Object Orientation in Ruby

4010-350 Personal Software Engineering



class Point

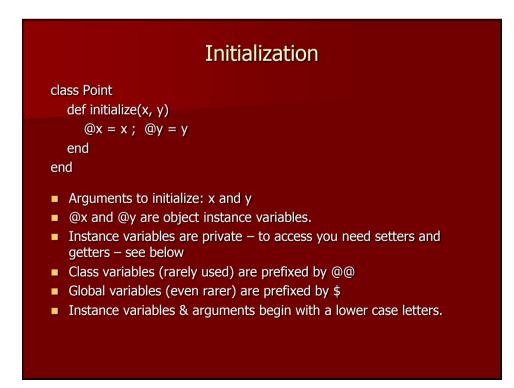
end

- Technically "Point" is a constant (as is any other entity whose name begins with a capital.
- By default, the super-class of Point is "Object."
- Note we can extend a class at any point by simply opening it up and adding behavior
- Do so *very* carefully.

Creating an Object in a Class

p = Point.new(x, y)

- new is a class method (like static in Java).
- It allocates space and calls the initialize method in of the new object.
- initialize looks like a constructor, but it is just a method called by the new class method.
- Since Ruby is dynamically typed, there is no way to create multiple initialize methods.



Default Arguments

class Point

def initialize(x = 0, y = 0)

@x = x; @y = y

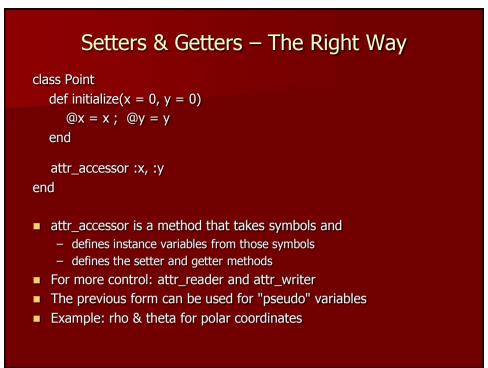
end

end

- p = Point.new p is initialized to the origin.
- p = Point.new(5) p is initialized to (5, 0).
- p = Point.new(3, 7) p is initialized to (3, 7)

Setters & Getters – The Wrong Way

```
class Point
    def initialize(x = 0, y = 0)
        @x = x ; @y = y
        end
        def x
            @x
        end
        def x=(newx)
            @x = newx
        end
        end
        end
        end
```



Other Instance Methods	
class Point def move_by(deltax, deltay)	
<pre>@x += deltax ; @y += deltay self end</pre>	# hmmmm???
def move_to(other_point) @x = other_point.x ; @y = other_j self end	point.y
def to_s #override default conver "(#{@x}, #{@y})" end end	rter to String

Class Methods & Variables

class Point @@count = 0 def initialize(x = 0, y = 0) @@count += 1 @x =x ; @y =x end def Point.count @@count end end

