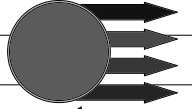


OO Design Process



• Assumes known requirements

- System operation contracts

• Define class/object details

- Assign responsibilities (patterns)

• Document

- Interactions (collaboration/sequence)
- Classes (design class diagram)

Document interactions & classes simultaneously.

Example: Use Case

Name: AddCourseSection
 (coursenum : CourseNumber)
 Responsibilities: Add empty section to a course.
 Exceptions: Invalid course number; error.
 Preconditions: Valid course number.
 Postconditions:
 • A *Section* was created.
 • *Section.section_number* was set to "next".
 • *Section.section_size* was set to zero.
 • The *Section* was associated with the *CourseInfo*.

Example: Classes Involved

• Schedule

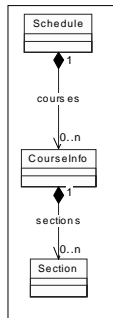
- Overall information

• Course

- Describes one course

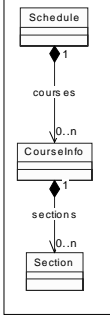
• Section

- One course section



Example: Choosing Controller

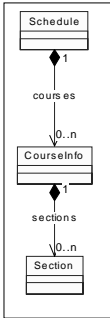
- **Schedule**
 - Entire “system”
 - Facade controller
- **DepartmentHead**
 - Real-world “thing”
- **NewSectionHandler**
 - Use-case controller



Which one is “best”?

Example: Creation Responsibility

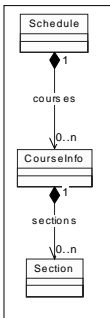
- **Schedule**
 - Receives request
- **Course**
 - Associated with Section
- **??**
 - Some other responsible object?



Who should create Section??

Example: Finding Course

- **Many courses in Schedule**
- **Must find right one to associate with new Section**
- **Who does lookup?**



...

Design Exercise

- **Faculty schedule system**
- **Pick one of these system events**
 - Assign section to professor
 - Calculate a professor's workload
- **Choose design and justify**
- **Document**
 - Collaboration diagram
