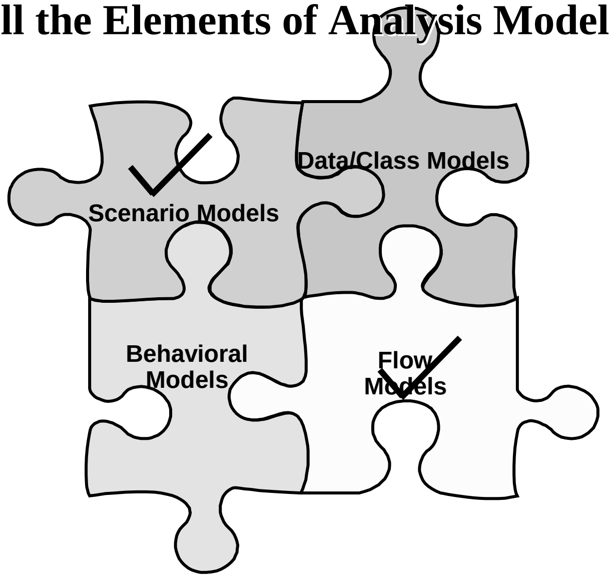
Software Engineering I CSC-382

Lecture 9

Software Engineering I CS-382

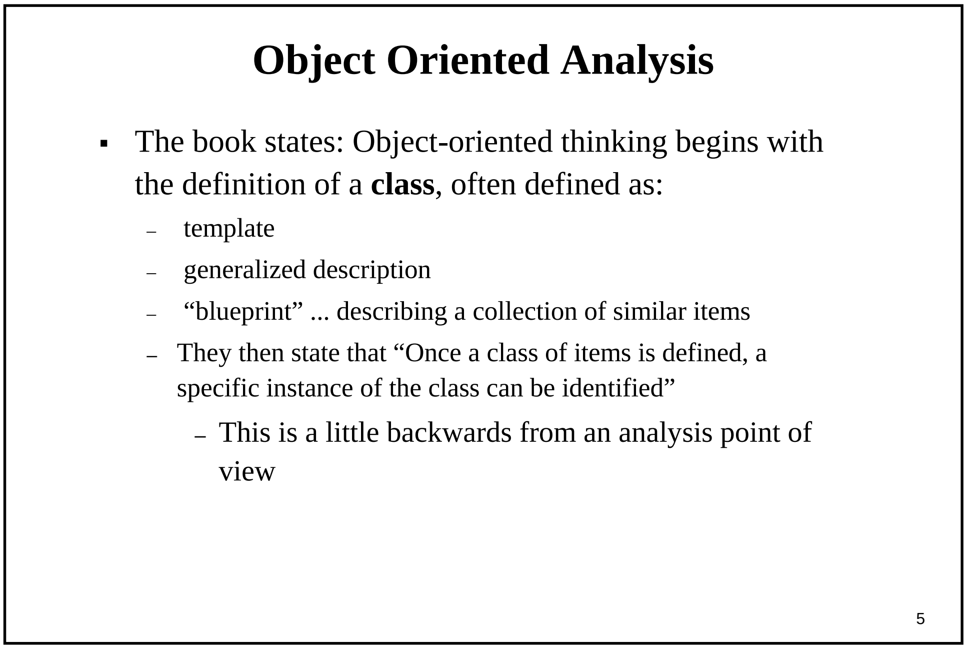
- Lecture 9
- What we will cover: (Details of Analysis Modeling)
  - Chapter 8 Sections 8.4 and 8.7 in Pressman
  - Goal is to understand the methods and tools available from Object Oriented Analysis for analysis modeling

Recall the Elements of Analysis Modeling



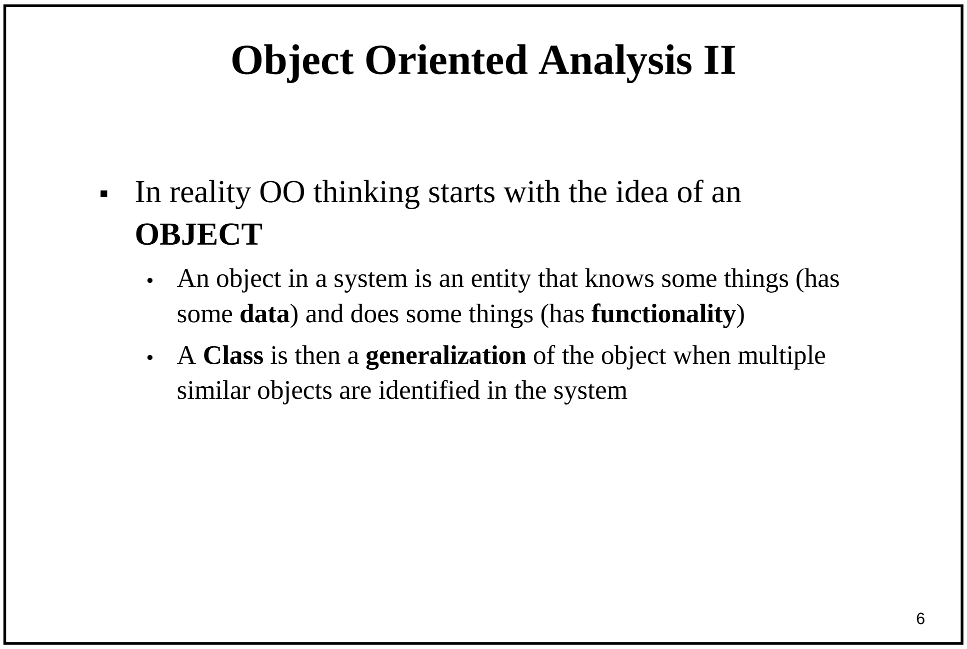
Analysis Modeling via Object-Oriented Analysis

- Must be understood to develop the class and data elements of the analysis model
- Key concepts:
  - Classes and objects
  - Attributes and operations
  - Encapsulation and instantiation
  - Inheritance/Aggregation



Object Oriented Analysis

- The book states: Object-oriented thinking begins with the definition of a **class**, often defined as:
  - template
  - generalized description
  - “blueprint” ... describing a collection of similar items
  - They then state that “Once a class of items is defined, a specific instance of the class can be identified”
    - This is a little backwards from an analysis point of view

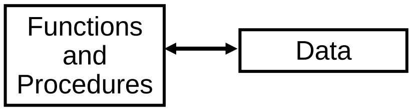


Object Oriented Analysis II

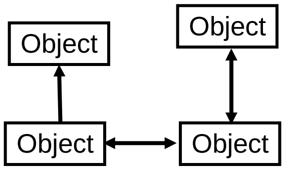
- In reality OO thinking starts with the idea of an **OBJECT**
  - An object in a system is an entity that knows some things (has some **data**) and does some things (has **functionality**)
  - A **Class** is then a **generalization** of the object when multiple similar objects are identified in the system

Object Oriented Analysis vs. Structured Analysis

Structured Analysis Paradigm



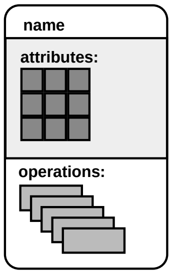
Object-Oriented Analysis Paradigm

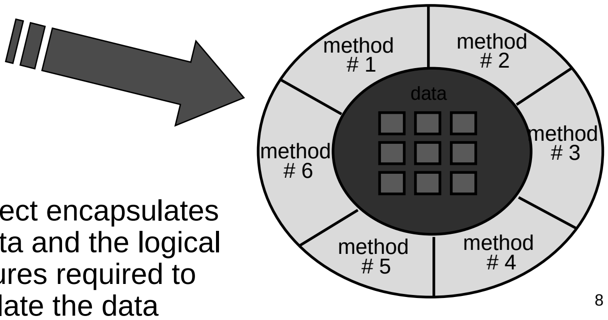


In the structured analysis model the data and the functions that operate on them are artificially separated.

Anatomy of an Object/Class

Recall the data and the functionality are now combined in the object.





The object encapsulates both data and the logical procedures required to manipulate the data

More on Objects

- “An object represents an individual, identifiable item, unit, or entity, either real or abstract with a well defined role in the problem.” – Wilkinson
  - Objects (and their classes) should represent tangible or visible things, roles, events, or concepts within the system.
  - **An object is more than just data and functions bound together**
  - Each object has a set of essential and unique static attributes
  - The ***state*** of an object is the values of these at any time.

More on Objects II

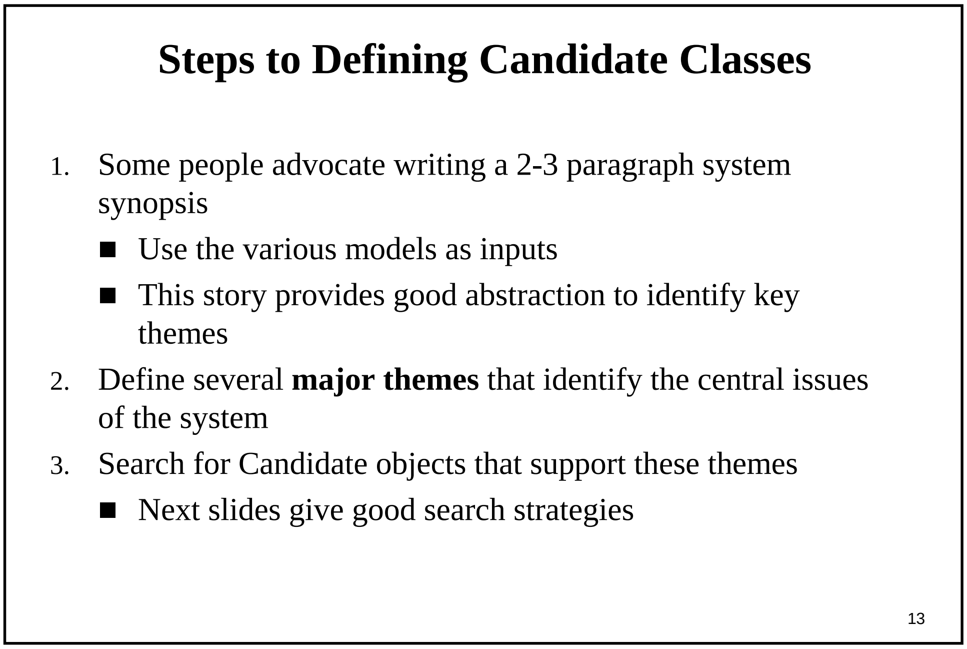
- It is the definition of the set of objects and a description of how they interact to satisfy a system’s goals that comprise the **class model.**

Class-Based Modeling Thru OOA

- Object Oriented Methods view a system as a collection of these objects that communicate to each other thru messages
  - These messages request the various other objects to perform some function or task

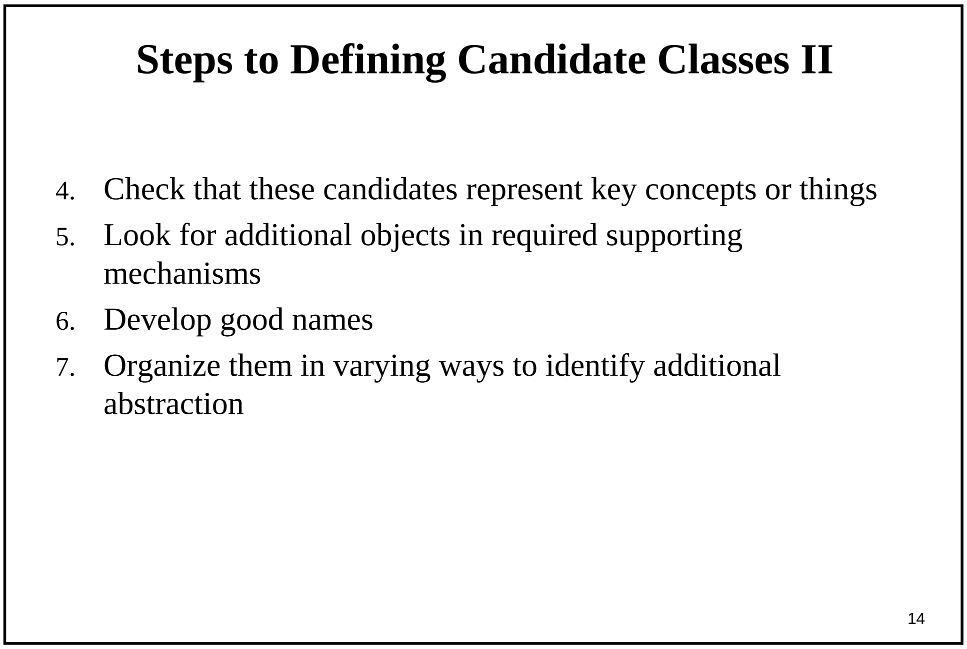
Steps to Class-Based Modeling Thru OOA

1. Identify **classes** by examining the problem statement
   - We will show a CRC-based method for making this a little easier
2. Identify the **attributes** of each class
3. Identify **operations** that manipulate the attributes
4. Later define **inheritances** and **aggregations** of these classes.
5. Also later on define **associations** of the classes based on how they collaborate to accomplish the required functionality



Steps to Defining Candidate Classes

1. Some people advocate writing a 2-3 paragraph system synopsis
   - Use the various models as inputs
   - This story provides good abstraction to identify key themes
2. Define several **major themes** that identify the central issues of the system
3. Search for Candidate objects that support these themes
   - Next slides give good search strategies



Steps to Defining Candidate Classes II

4. Check that these candidates represent key concepts or things
5. Look for additional objects in required supporting mechanisms
6. Develop good names
7. Organize them in varying ways to identify additional abstraction

Search Strategies for Objects

- Good Object candidates often represent:
  - The work the system performs
  - Things directly affected by or connected to the application
  - Information that flows thru the software
  - Decision making, control, and coordination activities
  - Structures and groups of other lower level objects
  - Representations of real-world things the system needs to know something about

Using Class Stereotypes

- It can also help thinking in terms of some common ***stereotypes*** of objects to get ideas
- Some candidate stereotypes include:
  - ***Entity classes***, also called *model* or *business* classes, are extracted directly from the statement of the problem.
  - ***Boundary (or Interface) classes*** are used to create the interface (e.g., interactive screen or printed reports) that the user sees and interacts with as the software is used.
  - ***Controller classes*** manage a “unit of work” [UML03] from start to finish.

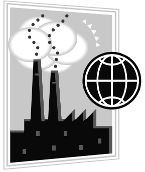
Using Class Stereotypes II

- More candidate stereotypes include:
  - ***Coordinator classes*** synchronize work of various other classes.
  - ***Information holder classes*** are method-less classes that hold data (database type applications have these).
  - ***Service Provider classes*** tend to calculate, compute and figure data.
  - ***Structurer classes*** tend to sort and organize objects into meaningful order.

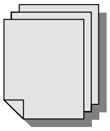
A Simpler Set of Stereotypes

Actor Classes:
•People
•Organizations

Interface Classes:
•Screens
•Menus



Business Classes:
•Places
•Things
•Concepts
•Events



Report Classes:
•Printed
•Electronic

Another Search Strategy for Objects

1. Look for anything that interacts with the system or is a part of the system
2. Ask if there is a customer
   - if there is then they are an object
3. Follow the money
   - The source of money is usually a customer (See 2) and the things or services that the money procures are good candidate

Another Search Strategy for Objects II

4. A Report is an object
   - Any report generated by a system is a good candidate object
   - Reports request information from other objects (collaboration)
5. A User display screen is an object

Additional Sources of Possible Objects and Classes

- ***External entities*** (e.g., other systems, devices, people) that produce or consume information to be used by a computer-based system.
- ***Things*** (e.g, reports, displays, letters, signals) that are part of the information domain for the problem.
- ***Occurrences or events*** (e.g., a property transfer or the completion of a series of robot movements) that occur within the context of system operation.
- ***Roles*** (e.g., manager, engineer, salesperson) played by people who interact with the system.

Additional Sources of Possible Objects and Classes II

- ***Organizational units*** (e.g., division, group, team) that are relevant to an application.
- ***Places*** (e.g., manufacturing floor or loading dock) that establish the context of the problem and the overall function of the system.
- ***Structures*** (e.g., sensors, four-wheeled vehicles, or computers) that define a class of objects or related classes of objects.

In-class Practice Finding Classes

- For Safehome try to identify 5 classes and state where they come from
- For the traction control system identify 5 classes and state where they come from

For Next Class

- Continue to Study Chapter 8;   Pressman (Analysis Modeling)
  - We will cover CRC cards for defining some of the details of our classes