

Software Architecture

**Object Oriented
Analysis & Design**

***Ron Kleinman
Adjunct Professor
De Anza College***



Goal

This is the first of two presentations designed to introduce you to **how a software architect thinks**.

It covers how to begin analyzing requirements and designing the type of complex software applications that combine to run a large commercial enterprise like United Airlines, Walmart, FedEx, Disney ... or (in this case) a typical College.





A useful Analogy: Architecting a House

Home Buyer's Functional Specification

- I want a single level, three-bedroom, 2-bathroom home of at least 2000 square feet, with a large living room for entertaining, and an attached 2 car garage.
- The master bedroom suite should contain a private bathroom with a large whirlpool path, twin sinks and a walk-in closet. The other bathroom should be accessible from all rooms.
- My home should feature solar heating, and support the latest internet functionality, so I can install intelligent devices.
- A security system should be built in

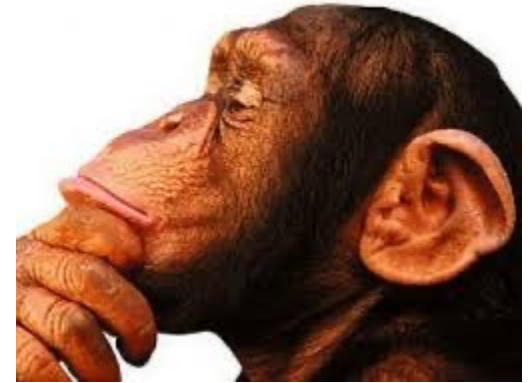
You are a Building Architect. You must build that house.

Where do you begin?

WRONG!!!



So what does a Building Architect actually DO when starting the design of a new building?



Building

Customer

**Planning
Documents**

**Physical
System Decisions**

**Physical
Installation**

A New Home

**Buyer
(Funct. Specs)**

**Architect
(Blueprints)**

**General
Contractor
(Piping, Wiring)**

**Skilled
Worker
(Plumber, Electrician)**

Solidify the Functional Specification!

- I want a single level, three-bedroom, 2-bathroom home of at least 2000 square feet, with a large living room for entertaining, and an attached 2 car garage.
- The master bedroom suite should contain a private bathroom with a large whirlpool path, twin sinks and a walk-in closet. The other bathroom should be accessible from all rooms.
- My home should feature solar heating, and support the latest internet functionality, so I can install intelligent devices.
- A security system should be built in

You are an Architect. You must build that house.

What do you do now?



Building Architect Deliverables

- **Identify all Components – ask questions!**
 - Shouldn't the house have ... a kitchen?
 - What about Closet storage? / A Laundry room?
- **Define interactions between components**
 - Relative sizes of bedrooms? Locations of each room?
- **Explore how high-level requirements can be met**
 - Security System details (electronic, locks, window bars)
 - Capacity of water heater, piping and wiring
 - Zoning: Must fit in with other existing structures
- **Identify Tradeoffs**
 - Construction materials vs. cost
 - Customized feature requirements vs. move-in date
- **Detail each component**
 - Exact size, material, layout...

So what does a Software Architect actually DO when starting development of a new software application?



Building

Customer

Planning
Documents

Physical
System Decisions

Physical
Installation

A New Home

Buyer
(Funct. Specs)

Architect
(Blueprints)

General
Contractor
(Piping, Wiring)

Skilled
Worker
(Plumber, Electrician)

A New Software Application

Domain
Expert
(Funct. Specs)

**Software
Architect**
(UML Artifacts)

Software
Architect/Designer
(Data Store, Security)

Software
Developer/Programmer
(Code / Debug)

Taking the first Step on the road ahead



FS: College Enrollment System

De Anza courses are offered by its departments (Ex: CIS) and available Quarterly (Ex: Fall and Winter). Each course has an identifying number (ex: 28), a Name (Ex: Object Oriented Analysis and Design), a description, a number of credits and an optional set of required prerequisites.

Each course is assigned a set of times during the week when it meets. It is also assigned a teacher who is both qualified and willing to teach that course, and who is free during the assigned times the course meets. It is also assigned a room, which also must be free during those times.

Students may then attempt to enroll in a course if they are paid up, have taken all its required prerequisites, are not already enrolled in the course, and if they are not already taking another course which overlaps the assigned times for this one.

Depending upon the size of the course and the number of students already enrolled, the student's enrollment request may either be accepted, or the student may be wait listed, or the request may be denied. If the student is accepted, her attendance will be tracked and at the end of the academic quarter she will receive a final grade.

You must produce a working solution. Where do you begin?



WRONG!!!

```
public function add(expression, replacement = "") {
    // count the number of sub-expressions
    // - add one because each pattern is itself a sub-expression
    $length = 1 + preg_match_all($this->SUB_EXPRS, $this->_internalEscape($expression), $matches);

    // treat only strings $replacement
    if (is_string($replacement)) {
        // does the pattern deal with sub-expressions?
        if (preg_match($this->SUB_REPLACE, $replacement)) {
            // a simple lookup? (e.g. "$2")
            if (preg_match($this->INDEXED, $replacement)) {
                // store the index (used for fast retrieval of matched strings)
                $replacement = (int)(substr($replacement, 1)) - 1;
            } else { // a complicated lookup (e.g. "Hello $2 $1")
                // build a function to do the lookup
                $quote = preg_match($this->QUOTE, $this->_internalEscape($replacement))
                    ? '"' : "'";
                $replacement = array(
                    'fn' => '_backReferences',
                    'data' => array(
                        'replacement' => $replacement,
                        'length' => $length,
                        'quote' => $quote
                    )
                );
            }
        }
    }

    // modified arguments
    return $this->_add($expression, $replacement, $length);
}
```

Software Architect Responsibilities

- **Identify all Components – ask questions!**
 - Read, understand and clarify Functional Specification
- ➔ *Direct interaction with the “Domain Expert” to identify and document the problem (requirements & constraints) that must be solved by the new system.*

Software Architect Responsibilities

- **Identify all Components – ask questions!**
 - Read, understand and clarify Functional Specification
 - Identify basic “abstractions” referenced in the FS
- *What are the words in the FS which must be further refined to determine exactly what the proposed system is required to do?*
- *Requires direct interaction with the “Domain Expert” to identify and document the problem (requirements & constraints) that must be solved by the new system.*

FS: Enrollment System at De Anza College

De Anza **College Courses** are offered by its **Departments** (Ex: CIS) and offered Quarterly (Ex: Fall and Winter). Each course has an identifying number (ex: 28), a Name (Ex: Object Oriented Analysis and Design), a description, a number of credits and an optional set of required prerequisite courses.

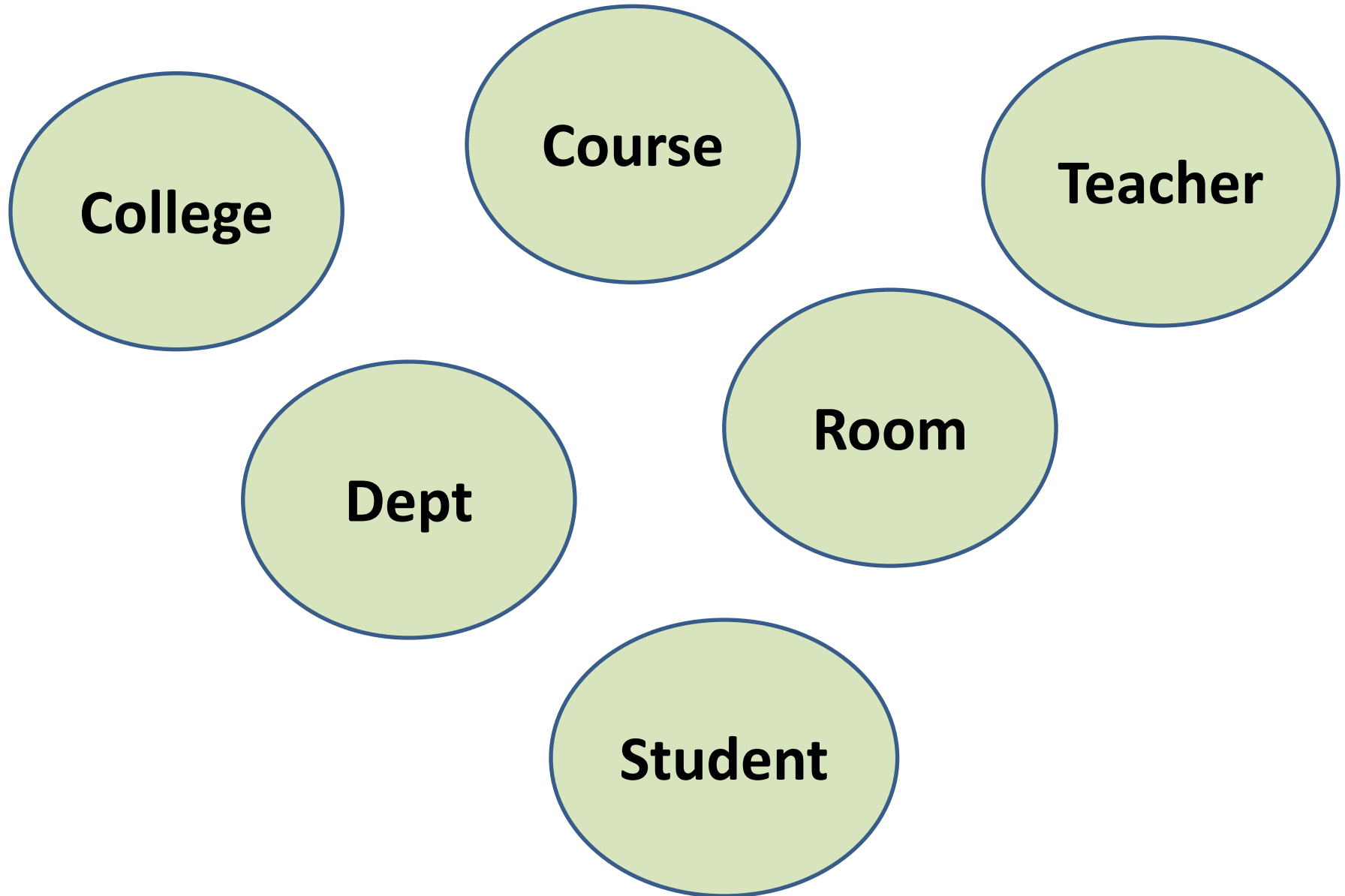
Each course is assigned a set of times during the week when it meets. It is also assigned a **Teacher** who is both qualified and willing to teach that course, and who is free during the assigned times the course meets. It is also assigned a **Room**, which also must be free during those times.

Students may then attempt to enroll in a course if they are paid up, have taken all its required prerequisites, are not already enrolled in the course, and if they are not already taking another course which overlaps the assigned times for this one.

Depending upon the size of the course and the number of students already enrolled, the student's enrollment request may either be accepted, or the student may be wait listed, or the request may be denied. If the student is accepted, her attendance will be tracked and at the end of the academic quarter she will receive a final grade for the course.

You must produce a working solution. Where do you begin?

Basic Abstractions in Enrollment System



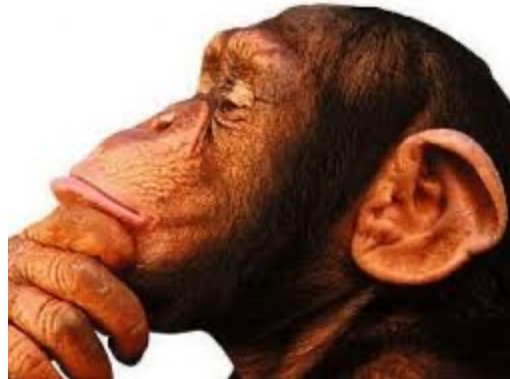
Refining the Abstractions (1)

- Can a Room / Teacher be assigned to a Course? **



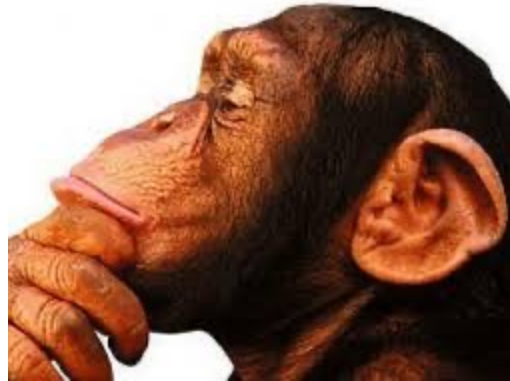
Refining the Abstractions (2)

- **Can a Room / Teacher be assigned to a Course?**
 - Course might be taught in several rooms
 - Course might be taught by several teachers
- **Can a Student “enroll” in a Course? ****



Refining the Abstractions (3)

- **Can a Room / Teacher be assigned to a Course?**
 - Course might be taught in several rooms
 - Course might be taught by several teachers
- **Can a Student “enroll” in a Course?**



No. We need a new abstraction



Refining the Abstractions

- **Can a Room / Teacher be assigned to a Course?**
 - Course might be taught in several rooms
 - Course might be taught by several teachers
- **Can a Student “enroll” in a Course?**
 - No. Need a new abstraction
- **Section vs. Course**
 - **Section** has Room, Teacher, Students, Hours to meet
 - **Course** has description, Name, # Credits, textbook, ...
 - **Relationship:** A Section “instantiates” a Course

Refining the Abstractions

- Teachers, Rooms, Students, all must detect scheduling conflicts when assigned a new Section?
- How can we avoid writing duplicate code?

A. Write “*schedule conflict detection code*” for one of them, cut and paste it into the other 2.

B. Create a new object containing “*schedule conflict detection code*” which is contained in all 3.

C. Create a new object containing “*schedule conflict detection code*” which is used by all 3



Refining the Abstractions

→ Have each of them contain their own **Schedule** object.



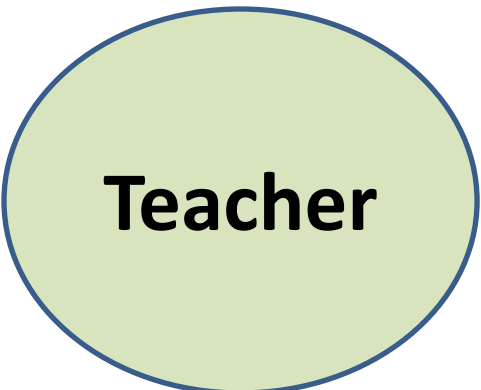
Basic Abstractions in Enrollment System



College



Dept



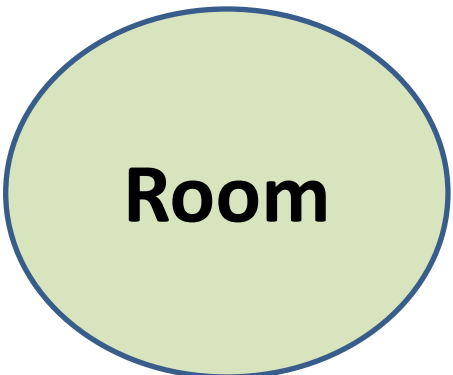
Teacher



Student



Course



Room

Refined Abstractions in Enrollment System

College

Dept

Teacher

Student

Course

Room

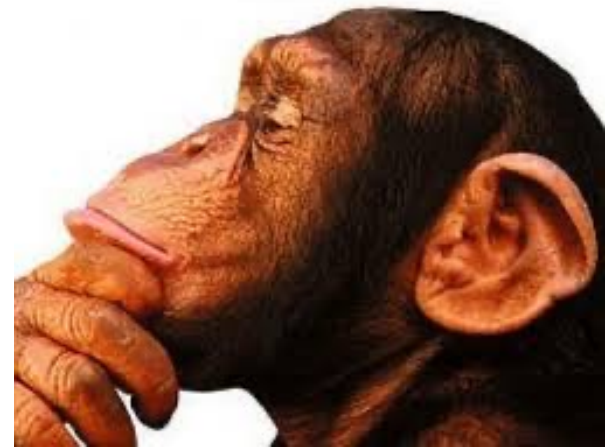
Schedule

Section

Re-Refining the Abstractions

Ex: What is a Prerequisite?

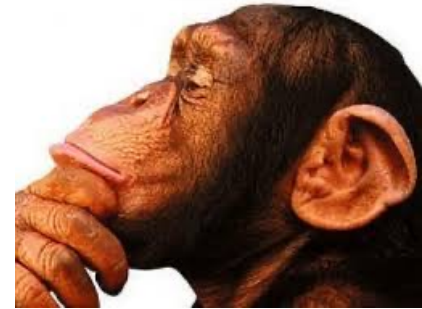
- A Course?
- A Section?
- A String (containing the Prereq Name & ID)?
- A different object entirely?



Re-Refining the Abstractions

A Prereq is a Course:

- Prereqs have Department & Course IDs (CIS 22a)
- A Course can have multiple Prereqs
- A Prereq can have multiple Prereqs



A Prereq is a Section

- ??? Any Section of a Course is the same Prereq

A Prereq Is a simple String

- What else is needed for the College catalog?

A Prereq is a new type of Object

- Why might a new object be needed?



What exactly is a Prereq?

Look at the catalog!

- Prereqs have “types” (*Advisory / Mandatory*)
- Prereqs may not refer to courses at all!!
 - Ex: “*Prior experience with OO language*”
- Prereqs don’t have Prereqs. Courses do.

→ A Prereq is an object optionally containing a Course ID (among other elements).

This is the sort of stuff a Software Architect THINKS about!

Re-refined Abstractions in the De Anza Enrollment System

College

Dept

Teacher

Student

Course

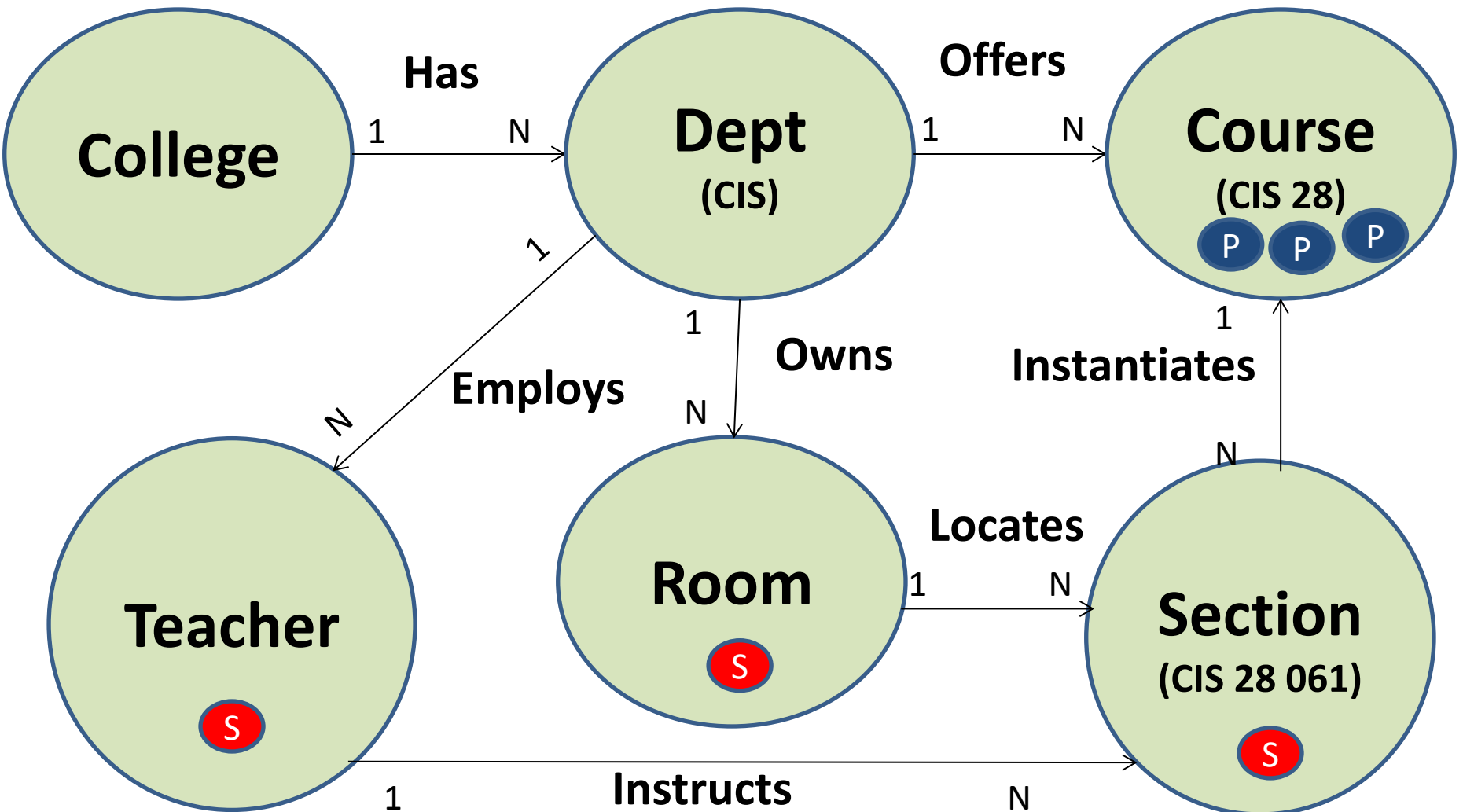
Room

Schedule

Section

Prereq

CCD: Schedule of Classes



Questions?



Software Architecture

**Object Oriented
Analysis & Design**

***Ron Kleinman
Adjunct Professor
De Anza College***



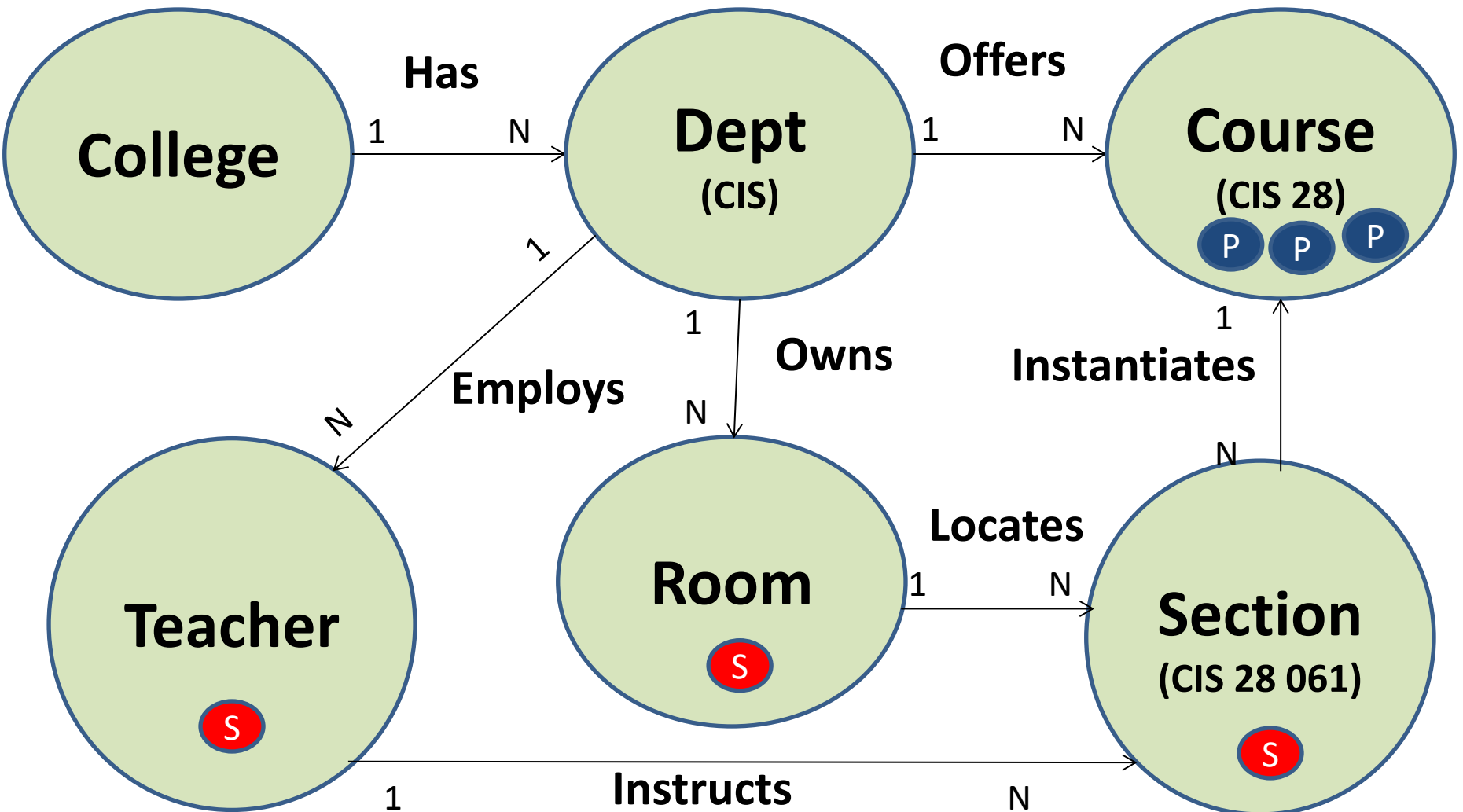
Goal

This is the second of two presentations designed to introduce you to **how a software architect thinks.**

It covers the creation of an application vocabulary and the use of Design Patterns.



CCD: Schedule of Classes



Object Relationships (#1)

- **1:1 (ex: Every Teacher has 1 Schedule)**

```
class Teacher {  
    private:  
        Schedule s;
```

- **1:N (ex: Every Teacher can teach many Sections
but a Section only has 1 Teacher)**

```
class Teacher {  
    private:  
        Section *sections[20];
```

Object Relationships (#2)

- **N:1 (ex: A Department has many Teachers, but a Teacher is only in one Department)**

```
class Teacher {  
    private:  
        Department *department;
```

- **N:N (ex: A Student may be enrolled in many Sections, and every Section can have many Students)**

Uhoh

Conceptual Class Diagram: Enrollment



How could we implement this? **



Conceptual Class Diagram: Enrollment

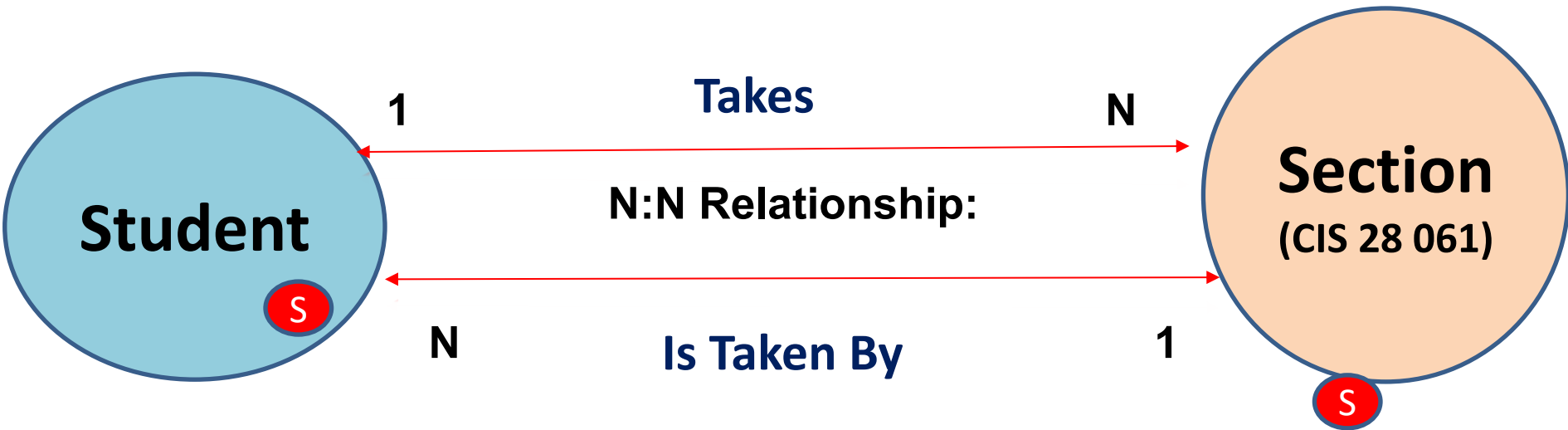
Try two separate
“Pointer Arrays”

Student has array
of Section Refs



Section has array
of Student Refs

Conceptual Class Diagram: Enrollment



But this N:N Relationship has its own Data!

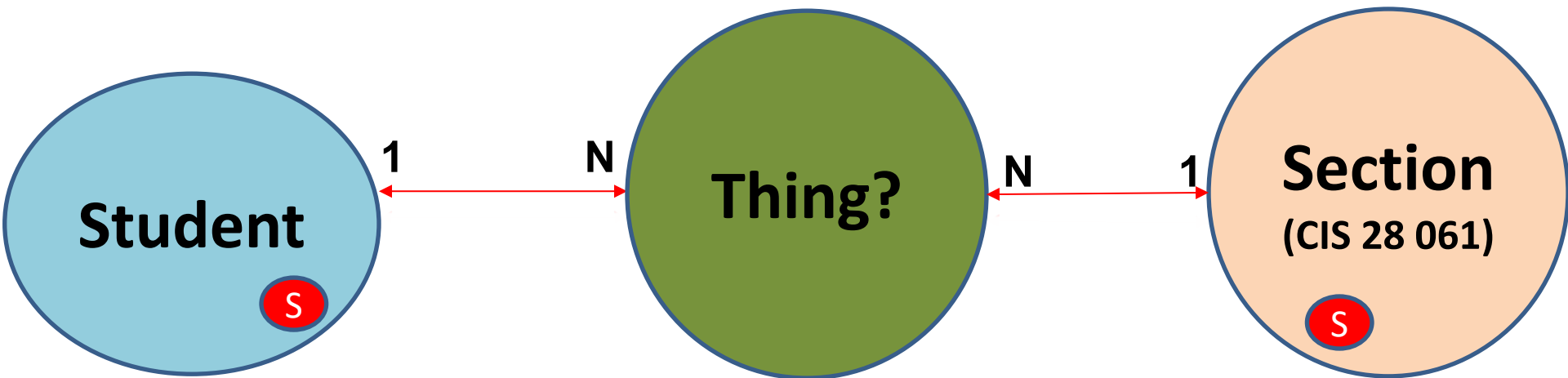
Final Grade
Attendance Records
Exam Grades
Teacher's Comments

→ Where is this info stored? ***



Note: N:N Relationship with data is usually a design error!

Conceptual Class Diagram: Enrollment



Design Pattern: “Junction Class”

Final Grade

Attendance Records

Exam Grades

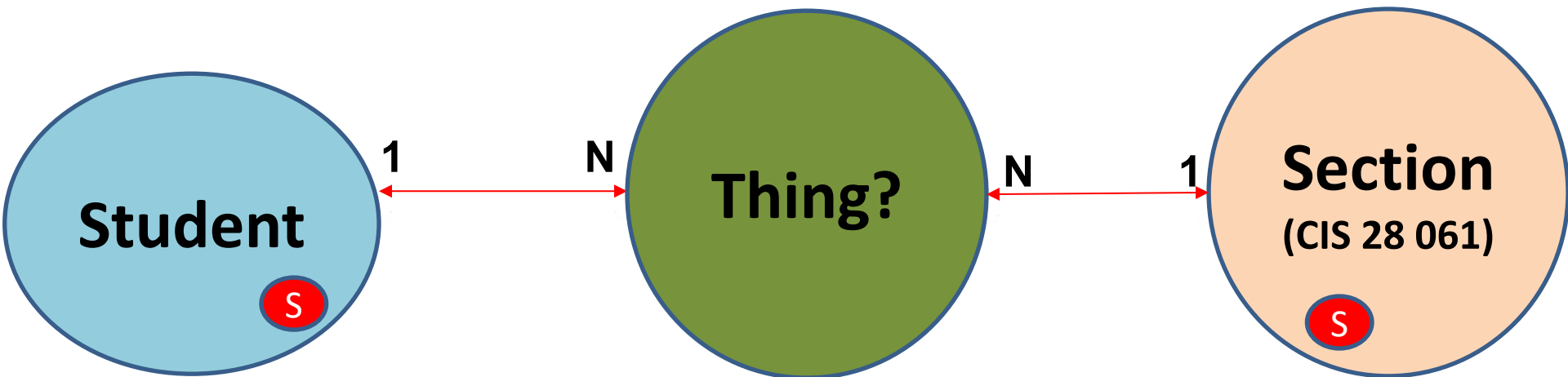
Teacher’s Comments

+

Student Reference / Pointer

Section Reference / Pointer

Conceptual Class Diagram: Enrollment



Design Pattern: “Junction Class”

Student Reference

Section Reference

Final Grade

Attendance Records

Exam Grades

Teacher’s Comments

So ... what is a “Thing”??

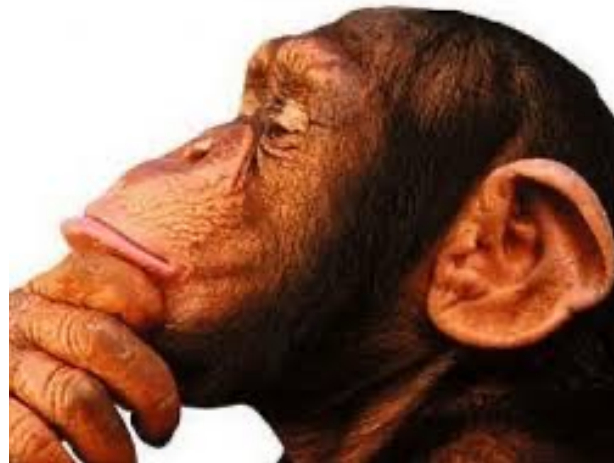
Schedule? Transcript? Roster? Registration?



But what exactly is a “Thing”?? **

What “binds” one Student to one Section?

- Student has multiple Things, one in each Section
- Section initially allocated with 55 Things (15 WL)
- Each Thing is either “Occupied” or “Empty”
- The Thing contains all Student info for that Section
 - Grade
 - Attendance Record
 - Teachers Comments
 - Final Exam Scores
 - Teachers Notes



What “binds” a Student to a Section?

- **An Enrollment (Good)**

- Student has multiple Enrollments, one in each Section
- Section initially allocated with 55 Enrollments (15 are WL)
- Each Enrollment is either “Occupied” or “Empty”

What “binds” a Student to a Section?

- **An “Enrollment” (Good)**

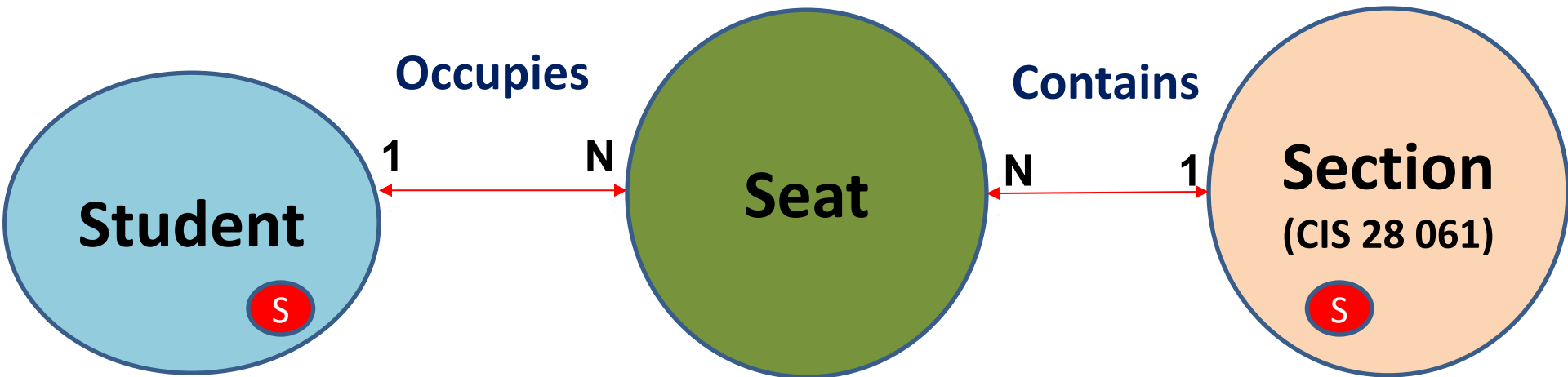
- Student has multiple Enrollments, one in each Section
- Section initially allocated with 55 Enrollments (15 are WL)
- Each Enrollment is either “Occupied” or “Empty”

- **A “Seat” (Better)**

- Student has multiple Seats, one in each Section
- Section initially allocated with 55 Seats (15 are WL)
- Each Seat is either “Occupied” or “Empty”

→ **Note lack of “physicality” + no “Seat ID” here**

Conceptual Class Diagram: Enrollment



Student has array
of Seat Refs

Student Reference
Section Reference

Section has array
of Seat Refs

Final Grade
Attendance Records
Exam Grades
Teacher's Comments

Original Abstractions in Enrollment System

College

Dept

Teacher

Student

Course

Room

Schedule

Section

Prereq

Refined Abstractions in Enrollment System

College

Dept

Teacher

Student

Course

Room

Schedule

Section

Prereq

Seat

FS: Enrollment System at De Anza College

In De Anza **College**, **Courses** are offered by its **Departments** (Ex: CIS) and available Quarterly (Ex: Fall and Winter). Each Course has an identifying number (ex: 28), a Name (Ex: Object Oriented Analysis and Design), a description, a number of credits and an optional set of required prerequisites.

Each offered Course is “instantiated” by one or more “**Sections**”. Each Section is assigned a Section number (ex: 061), and a **Schedule** which defines the set of times during the week when it meets. It is also assigned a **Teacher** who is both qualified and willing to teach that course, and who has no Schedule conflicts with it. It is also assigned a **Room**, which also must be free during those times the Section meets.

Students may then attempt to enroll in a Section if they are paid up, have taken all its required **Prerequisites**, are not currently enrolled in any Section of the course, and if they are not already enrolled in another Section with a conflicting Schedule .

Depending upon the size of the Section and the number of Students already enrolled, the Student’s enrollment request may either be accepted, or the Student may be wait listed, or the request may be denied. If the Student is accepted, she will be assigned a **Seat**, her attendance will be tracked and at the end of the academic quarter she will receive a final grade for the Section.

Common Application-specific Vocabulary

(created by Software Architect, spoken by everyone)

Domain
Expert

Software
Architect

Programmer

<Noun = Class / Verb = “method” / Adjective = “type of”>

Ex: *“A foreign student enrolls in a virtual section”*

- *“Student”* is a class
- *“Foreign”* is a type of *Student* (child class?)
- *“Section”* is a class
- *“Virtual”* is a type of *Section* (child class?)
- *“Enroll”* is a public method of either:
 - *Student:: enroll (Section&)*
 - *Section:: enroll (Student&)*

Design Patterns

- Provide solutions to **software design problems** that pop up again and again in real-world application development (like “N:N relationships”).
- Formally document these designs so they can be reused (in the same way Object Libraries allow code to be reused).
- Design Patterns (Elements of Reusable OO Software)
 - Gamma, Helm, Johnson, Vlissides (Gang of 4)

<http://www.uml.org.cn/c++/pdf/DesignPatterns.pdf>

Ex: Policy Customization

We hope to resell our Enrollment System to other city and county Colleges, educational training departments and educational institutions. Each of these will have different policies regarding things like:

- Maximum number of credits a Student can take in a given term
- Minimum number of office hours made available to Students in a Section
- Minimum number of enrolled Students a Section needs for it not to be cancelled.

So how do we avoid / minimize “recoding” parts of our application every time we make a new sale, to reflect the set of policies in place within the new customer’s organization?

(Note: Eliminating ALL such policy-based coding changes pays huge dividends. We can then have a common source code tree for all our customers, greatly reducing costs concerned with adding new features and issuing patches, because we would only have to “stage” one version of the code for testing the effects of any new code change across our entire customer base.)

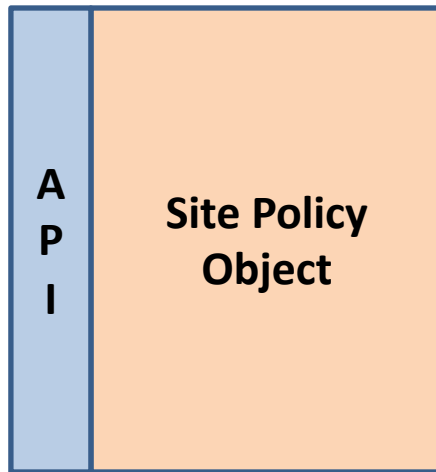


Fixed Property Design Pattern

- **Name:** Fixed Property
- **Classification:** Structural
- **Intent:** Minimize changes by encapsulating Site Policies
- **Applicability (use when)**
 - Many business constants are site specific
 - Many business constants actually change on a regular basis
- **Participants / Collaborations:** N/A
- **Implementation Guide**
 - Single “value = get (key)” method
- **Known Uses**
 - Reduces support costs for multi-site application software

Encapsulate the Site Policy: “Fixed Property” Design Pattern

<http://martinfowler.com/apSUPP/properties.pdf>



On Disk Site Configuration File

Key (<i>PolicyName</i> String)	Value (Site Specific)
MaxDaysOnLoan	10
FinePerDay	25
LibCardExpireMnth	18
SiteLicense	X31PGJ6

The Site Policy Object has a Hash Table API with only a single method:

```
String Value = Policy::get(String Key);
```

Any LIS28 object relying on a site-specific policy value may get that value by going to the Site Policy object, which reads and saves the contents of the Site Configuration File as part of its constructor logic.

➔ No code needs to change when application is run at a new site!

Can you
architect a
software
Application?

YOU CAN DO IT



YES YOU CAN

memegenerator.net

Questions?

