Today - Lecture 8 - CS202

1) Object Oriented Design
2) Example Design - Procodural vs. OO
3) Programming Examples
   - C#
   - Java

   } Evaluate designs

4) Evaluation OO Designs
   - participation for both in-class & online

Announcements:

Designing with Objects

1) objects are defined by What they can do not how they do it.

2) The data in an object is part of the "how".

3) objects are defined by the messages they receive & send and not by their data.

4) This is why we focus on developing classes that have a clearly defined job or purpose. They are not just a "gate keeper" for the data!

5) "Set and "Get" functions take that "job" away and place it back into the control of classes using this class' objects.

6) Emphasize — what are the capabilities of what an object can do NOT on how those capabilities are implemented. The data is irrelevant.

Rules of OOP

What ARE the "rules" of OO Programming?

Consider These ... Any Others? (JavaWorld)

1) "All data is Private. Period" ← protected ok?

2) "get and set functions are evil. (They're just elaborate ways to make the data public.)

3) "Never ask an object for the information you need to do something; rather, ask the object that has the information to do the work for you." ← Look closely! Aren't these really the rules of Data Abstraction?

The Key:

1) Abstraction - organize code into discrete units, relatively self contained, limiting focus

Object Oriented Concepts

1) Build Abstractions - self contained units or systems of classes

2) Their job(s) are to perform a particular task – offloading the responsibilities of other units or systems.

3) Therefore we do not need all of the code in existance to design and manage one of these units or systems

4) OO techniques are a way to formalize the organization of code into such units or systems

Questions to Ask - when evaluating a design

For each class:

1) Does it do anything?

2) what is its job?

3) If there are set & get functions, 3a) what does the "using" class need to know to work with the data?

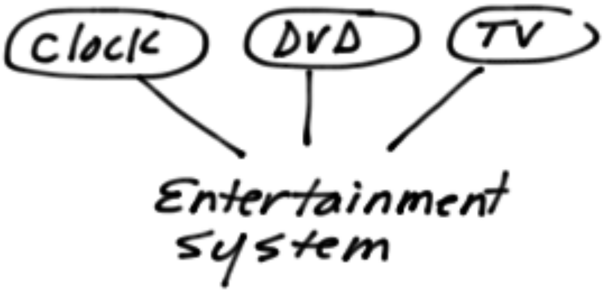
3b - and, what does the "using" class do with the data before/After the set and get calls?

4) Is there a larger "system" this class could be part of (commonalities & specializations)

5) Is the object responsible for too much? (would it be more reasonable to break this into smaller units with more distinct levels of responsibility?)

Avoid with OO & Inheritance

1) using multiple inheritance rather than "has a" to create a general class from two or more specific classes



2) Using multiple inheritance without restriction

3) Base class responsibilities (functionality) incomplete (too little) or overly broad (too much!)

4) Base classes having –

a) no public or protected interface (which means the class is not specifying any protocol to follow)

b) no implementation (only specifie's protocol)

c) subclasses duplicate code

d) most function implementations are overridden

Process of OO Design – CRC Cards

"Class/Responsibilities/Collaborators"

important

1) Provide spatial groupings to explore relationships

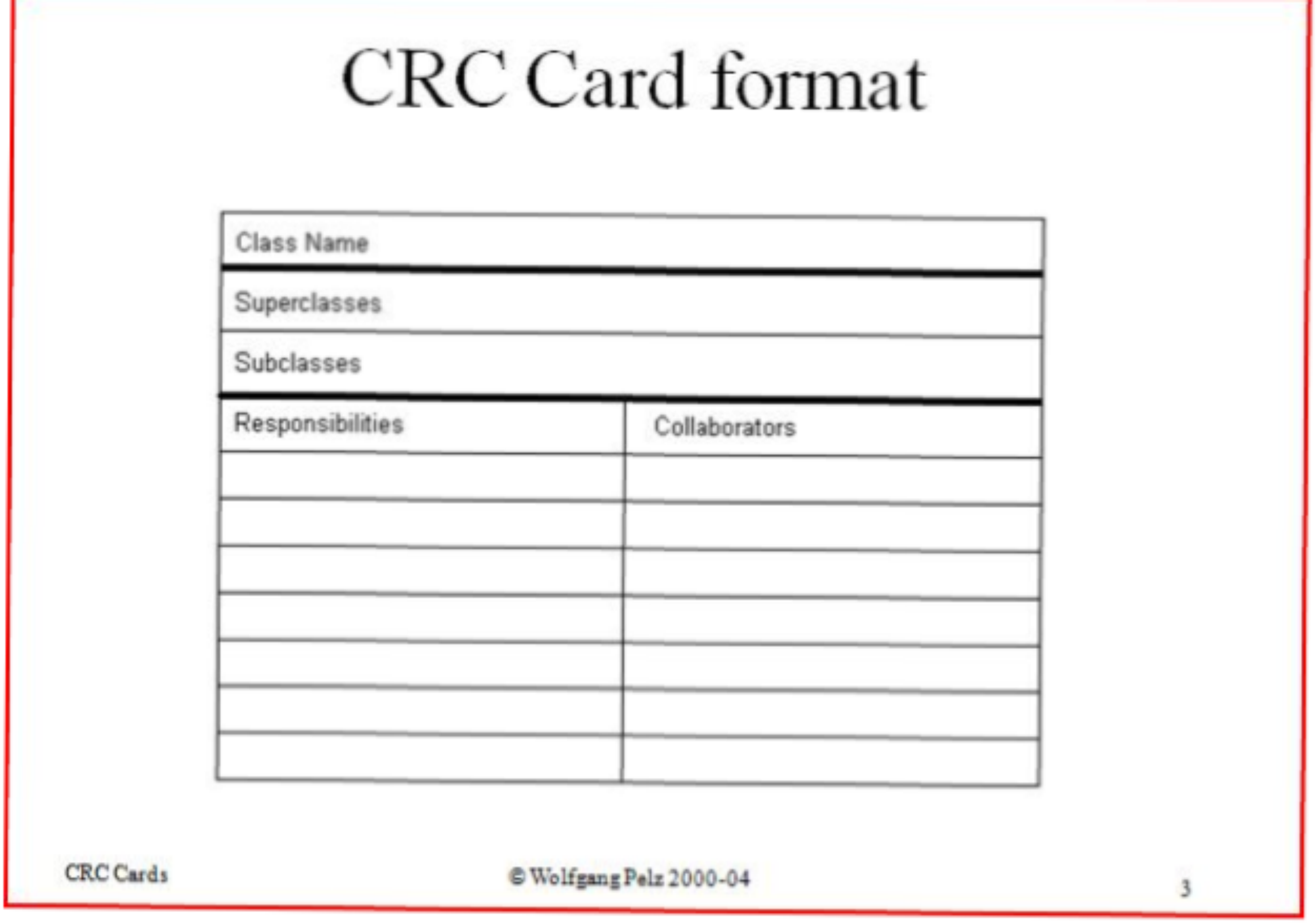
2) Assists in understanding the separation and boundaries of abstractions/units/systems

3) CRC cards should specify:

   - a class of objects
   - their behavior
   - their interactions

4) "Responsibility" – Knowledge, "job", service the class provides and maintains

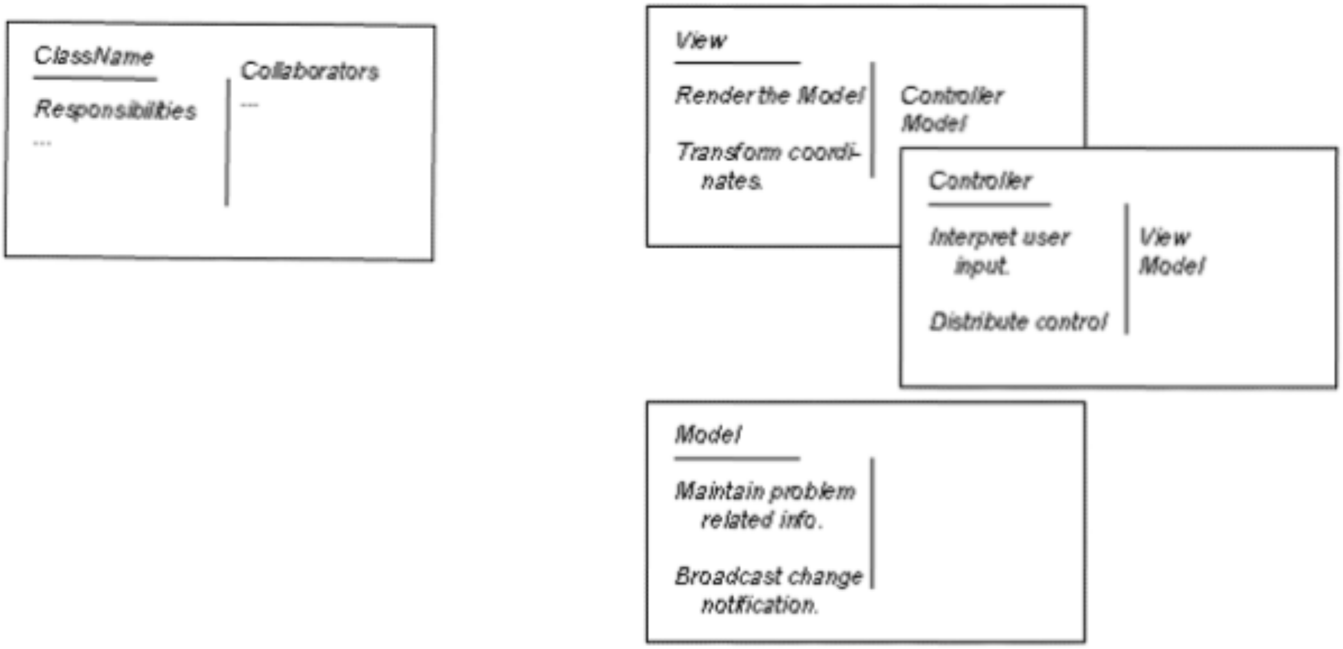
5) "Collaborator" – a class that fulfills a level of responsibility. It has knowledge or a service that is needed to be complete.



CRC Card format

Class Name	
Superclasses	
Subclasses	
Responsibilities	Collaborators

CRC Cards © Wolfgang Pelz 2000-04 3



ClassName
Responsibilities ...
Collaborators ...

View
Render the Model — *Controller Model*
Transform coordinates.

Controller
Interpret user input. — *View Model*
Distribute control

Model
Maintain problem related info.
Broadcast change notification.

Beck, Kent & Cunningham, Ward,
"A laboratory for teaching Object-
Oriented Thinking", OOPSLA '89

Collaboration ?

1) Sometimes a class has a job to do (responsibility) but does not have enough information or abilites to fullfil it!

Therefore - the class object needs to interact with another class (and NO that does not mean they should be friends)

object "uses" functionality of another class via using or has-a relationships & function calls

object "is" the other class + more and has access thru
- protected
- public methods

Iteratively Building CRC Models

1) Find classes
   Let's start with 3-5

2) Find Responsibilites
   what does the class do?
   what does it need to know?
   Are there things it needs to know/do for other collaborating classes?
   Are there things it can use from other classes to minimize re-invention?

3) Define Collaborators
   what other class can be used to assist?
   how would they assist? (what kind of relationship)
   Are there things that are difficult for this class to do – which can more clearly be done by another class

— Does the list of responsibilites Need to change?
— Are Additional classes needed?
— Is there duplication of effort?

Exercises - OOD & CRC cards

Pick one & create an overall design

(online students - login to D2L & begin survey for OOP Activity #1)

1) common word processor that allows for creating, editing & printing documents & pictures

2) A simple compiler that has data types, control structures (if), & loops

3) Software such as Outlook that can manage your contact list, todo list, & email

4) Select one of your assignments this term & create the CRC cards for that assignment. (base this off of what should be done versus what was done!)