

Object-Oriented Design

Instructor: Dmitri A. Gusev

Spring 2007

CSC 120.02: Introduction to Computer Science

Lecture 6, February 8, 2007

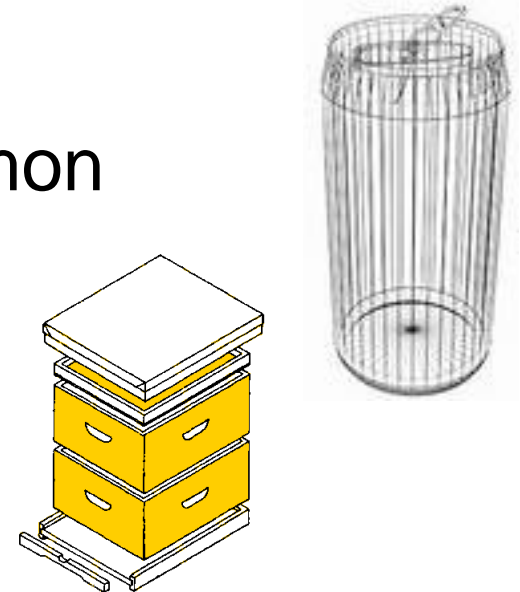
Objects and Classes (recap)

An *object* is a self-contained entity composed of both data and operations (*methods*) that manipulate the data.



Objects are *instances* of *classes*.

A *class* is a description of a group of similar objects. It captures their common characteristics in a formalized way.



Fields and Methods

Fields (instance variables) store data for an object to use.

A *method* is a named algorithm that manipulates the data values in the object.

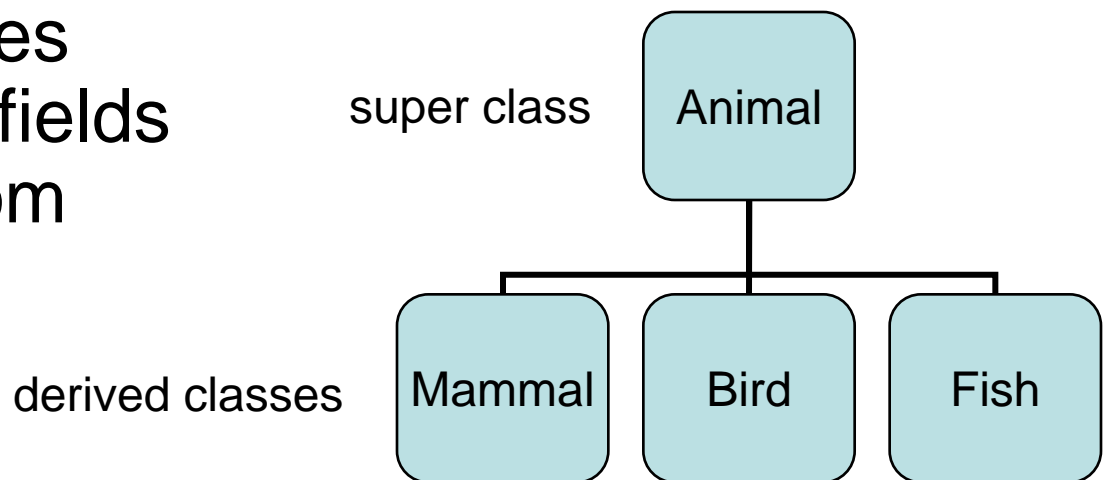


Relationships between Classes

Containment: An instance of a class can contain an instance of another class as a field.



Inheritance: Classes can inherit data fields and methods from other classes.



Relationships between Classes (cont'd)

Collaboration: An instance of one class can call upon an instance of another class to provide information or service.

Object-Oriented Design Methodology

1. *Brainstorming*: Determine the initial set of classes.
2. *Filtering*: Review the classes to see if any of them can be combined, or if more classes are needed. Record the resulting classes on *CRC cards*.

Class name:	Superclass:	Subclasses:
Responsibilities	Collaborations	

Object-Oriented Design Methodology (cont'd)

3. *Scenarios* is the stage in which the behavior (*responsibilities*) of each class is determined.
4. *Responsibility algorithms*: The algorithms are written for each of the responsibilities.

Encapsulation, Information Hiding, Abstraction

Encapsulation is bundling the data and actions so that the logical properties of the data and actions are separated from the implementation details.

Information hiding is the practice of hiding the details of a module with the goal of controlling access to the details of the module.

Abstraction is a model of a system that includes only the details essential to the viewer.