Review

• In Python, everything is an object
  – E.g., classes and instances of classes

• An abstract data type is a set of objects and the operations on those objects
  – Specifications define interface between objects and clients

• We implement abstract data types using classes
Review

• A class definition creates an object of type `type` and associates with that object a set of objects of type `instancemethod`

• **Instantiation** is used to create instances of a class.

• **Attribute references** use dot notation to access attributes associated with the class.
Review

• When data attributes are associated with a class we call them **class variables**
• When they are associated with an instance we call them **instance variables**
Review

• Classes provide **encapsulation**. The data structures and the methods that operate upon them are encapsulated in a single module.

• Classes do no provide **information hiding**. Client code can directly access the data attributes. It shouldn’t!
Violating Abstraction Boundaries Is Evil
Inheritance

• Provides a convenient mechanism for building groups of related abstractions
• Most of the time, this involves a hierarchy of related types—with object at the top
• Subclass inherits all of the attributes of its superclass
  – Can also add new attributes
  – Can also override inherited attributes