

A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light green. They are positioned diagonally, with the blue one partially covering the green one.

CIS 110 Recitation Objects

July 25th 2017



Class Members: Static vs. Instance Variables

- Each instance of a class (i.e. each object you create) has a distinct copy of its instance variables (non-static).
 - ex: Each Car object created using its constructor has its own values for make, model, and speed and its own places in memory to store these values.
- Each class has one single copy of its static variables
 - ex: The Car class has a static field called numCars that is equal to the number of Car objects that have been created.



Class Members: Static vs. Instance Methods

- Each instance of an object can use instance methods that have access to that object's fields and to its other instance methods.
 - ex: `String s = "hello";`
 - `int length = s.length();`
- Each class has static methods that only have access to the class's static fields.
 - ex: `Car.getNumCars();`
 - Notice the invocation begins with the class name and not the name of an instance of the class (i.e. the name of an object).



Exercise: Ball Class

- Let's say we want to simulate a ball subject to gravity that bounces off the sides of the PennDraw canvas window.
- What properties or fields should each ball object have?
 - How about properties that the class as a whole should have?
- What will our constructor look like?
- What instance methods do we need?
- Let's code it!