Objects and Methods
Announcements

• Lab 4 posted; due Saturday
  - Start early, this one is a big step up

• Quiz 2 on Friday; study guide posted

• This is a busy week for us—work hard and you’ll be in good shape
Today’s plan

• Review

• Using objects to store and interact with compound data

• Strings and lists as objects

• Zelle graphics library

• Examples
Review program
Objects

• Types: values with associated operations
  - int, float, bool
  - Unchanging or immutable

• Objects: compound data objects with associated methods. Methods can access and modify the data.
  - Point, Circle, Rectangle
  - Student, Course
  - Changeable or mutable
Constructors

• Each object is an **instance** of a **class** of objects, i.e. Point or Student

• We create an instance by invoking the class **constructor**, providing some initial data

  - `p1 = Point(0, 0)`

  - `<class-name>(<param1>, <param2>, ...)`
Methods

- We interact with objects/instances of a class by calling methods.

- Methods allow us to enforce information hiding or encapsulation: an object’s data can only be accessed or modified through methods.
  - p1.getX()
  - p1.move(10, 5)
  - <object>..<method-name>(param1, param2, ...)

- Calling a method is like passing a message to an object. Methods determine how you can interact with an object’s internally-stored data.
Why use objects?

• A way of splitting programs into more manageable pieces, as a complement to functions.

• As with functions, creates an interface that hides implementation details.

• Meshes with the way we sometimes think about the world.

• Python is multi-paradigm, encourages hybrid functional and object-oriented approach.
Strings and lists as objects

• Not created with a constructor, but they do have methods
  - sequence methods: .count(), .index()
  - string methods: .upper(), .lower()
  - list methods: .append(), .reverse(), .sort()

• Strings are immutable, lists are mutable
Graphics

- `from graphics import *`

- Start by creating a graphics window object using the `GraphWin` class

- Create other shapes, draw them in the graphics window using the `.draw()` method

- Other shapes: `Point`, `Line`, `Rectangle`, `Circle`, etc.
Draw this scene
Have a nice day : )