

Defining Classes, Part 1

Rose-Hulman Institute of Technology
Computer Science and Software Engineering

Check out [17-DefiningClasses](#) from
SVN

Review: What is an Object?

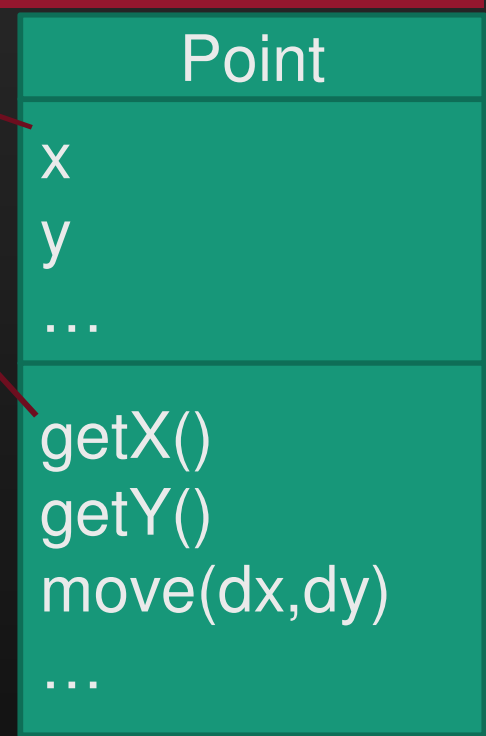
- An Object is an active datatype:
 - Knows things about itself
 - Stored in *fields* (a.k.a. *instance variables*)
 - Can be asked to do things
 - *mutator* methods
 - Can be asked provide information
 - *accessor* methods

Object Terminology

- Objects are *instances* of some *class*
- Objects are created by calling the *constructor* of a class
- We use UML class diagrams to visualize

instance variables

methods



Object Analogies

Key
Concept

- A class is an “object factory”
 - Calling the constructor tells the class to make a new object
 - Parameters to constructor are like “factory options”
 - Used to set instance variables
- Or think of class like a “rubber stamp”
 - Calling the constructor stamps out a new object
 - Parameters to constructor “fill in the blanks”

Look at `object_example.py`

Review: Using Objects in Python

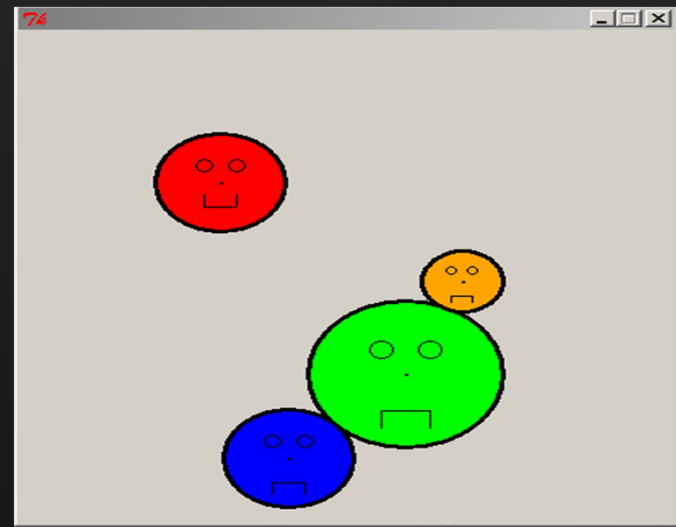
Example Object Diagram

`p = Point(200, 100)`

`t = Text(p, 'Go Colts!')`

Defining Our Own Classes

- Custom objects:
 - Hide complexity
 - Provide another way to break problems into pieces
 - Make it easier to pass information around
- Example: Moving “Smiley” class



Coding Moving Smileys

- Create constructor noting default parameters
 - Defaults for **size**, **color**, and **isSmiling**
 - Study the code for creating **parts**
 - Explore how **parts** list is created
- Create **draw()** method and run **scene1**
- Add **move()** method, and run **scene1**
- Add **smile** and **frown** methods, which need to know about size
- Run **scene2**, point out that 3 other methods needed for collisions to work

Review of Key Ideas

- *Constructor:*
 - Defined with special name `__init__`
 - Called like `ClassName()`
- *Instance variables:*
 - Created when we assign to them
 - Live as long as the object lives
- **self** formal parameter:
 - Implicitly gets the value before the dot in the call
 - Allows an object to “talk about itself” in a method

Project introduction

- on separate slides