


## Questions First

- n On class?
- n On project?
- n On material we've discussed?
- n Other?



## CSE403 Section 3:

### Best Practices for Software System Design (continued)


*"There are two ways of constructing a software design: one way is to make it so simple that there are obviously no deficiencies; the other is to make it so complicated that there are no obvious deficiencies." C.A.R. Hoare (1985)*

Valentin Razmov, CSE403, Sp05



## Design in a Nutshell

- n "Treat design as a wicked, sloppy, heuristic process. Don't settle for the first design that occurs to you. Collaborate. Strive for simplicity. Prototype when you need to. Iterate, iterate, and iterate again. You'll be happy with your designs."  
Steve McConnell, *Code Complete, 2nd ed.*, chap 5,  
<http://www.cc2e.com/docs/Chapter5-Design.pdf>




## Why Learn How to Design?



## Best Practices for Software Design

- n Favor composition over inheritance.
  - n Example:



## Principles for Good Design

(from Robert Martin's "Agile Software Development – Principles...")

- n **Open-Closed Principle**
  - n "Software entities (classes, modules, functions, etc.) should be open for extension but closed for modification."
  - n Example: An abstract class to extend (with as many new subclasses as needed) rather than modifying an existing class to accommodate each new addition.
- n The designer chooses what changes to anticipate.

## Principles for Good Design (cont.)

- Dependency Inversion Principle
  - (A) "High-level modules should not depend on low-level modules. Both should depend on abstractions."
  - (B) "Abstractions should not depend on details. Details should depend on abstractions."
  - Example: Separation of policy and mechanism

## Principles for Good Design (cont.)

- Single Responsibility Principle
  - "A class should have only one reason to change."
  - Example 1: Putting state in a GUI class.
    - Can be avoided using the Model-View-Controller pattern.
  - Example 2:

```
interface Modem {
    public void dial (String pno);
    public void hangup();
    public void send (char c);
    public char recv();
}
```

## Principles for Good Design (cont.)

- Interface Segregation Principle
  - "Clients should not be forced to depend on methods that they do not use."
  - Example: Dogs jump but don't sing.

## Examples of Bad Designs Practices

- "Design by committee"
  - Everyone on the committee puts in their favorite features into the soup. What is the result?
  - Moral: The design must be owned and managed by someone.
- Others you have heard of?

## One-minute Feedback

- What one or two ideas discussed today captured your attention and thinking the most?
- List any ideas / concepts that you would like to hear more about in this class. Be specific.