CSE 142

Relationships Between Classes
Introduction to Inheritance

A Design Exercise

- Suppose we are asked to design a set of classes to represent the items in a library's collection
  - Books
  - Magazines/journals
  - CDs
  - Videos/DVDs
  - Etc.
- Ignoring overlap between classes, what object properties and responsibilities would be needed/ appropriate for this?

Your Design Here (1)

Your Design Here (2)

Critique

- What do these classes (objects) have in common?
- How do they differ?
- How do we capture the common parts of the design?
  - Want to describe/define these once, not repeatedly in every class
- How do we relate the specific classes to the common parts of the design?

Notes

Relationships Between Classes

- We already know about objects that contain references to other objects
  - "Contains" or "has-a" relationship
  - Example: a car "has-a" engine, 4 tires, steering wheel, etc.
  - In this case, the relationship is one object being a component of another object
- For the library collection, we'd like to capture the notion that a book or journal "is-a" specialized kind of item in the collection
  - New kind of relationship between classes, not "has-a"

Inheritance

- Key idea of object-oriented programming
- A class can be defined as an extension of another class
  - In Java

    class Book extends CollectionItem { … }
- The extended class ***inherits*** all of the properties and responsibilities of the original class
  - Objects in the extended class have all of the state and methods of the original class
  - Allows us to factor properties/responsibilities common to several classes into a single class that can be extended
- Extended classes can define additional properties and responsibilities that are appropriate for it

Library Classes Revisited

- Re-work your design for the library collection classes to use inheritance
  - Define a single base class CollectionItem with properties and responsibilities common to Books, Journals, CDs, etc.
  - Define extended classes for each of the different kinds of collection items with additional properties/responsibilities appropriate for those classes but not for all CollectionItems in general

Design Using Inheritance (1)

Design Using Inheritance (2)

Summary and Terminology

- Classes can be related by ***inheritance***
- A ***base*** class (or ***superclass***) defines properties/ responsibilities shared by a set of related classes
- A ***derived*** class (or ***subclass***) extends a base class
  - ***Inherits*** all of the properties/responsibilities of the base class
  - Can define additional properties/responsibilities
- Extended classes are related to base classes by "is-a"
  - A Book object "is-a" CollectionItem object
    (in addition to whatever else it can do)
- Next Lectures: Work out the implications of these ideas
  - And look at some specific details of how its done in Java