Object Oriented Analysis and Design
Historical Perspective

• Decades of work to provide more structure to software
• Why?
  – S/W becomes ever larger
  – ‘lone programmer’ is a thing of the past
  – Long development times
  – S/W is used for more complex tasks
Software as a Model

• We talk about S/W abstractions
• We hide the details
  – E.G. you don’t have to know how the transmission changes gears to drive a car
• Function decomposition into modules
  – Also called procedural decomposition
• Object Oriented analysis into classes
What’s a Class?

- They divide the problem domain into ‘chunks’
- Self-contained data and functions make it easy to share
- All data needed is contained in the class
- Only functions defined in the class can directly access the data
Why Classes?

• They provide encapsulation
  – Avoids problems with global variables
  – If data is changed it is done within the class

• They provide inheritance
  – Items can be organized in hierarchies
  – Higher levels are more general
  – Common features are ‘shared’ by lower classes
Object Oriented Analysis

• Simply- we break down the problem domain into classes
• Identify potential classes
• Compare classes to look for common elements
• Collect those elements into superclasses
• If a class does not have a single purpose split it into 2 or more classes
Object Oriented Design

• Convert the class hierarchy into classes in the targeted programming language
• Revisit OOA as details are refined
  – Implement new classes
  – Integrate with inheritance requirements
Object Oriented Programming

• Implement the class hierarchy into classes in the targeted programming language
• Develop code to instantiate classes as objects
• Revisit OOA as details are refined
  – Implement new classes
  – Integrate with inheritance requirements
• Conduct unit testing of classes/objects
Nothing new?

- OOAD uses the standard S/W lifecycle
  
  **Analysis**
  **Implementation**
  **Design**
  **Testing**

- The difference is decomposition into classes
- And implementation to create objects