Binding Design

Lawrence Angrave
Binding

- Early vs Late Binding
Design
What's the difference between being good at "designing a system" and good at "coding"?
Student
- Name
- Address
- Phone Number
- Email Address
- Student Number
- Average Mark
- Is Eligible To Enroll
- Get Seminars Taken

Enrollment
- Marks Received
  - Get Average To Date
  - Get Final Mark

Seminar
- Name
- Seminar Number
- Fees
- Add Student
- Drop Student

Professor
- Name
- Address
- Phone Number
- Email Address
- Salary

- 0..1
- instructs

0..* 0..{* {ordered, FIFO} on waiting list

- 1 enrolled
- 1..* 1 in
- "Some seminars may not have an instructor?"
Class/Object Notation

- Class definitions

```
AbstractClassName

AbstractOperation1()
Type AbstractOperation2()
```

```
ConcreteClassName

Operation1()
Type Operation2()

instanceVariable1
Type instanceVariable2
```

- Abstract in italics
- Methods have parentheses
- Variables do not
- Types are optional; included when useful
Observer Pattern

- Observers can “attach” to a Subject.
- When Subject is updated, it calls Update() on all Observers.
- Observers can query Subject for updated state.
Strategy Pattern

- Strategy abstract base class exposes algorithm interface.
- Context object HasA Concrete Strategy object.
- Context object invokes algorithm interface from strategy.

![Diagram showing Strategy Pattern]

The diagram illustrates the Strategy Pattern with classes and interfaces:

- **Context** with a method `ContextInterface()`
- **Strategy** with an interface `AlgorithmInterface()`
- **ConcreteStrategyA**, **ConcreteStrategyB**, and **ConcreteStrategyC** each with an interface `AlgorithmInterface()`
Class/Object Notation (cont.)

• Class relationships

- Diamond = Has A collection of
- Solid dot = multiple
- Triangle = Inheritance (Is A)
- Dashed line = creates
- Solid line = Has A (containment)