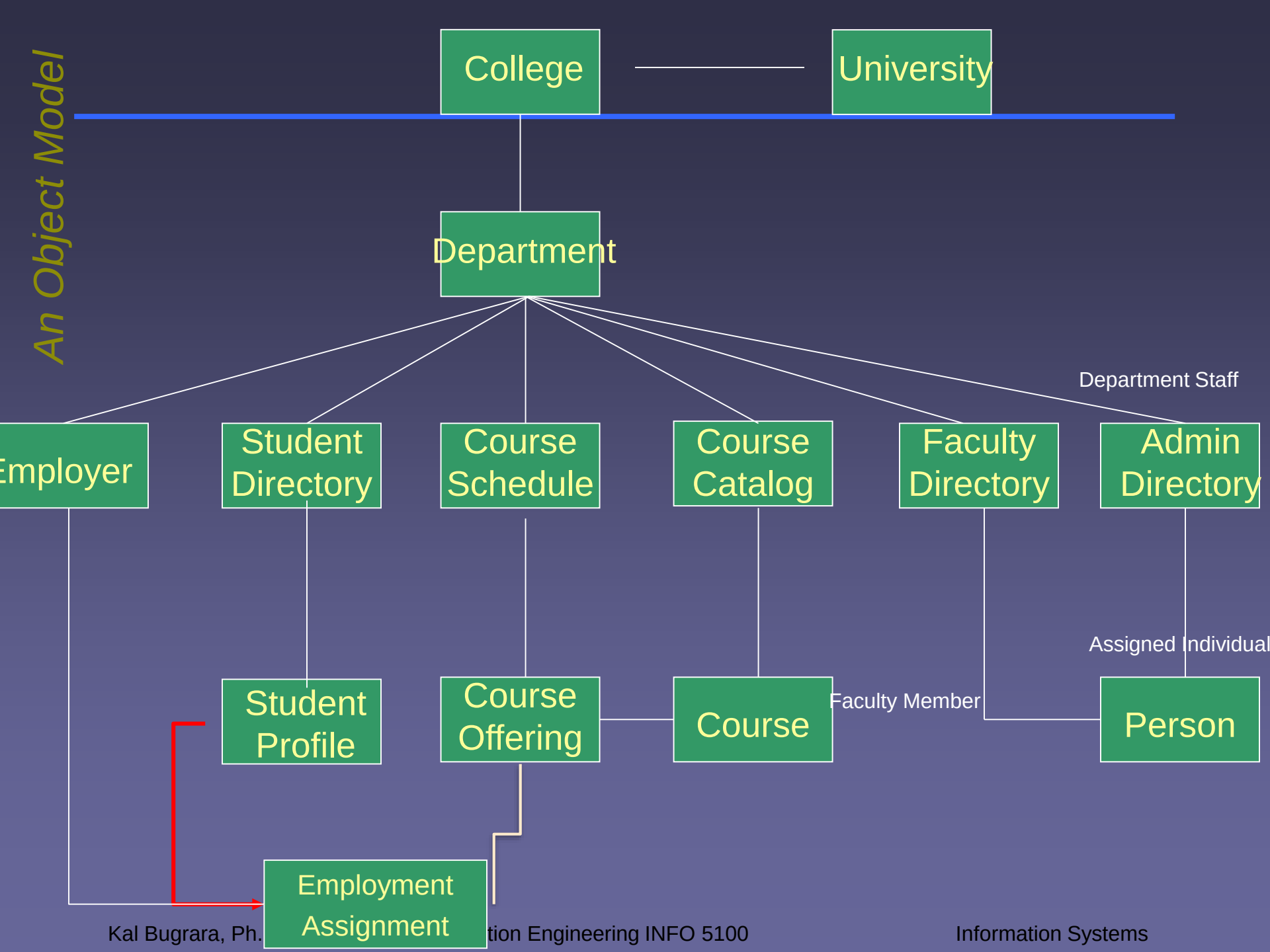
How to maximize the value of learning

Scenario: A digital platform to support A professor as a service

- **A professor offers courses in their field**
- **Professor services are accessible world wide**
- **Manages their own courses**
- **Students take classes from many professors**
- **A third party ceritification to certify degree**
- **Pricing can vary and easy to adjust**
- **Universities can focus on research**

An Object Model




An Object Model
College
University
Department
Department Staff
Employer
Student Directory
Course Schedule
Course Catalog
Faculty Directory
Admin Directory
Assigned Individual
Student Profile
Course Offering
Course
Faculty Member
Person
Employment Assignment

Course Offering vs Course

Course Offering

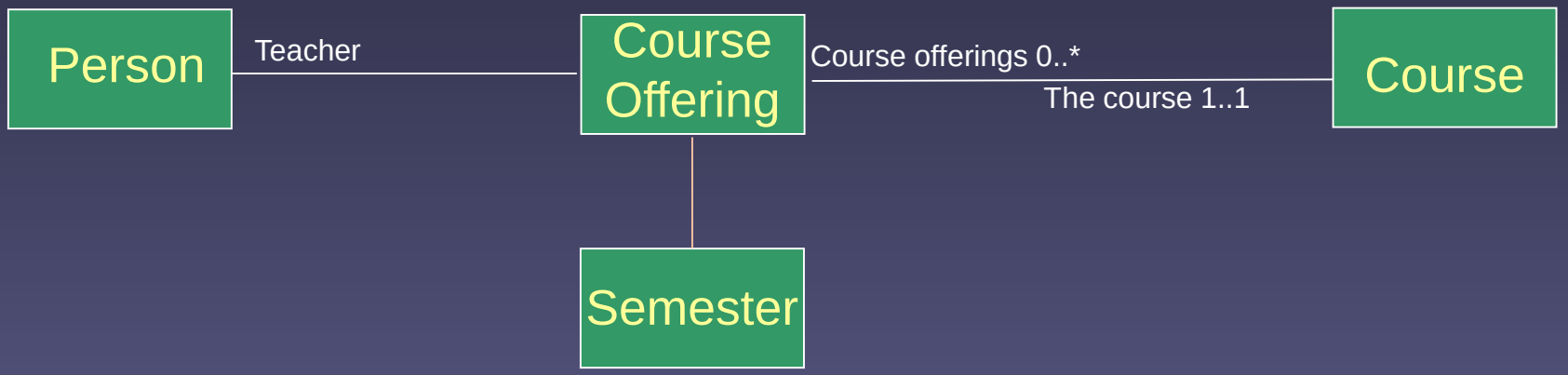
Models a specific offering of course
An actual class
Time bound: month and year
Has a teacher
Class room

Course

Course description
Exists in course catalog only
Independent of teacher


Course Offering
Course offerings 0..*
The course 1..1
Course

Order of generating the data



Professor Jones taught Info 6205 last summer
Professor Ozbek taught Info 6205 last spring
Professor Jag taught Info 6205 this Fall

Info 6205

What did prof Jag teach last fall? Did Prof Dino teach any infor 60250

Who taught info 6205 lately?

Course Offering vs Course

Course offerings

0

Course	Fall 2020	Spring 2021	Summer 2021	Fall 2021
Info 5001				
Info 5100				
Info 6210				
Info 6101				

1 2 2

Every Semester there is a schedule

Course offerings

Course	Fall 2020	Spring 2021	Summer 2021	Fall 2021
Info 5001				
Info 5100				
Info 6210				
Info 6101				

```
HashMap<String, CourseSchedule> mastercoursecatalog;
```

```
public Department(String n) {
    name = n;
    mastercoursecatalog = new HashMap<String, CourseSchedule>();
```

Transcript == Course Loads

Course offerings

Course 0	Fall 2022	Spring 2023	Summer 2023	Fall 2023
Course load 1	Info 6210 Info 5100			
Course load 2		Info 6205 Info 6105		
Course load 3			Info 6150	
Course load 4				Info 6250 CSYE 7280
	1	2	3	4

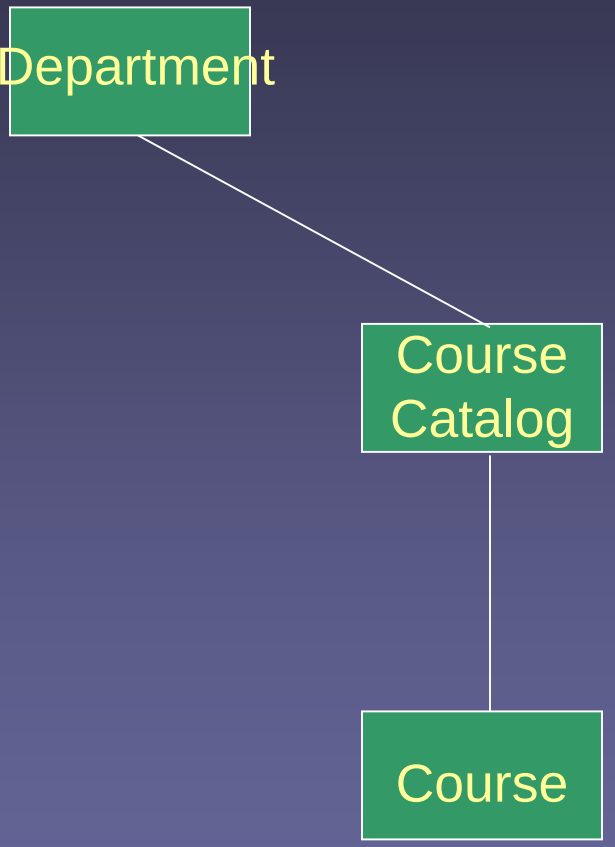
Department Manages Course Catalog

Manage the department course catalog which represent all the courses the department has to offer or the courses the department is capable of teaching (Capability)

Course catalogs are updated once a year (if at all); so course catalogs are independent of time but we keep track of last time the catalog was updated

Example of cc responsibilities:

- Find course given the course name
- Find course given course number
- List all courses
- Update course



Attributes:
name, Number, description

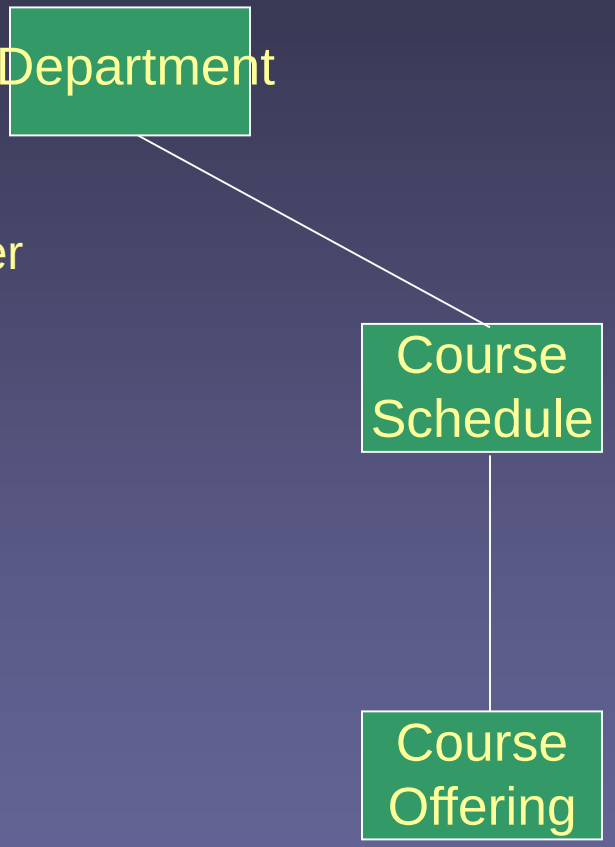
Department Manages Course Schedule

Manage the department course schedule which represent all the courses the department is scheduled (committed) to teach in a given semester

Course schedules are created every semester.
Course schedules are time dependent

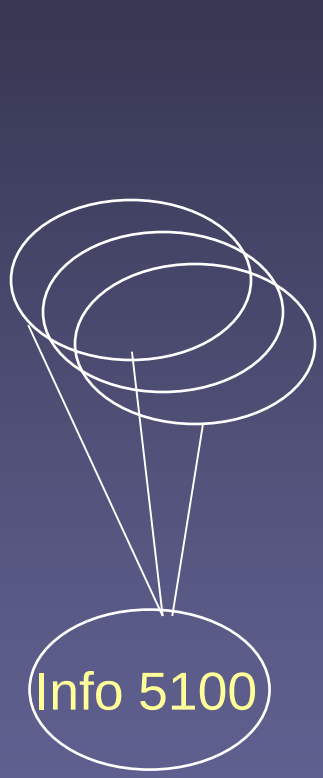
Example of CS responsibilities:

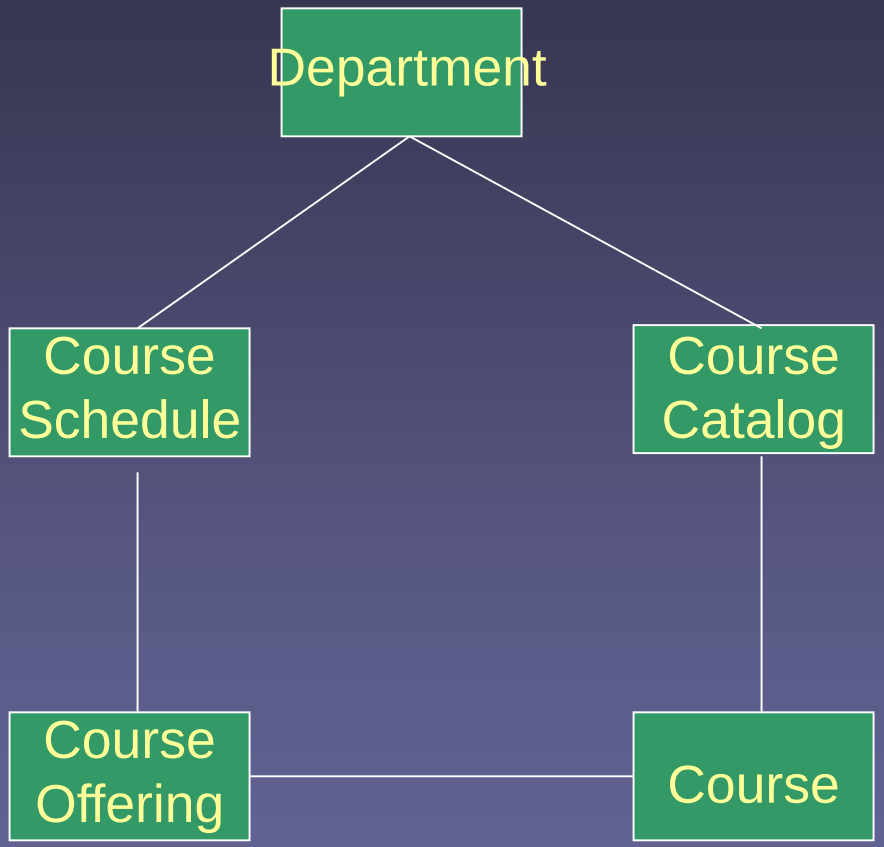
- Which courses are taught in a given semester
- Add a course to the course schedule
- Assign a teacher to teach a scheduled course



Attributes:
name, Number, description

Course Offering vs Course





What is a course offering

Course Offering

It is the actual class being taught at some point in time. It answers questions such as:

What is the course being offering?

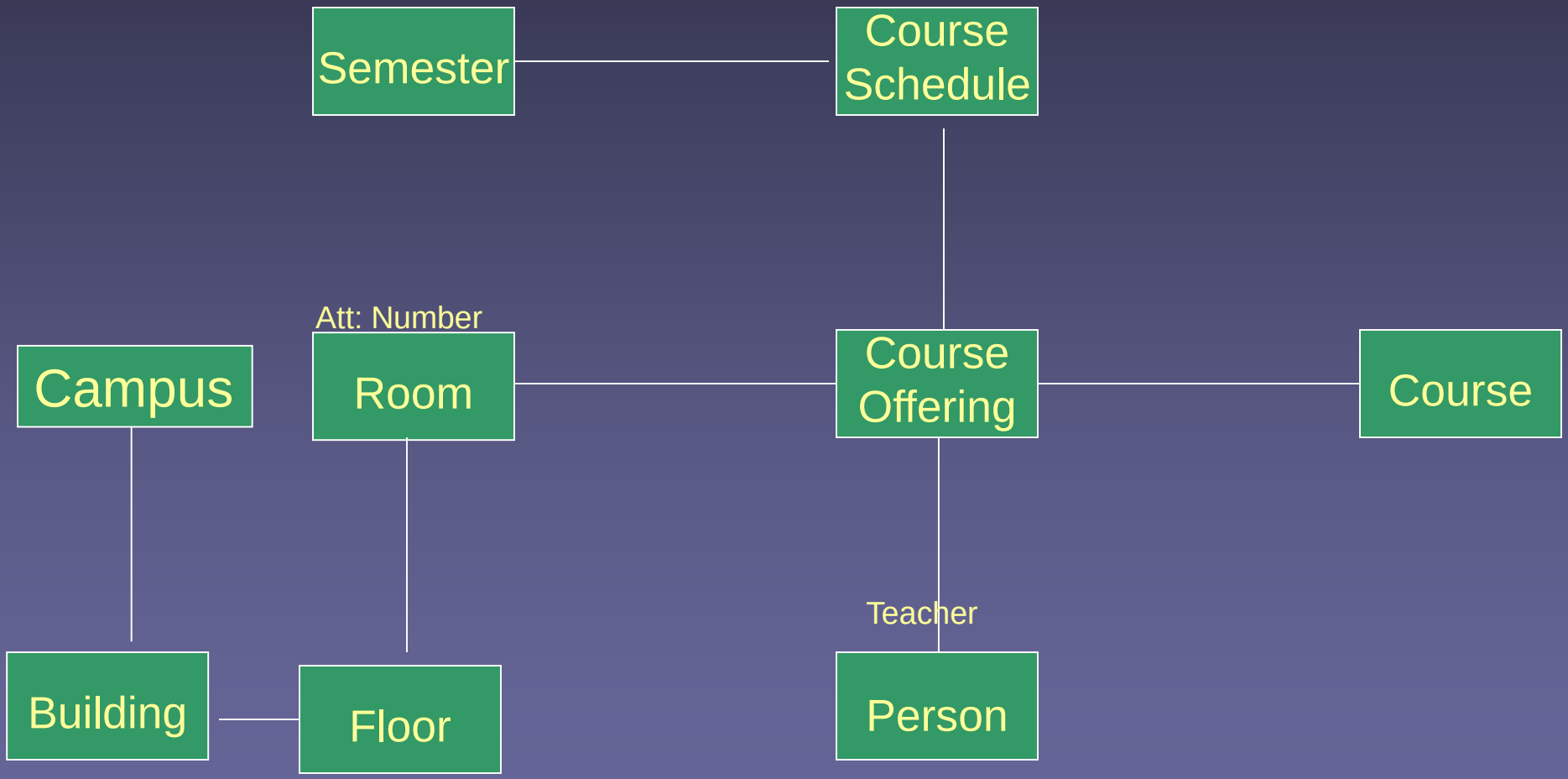
Who is teaching the course

When is it offered?

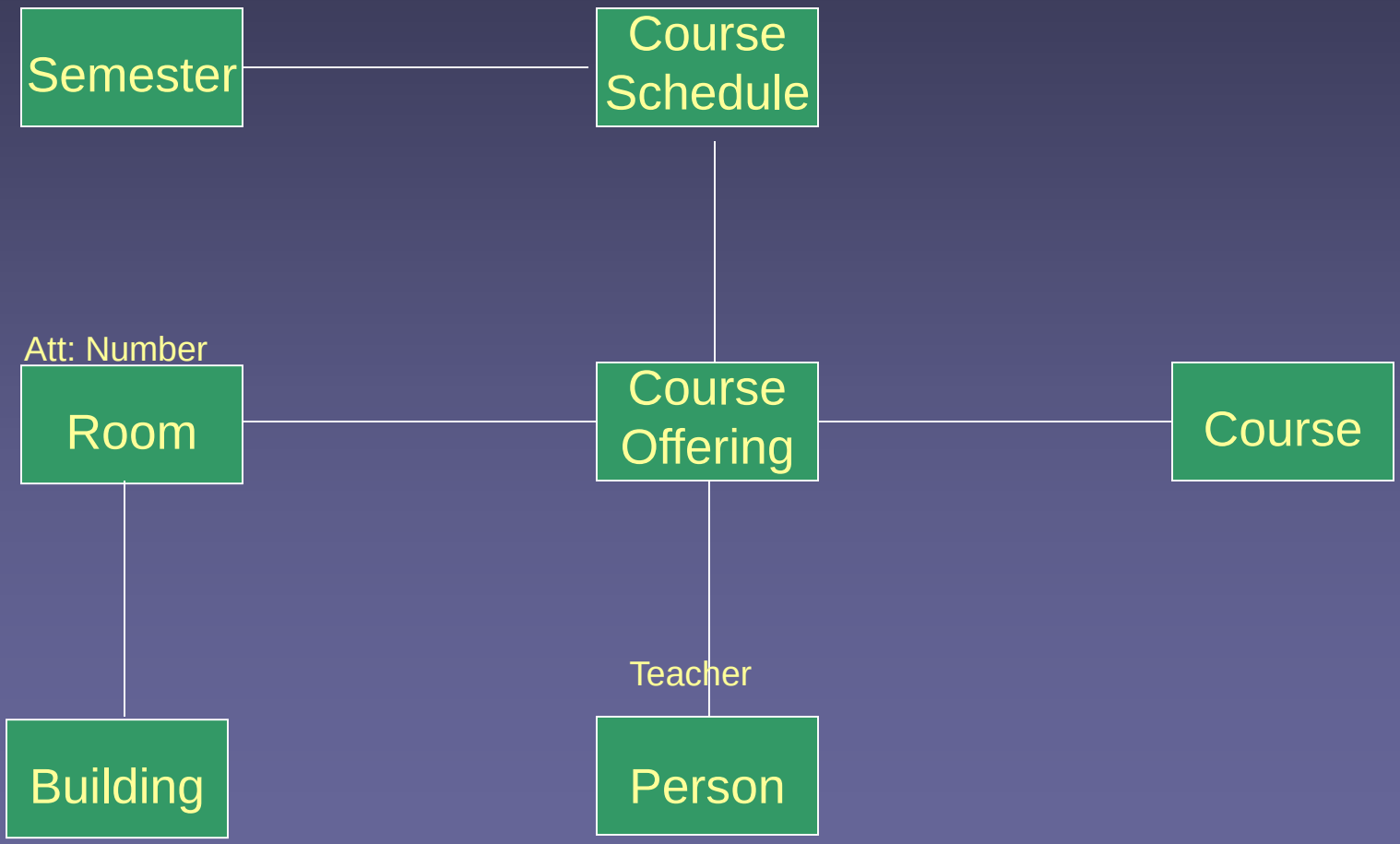
Where is it offered

Who are the students signed up for the class?

What is a course offering

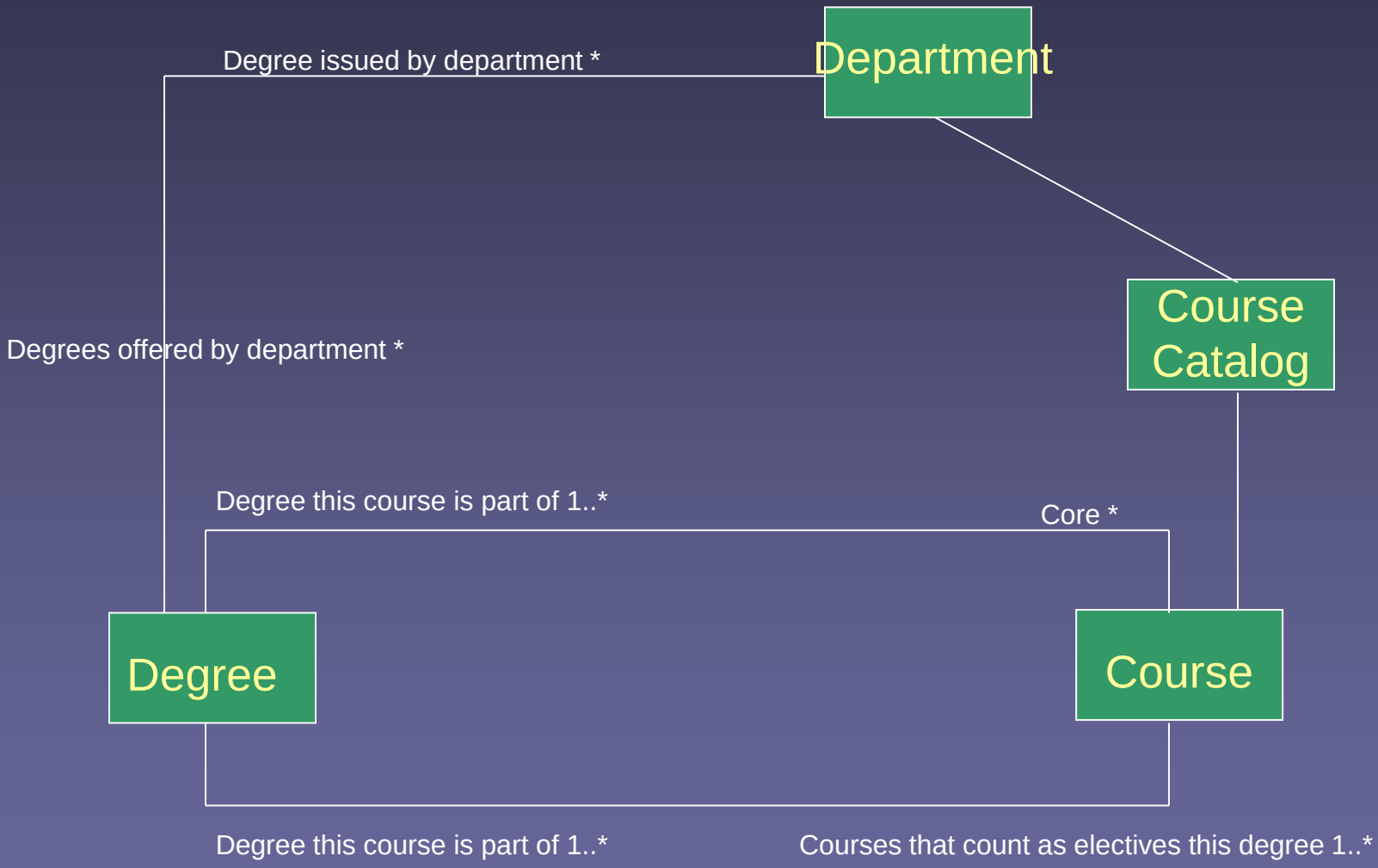


What is a course offering

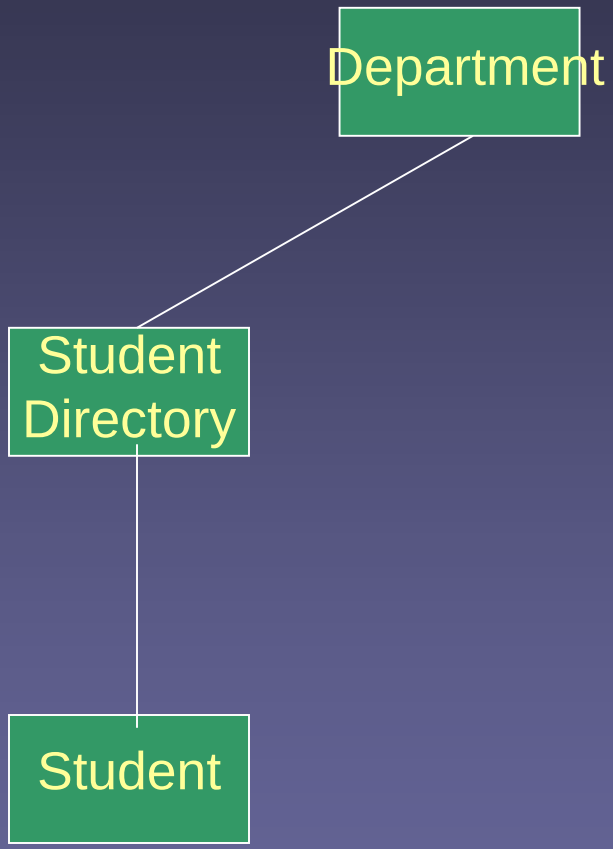


Courses students must take to qualify for degree

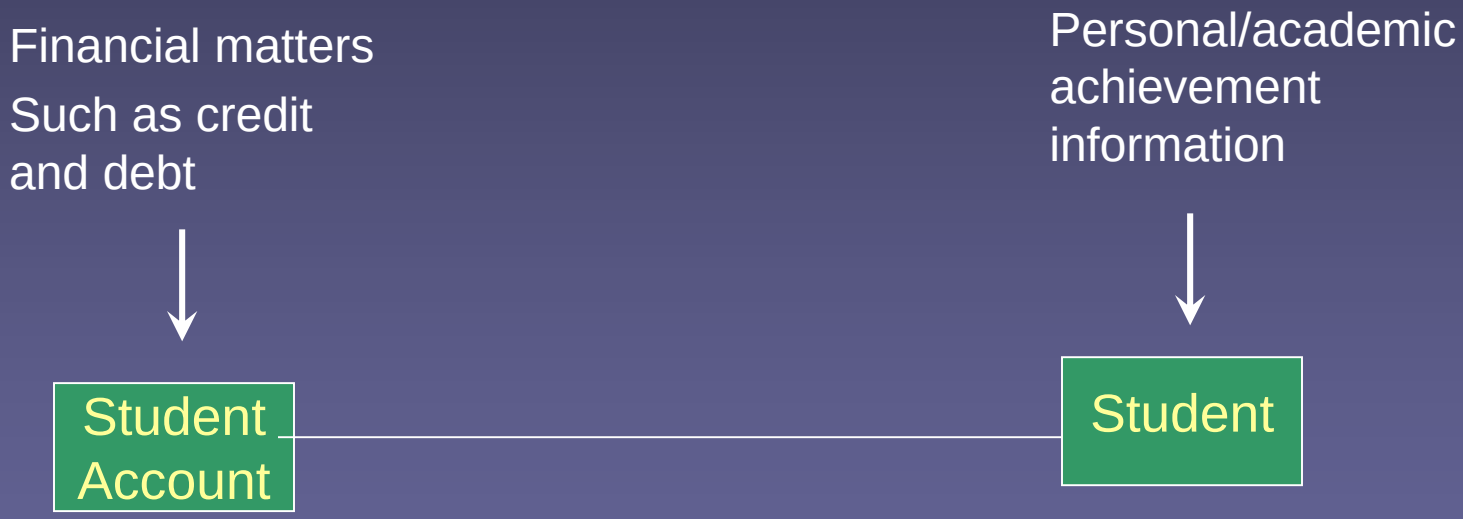
Department-Course-Degree Connection



The Student Dimension



Split the student into two concepts



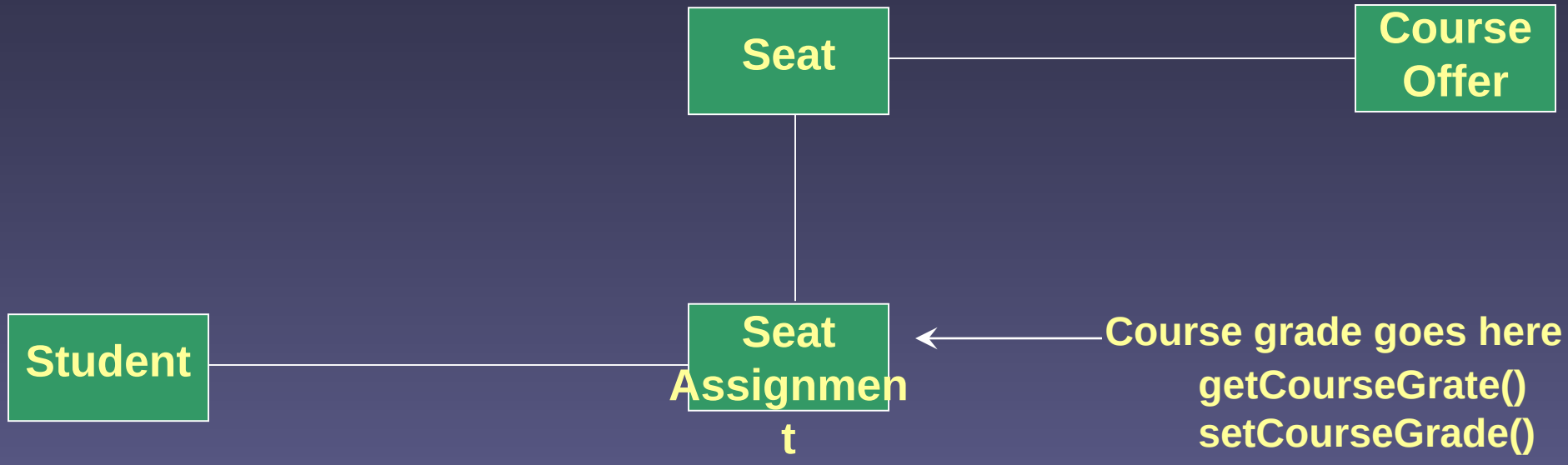
Assign Operations

Student

Person authorized to get a college degree from the university
Has a record of all courses taken during their stay
Has a grade point average (GPA)
A customer

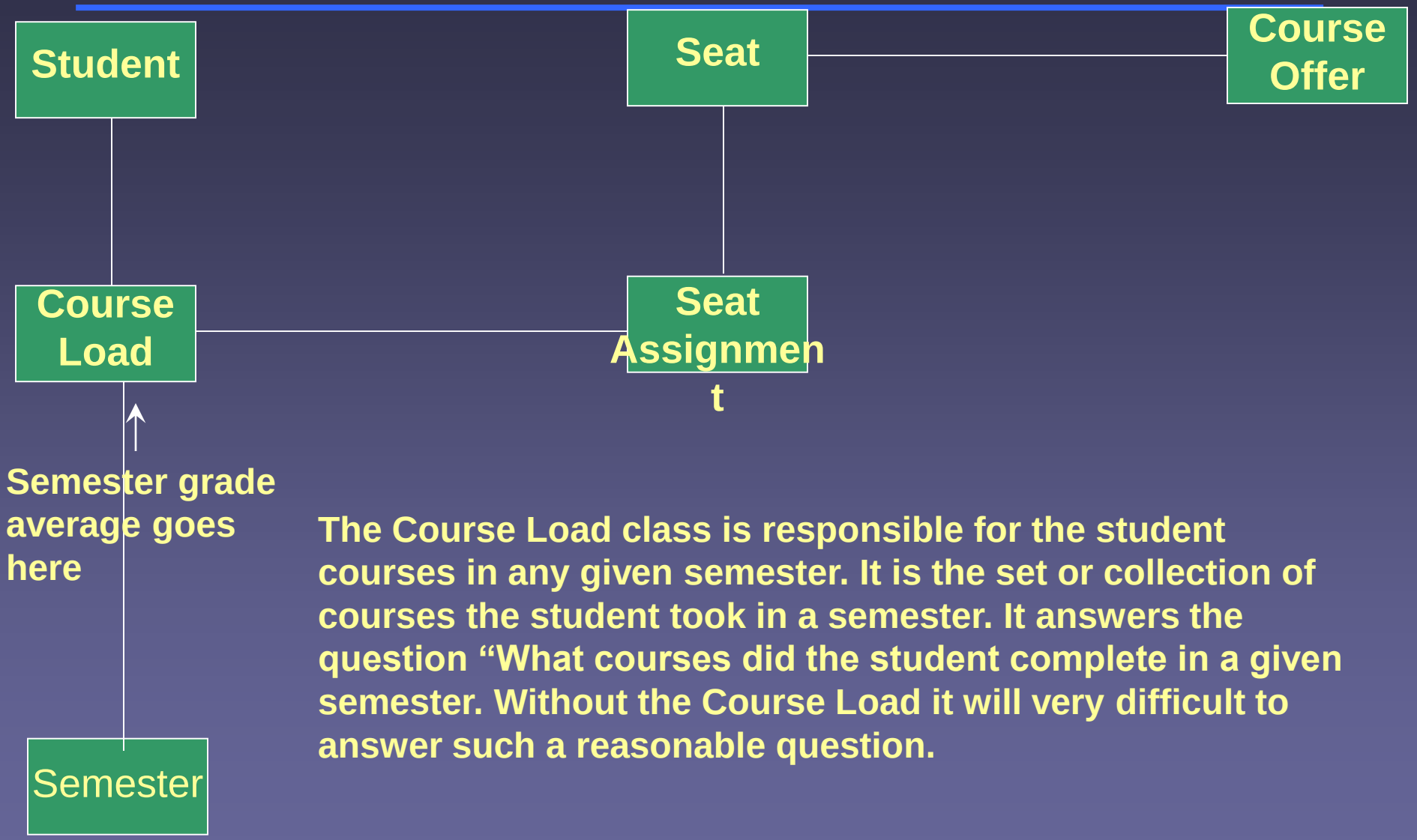
Operation Example:

```
Student.getRegisteredCourses():
{
set list of registered courses to empty
for each seat assigned to student invoke
    seat.getCourse() and add the course to the list of ;
    when done return the list of courses.
   }
```



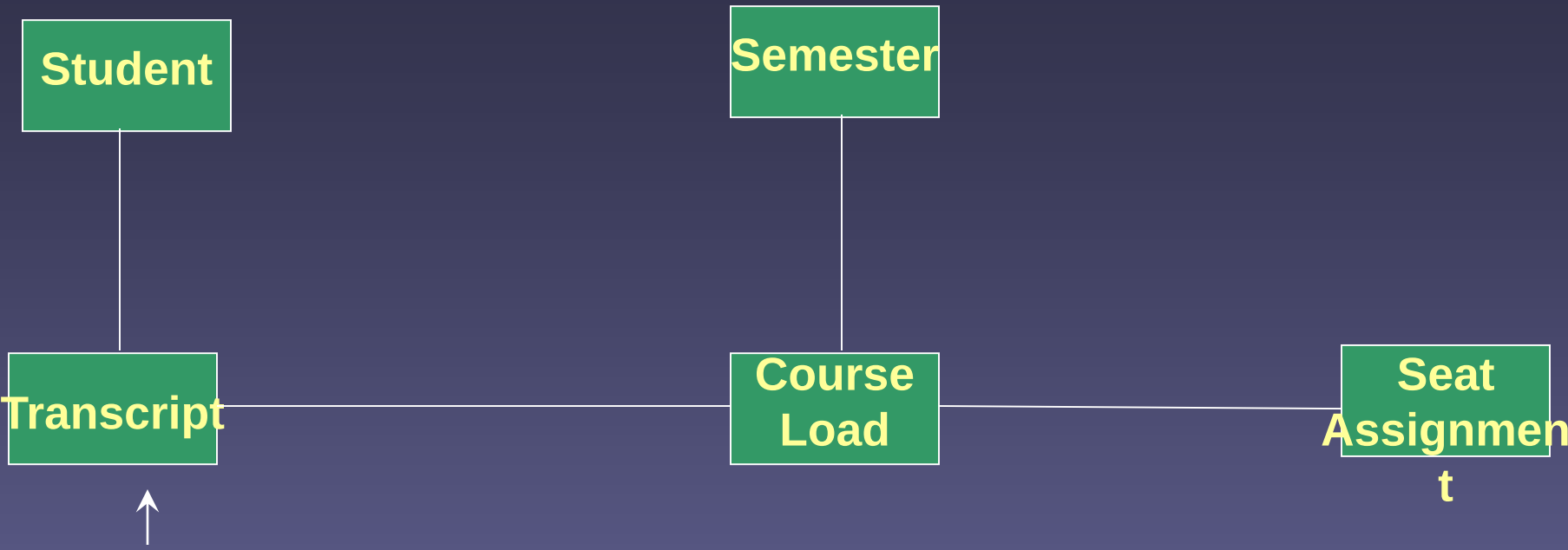
This model captures that fact the student tool a number of courses (two, three, four, etc). It also captures the grades the student got in each of these courses. It does not capture what courses the student signed up for in any given semester.

For example, what courses did “John the student” take in the spring semester? What is John’s grade average for the fall semester of 2004?



The Course Load class is responsible for the student courses in any given semester. It is the set or collection of courses the student took in a semester. It answers the question “What courses did the student complete in a given semester. Without the Course Load it will very difficult to answer such a reasonable question.

Notice there is no need to link the student to the course load anymore. It is redundant. Given a course load object it is easy to find the student (just follow the transcript then student relationships)

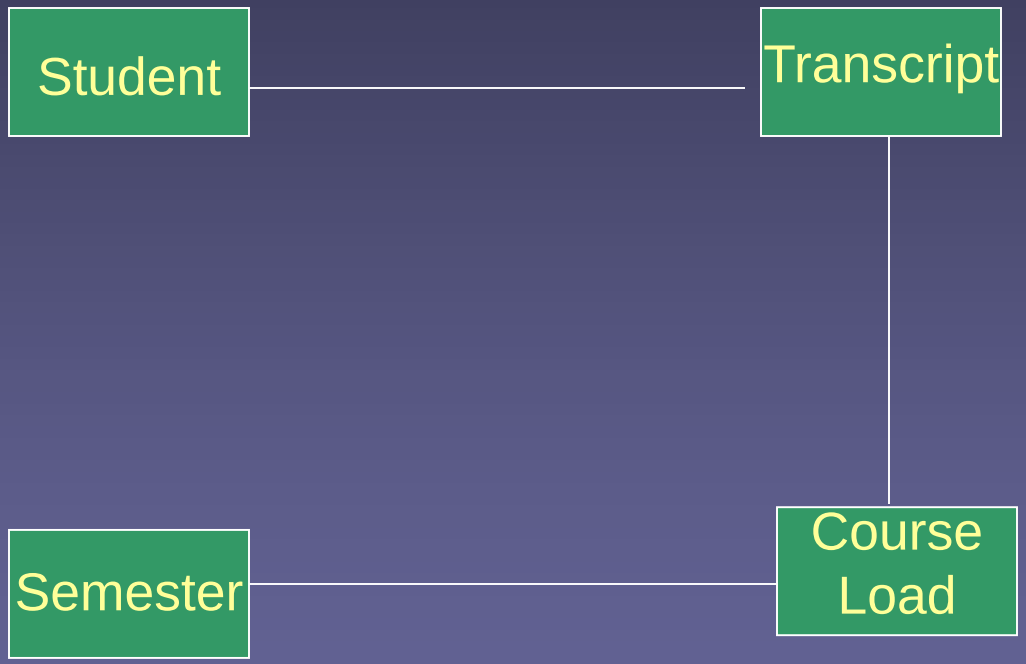


Student GPA goes here. It is a calculated attribute (operation).

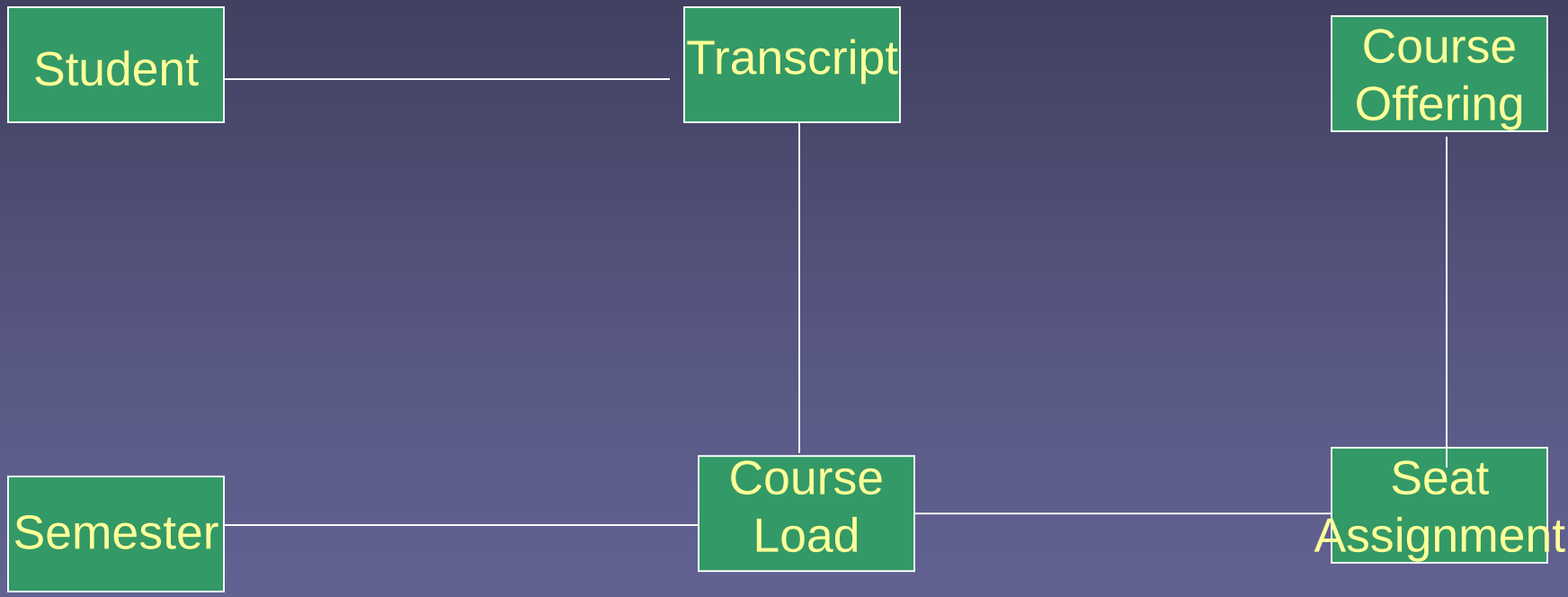
The Transcript class is a historical record of all the courses the student took during their stay at the university. The transcript class knows who the student is. It knows the student course load for any given semester.

To find the GPA. retrieve all the course load objects. For each course load object, get all the seat assignments. Iterate over all the seat assignments. For a given seat assignment get the course grade. Total them up grades. Take the average.

What is a transcript then?



What is a transcript then?

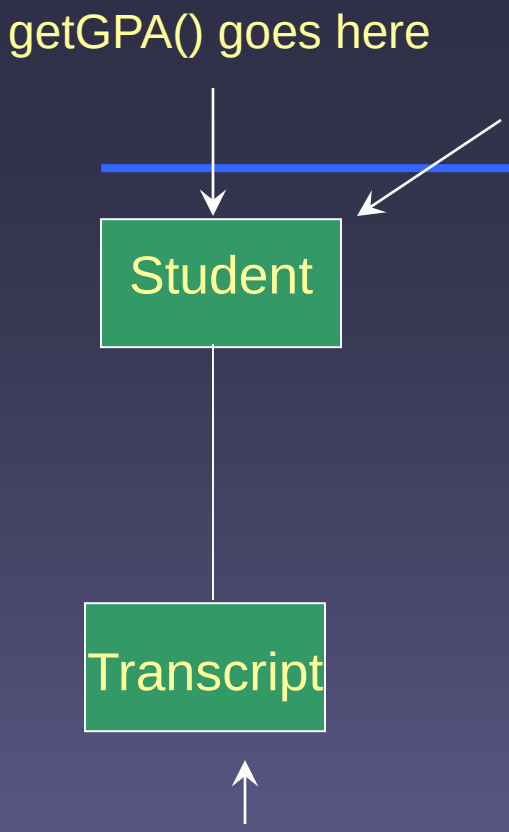


Transcript

Student GPA goes here. It is a calculated attribute (operation).

The Transcript class is a historical record of all the courses the student took during their stay at the university. The transcript class knows who the student is. It knows the student course load for any given semester.

To find the GPA. retrieve all the course load objects. For each course load object, get all the seat assignments. Iterate over all the seat assignments. For a given seat assignment get the course grade. Total them up grades. Take the average.



The question is usually something like what is Jim's GPA?

Therefore we need to define an operation of the student class tha returns the student GPA. We need to define getGPA() operation o the student class.

The student class has a reference to the transcript instance. The getGPA() operation on the student class calls the operation

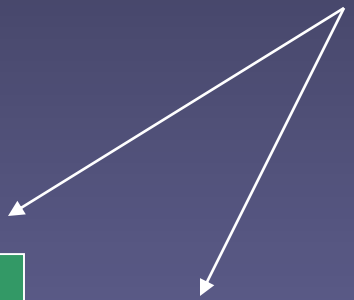
transcript.CalculateGPA() which calculates the student GPA when requested. The logic is as follows:

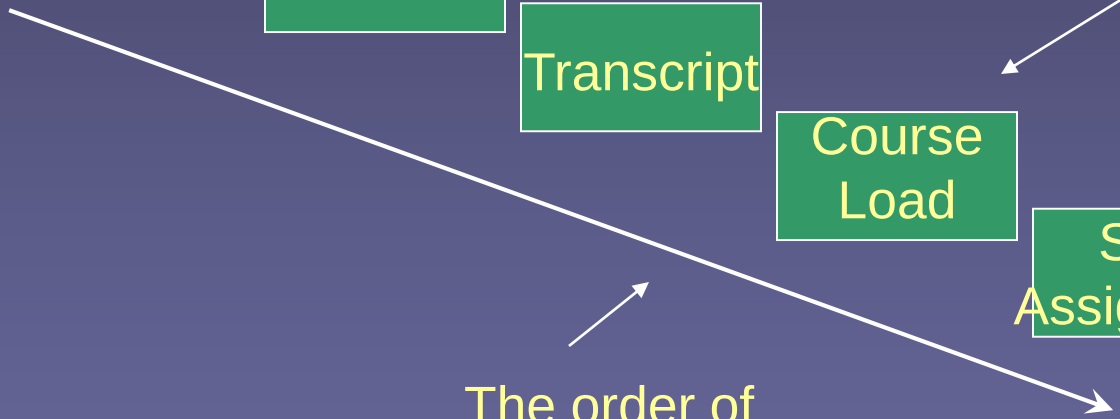
The CalculateGPA() operation retrieves all the course load objects. For each course load object, get all the seat assignments. Iterate over all the seat assignments. For a given seat assignment get the course grade. Total them up grades. Take the average.

How to determine student GPA?

Start her (find student)

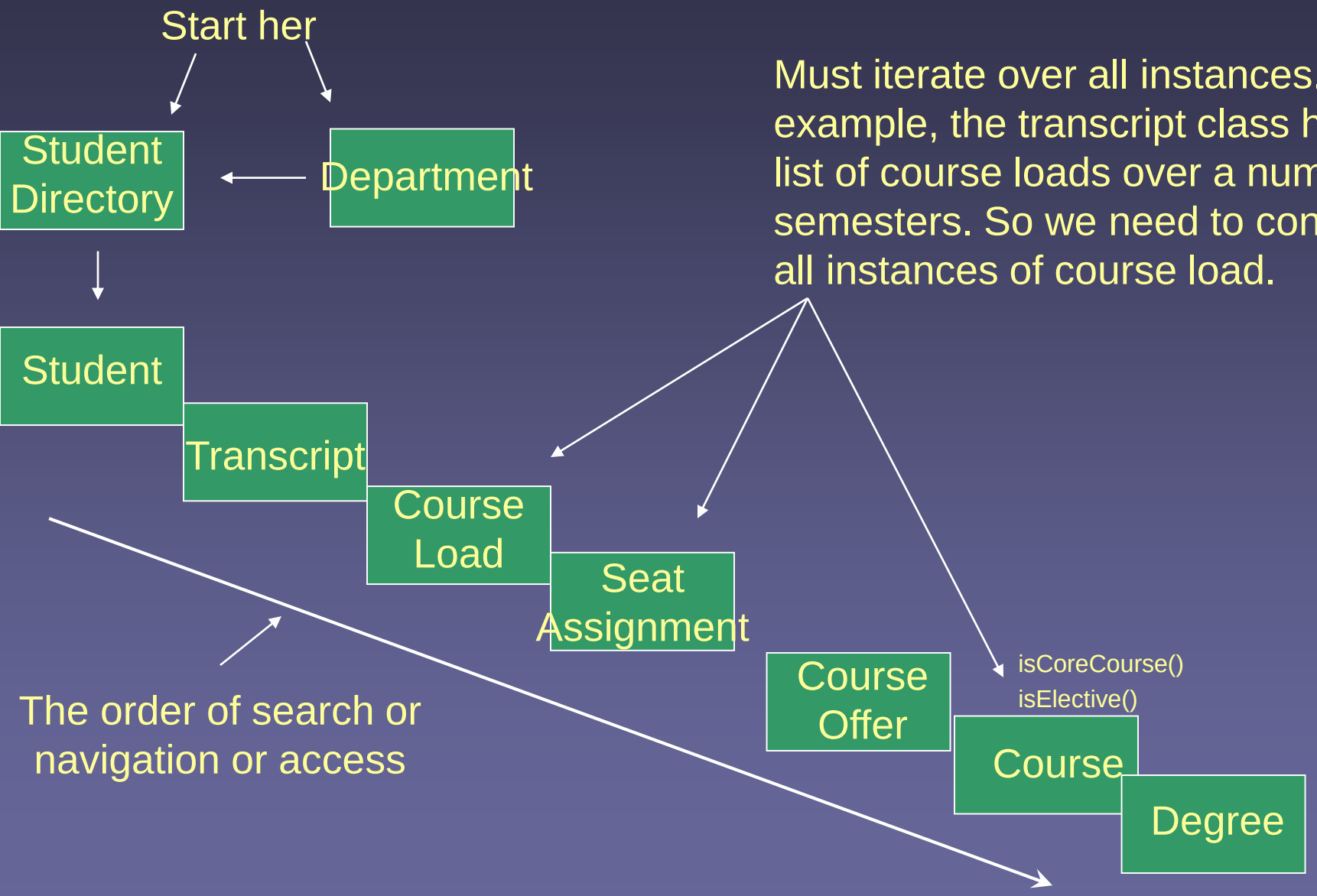
Must iterate over all instances. For example, the transcript class has a list of course loads over a number of semesters. So we need to consider all instances of course load.



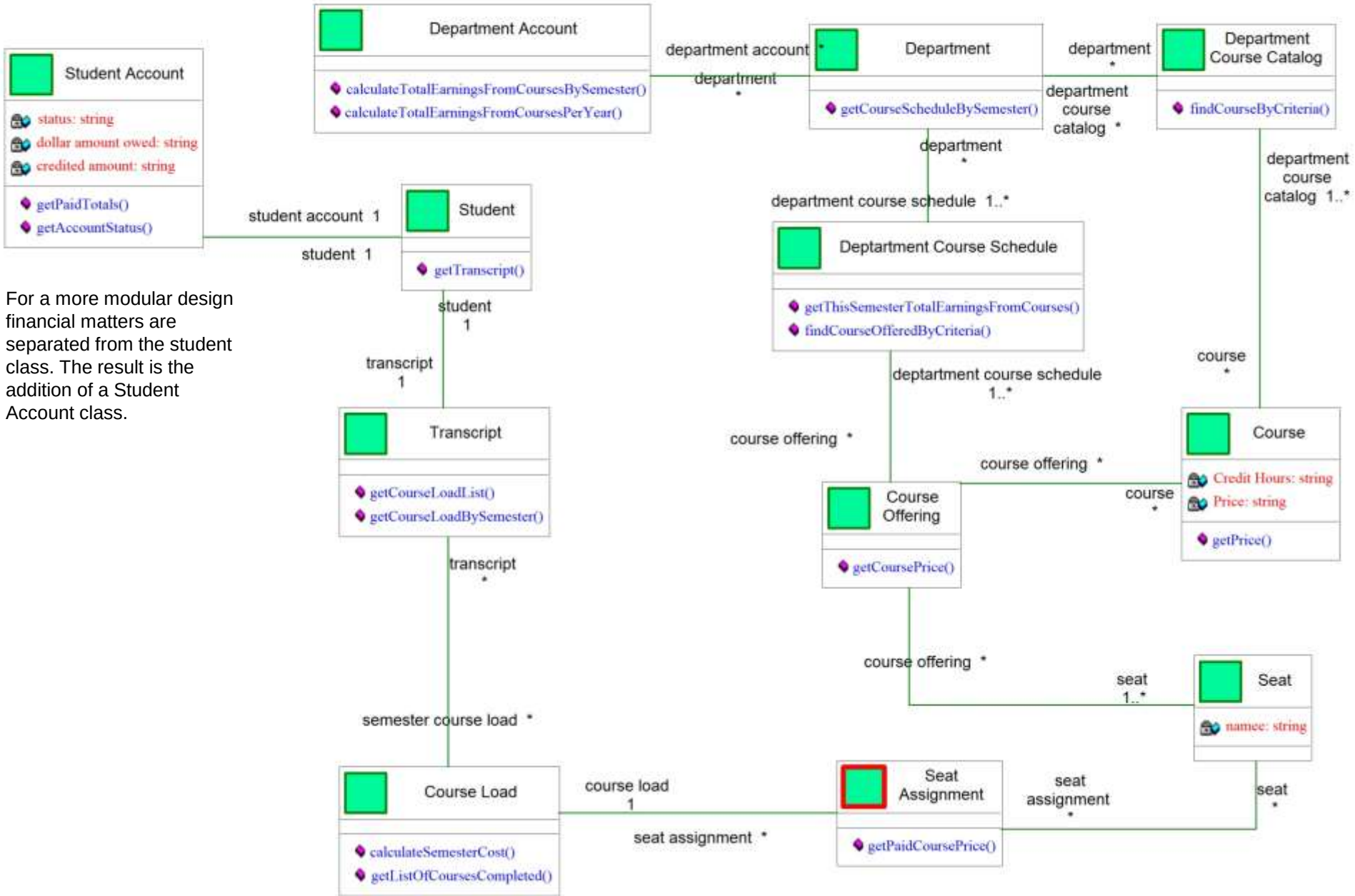


The order of search or navigation

How to determine if a student fulfilled the degree requirements?

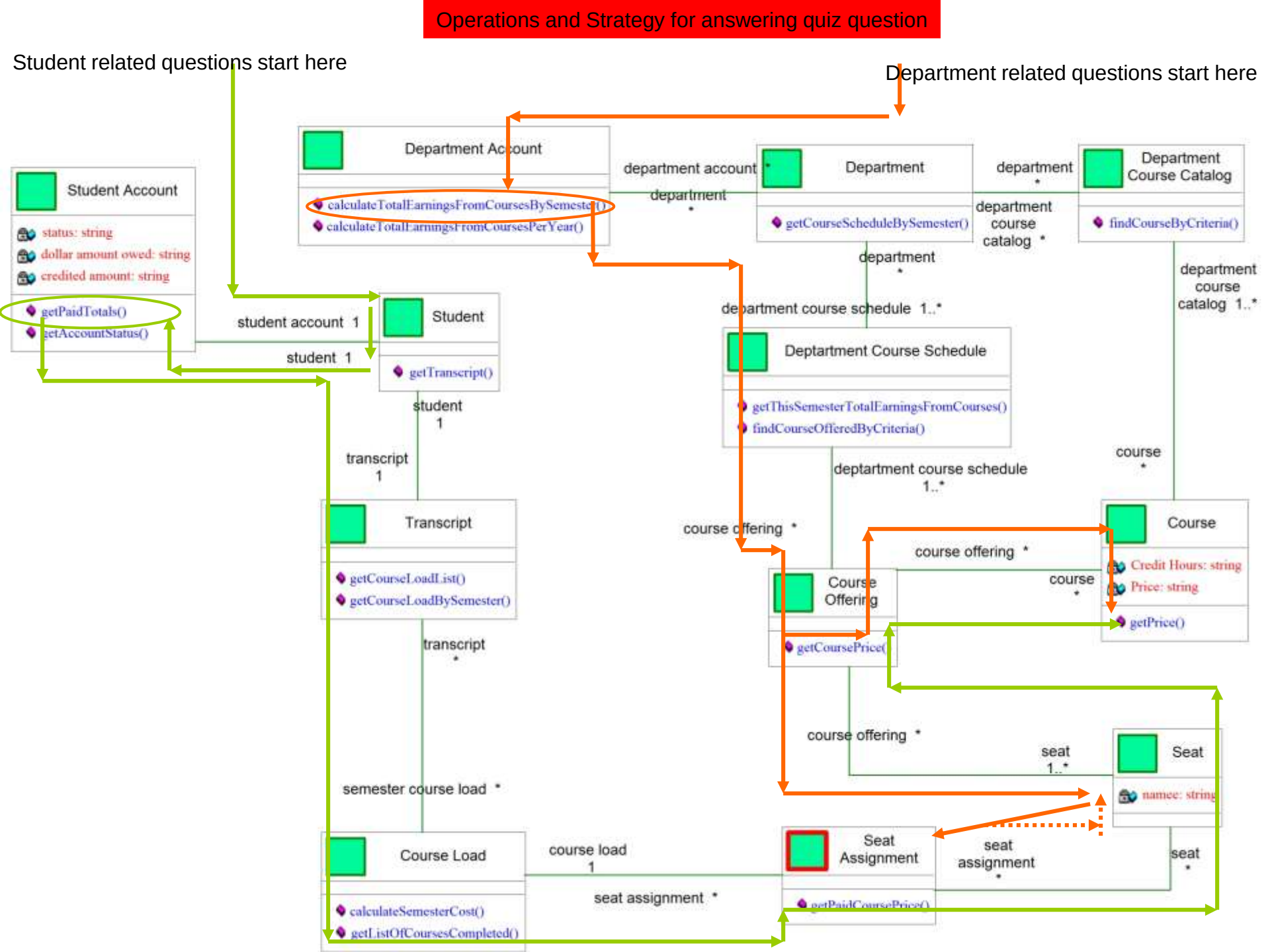


We extracted the following classes from the original model.
The focus is on classes to do with revenues.



For a more modular design financial matters are separated from the student class. The result is the addition of a Student Account class.

Operations and Strategy for answering quiz question



University:

Responsible for

academic matters such as colleges and students

Business matters such as marketing, financials, course pricing, etc.

Administrative matters such as Human Resources

Issues important at this level

Are we profitable?

What is the faculty/student ratio per class broken down by college? What is the current student enrollment in our university broken down by college?

College:

Responsible for

Academic matters such as departments and students

Business matters such as staff, faculty, etc.

Issues important at this level

What is the faculty/student ratio per class broken down by Department? How do we compare with other colleges?

What is the average number of students per class? Largest class? Smallest class?

What is the current student enrollment in our college broken down by department?

What is the administrative staff to faculty ratio?

What is the ratio of full time faculty vs part-time?

What is the percentage of faculty with Ph.Ds?