Components and Inheritance

Elements of Graphics
CS324e
Spring 2018
Object Review

- Objects have fields and methods
  - Fields are attributes of that object
  - Methods are functions of that object
- Objects can be fields of other objects
Composite Objects

- Objects that include other objects
- Build higher levels of abstraction
- Create greater modularity
- Component-based design allows for an object to be composed of other object instances with desired functionality
Component-based Example

- Components of a Bike object?
  - Frame
  - Wheels
  - Brakes
  - Drivetrain
  - Handlebars
Inheritance

- A class can *inherit* fields and methods from another class
- An object that inherits from another is a subclass (derived class)
- The object it inherits from is the superclass (base class)
- A subclass extends a superclass
- A subclass contains all methods and fields of the superclass
Why Use Inheritance?

- Inheritance allows for more generalized code
- A general class of behaviors can be extended to a group of more specialized subclasses
- A superclass method can be overridden in the subclass to create that specific behavior
- Superclass: Vehicle
- Subclasses: Car, Train, Ship, Plane etc
Consider…

❖ Consider superclass Vehicle and subclasses Car and Train

❖ What is a method/field in the Vehicle class that would lend itself to use in both the Car and Train classes?

❖ What is a method/field in the Train class that the Car class wouldn’t need?
ColoredSpot Example
Hands-on: Building with Inheritance

Today’s activities:

1. Get Spot subclass, ColoredSpot, working in your code
2. Create a Spot subclass, TwoSpots. TwoSpots displays two ellipses around a center point
3. Move a TwoSpots object across the canvas