

# CRC Cards

# An approach to learning OO

- “The most difficult problem in teaching object-oriented programming is getting the learner to give up the global knowledge of control that is possible with procedural languages, and rely on the local knowledge of objects to accomplish their tasks.”
  - Beck and Cunningham, 1989.
- We can use CRC cards in OOA/D and P!

# What is CRC?

- Class
  - the class name of an object creates a vocabulary for discussing a design.
- Responsibility
  - identifies the problems to be solved by this object.
- Collaborator
  - these are other objects that send or are sent messages in the course of satisfying responsibilities.

# The CRC Card

<u>Class Name</u>	
Responsibilities	Collaborators
...	...

# The CRC Process for OOA pt1

- Decide which objects are needed in the OOA.
  - Do a noun search, for instance.
- Write the class names on the CRC cards.
  - This should be a simple noun or noun phrase.
- Think about what responsibilities each object should have.
  - What do the objects have to do? What jobs do they do?  
Write the responsibilities as verb phrases.

# The CRC Process for OOA pt2

- DISCUSS!
  - Talk to people, ask if these are the right objects. Are these objects really part of the problem domain we are modelling? If not, why not? If yes, make sure this is the case.
  - Are the objects doing the right things? That is, are the responsibilities cohesive to this particular object - do they belong here?
  - Are there objects that don't have any responsibilities? If so, then these might not be objects at all (they might be attributes of other objects).

# The CRC Process for OOA pt3

- Do the objects need other objects to do stuff for them?
- REMEMBER: No object is an island! Objects must link to other objects in some way.
- Which objects do we need to collaborate with? Do they have the right responsibilities?

# The CRC and OOD

- We can use CRC cards to help in designing the internal structure of our system.
- Objects that are part of the PD but not the idesign are removed here.
- We use CRCs in conjunction with use cases to help identify objects that are used in the use case.



# The CRC and OOP

- Again, we can use CRC cards to help us determine the implementation. Use the cards in the same way to determine OOD and from the OOD we can decide on our functions for objects, consider how we modularise the code (if not already done) and think about alternatives.

# Alternatives

- It is very important to consider alternative models of analysis and design.
  - Simply move the cards around, change collaborations and responsibilities.
  - Do we now have a better design?
  - Compare designs side by side. Which is more elegant? Which do you prefer? Why?

# Write it down!

- Don't forget to note the CRC design on paper. Otherwise all that work has gone to waste.
- Once written down and given to all members of the team (in the know) take some time to think about the design.
  - What's missing? What's not relevant? Do these objects meet your perception of the PD?
- “Often, the design turns around a few key decisions about the allocation of responsibilities. Write these down.”
  - Alistair Cockburn