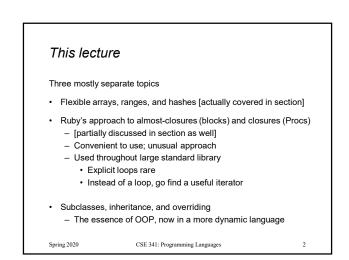
W PAUL G. ALLEN SCHOOL of computer science & engineering

CSE341: Programming Languages

Lecture 20 Arrays and Such, Blocks and Procs, Inheritance and Overriding

> Brett Wortzman Spring 2020



Ruby Arrays

- Lots of special syntax and many provided methods for the Array class
- Can hold any number of other objects, *indexed* by number
 Get via a[i]
 - Set via a[i] = e
- Compared to arrays in many other languages
 - More flexible and dynamicFewer operations are errors
 - Less efficient
- "The standard collection" (like lists were in ML and Racket)

CSE 341: Programming Languages

Using Arrays

- · See many examples, some demonstrated here
- Consult the documentation/tutorials
 If seems sensible and general, probably a method for it
- Arrays make good tuples, lists, stacks, queues, sets, ...
- Iterating over arrays typically done with methods taking blocks
 Next topic...

CSE 341: Programming Languages

Spring 2020

4

Blocks

Spring 2020

Blocks are probably Ruby's strangest feature compared to other PLs

But almost just closures

- Normal: easy way to pass anonymous functions to methods for all the usual reasons
- Normal: Blocks can take 0 or more arguments
- Normal: Blocks use lexical scope: block body uses environment where block was defined

```
Examples:

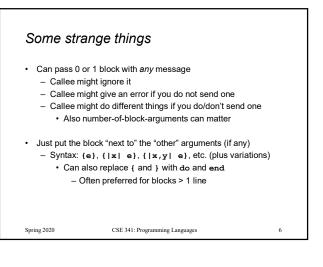
3.times { puts "hi" }

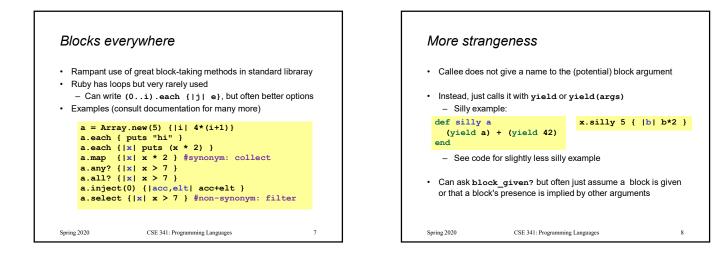
[4,6,8].each { puts "hi" }

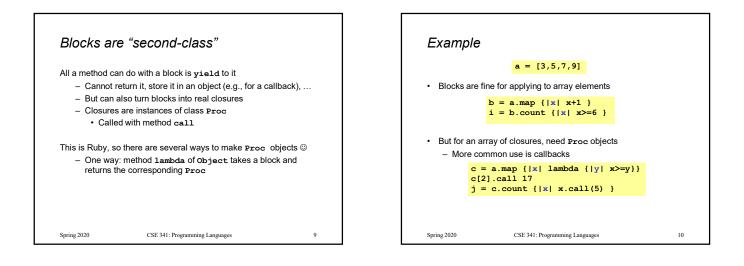
i = 7

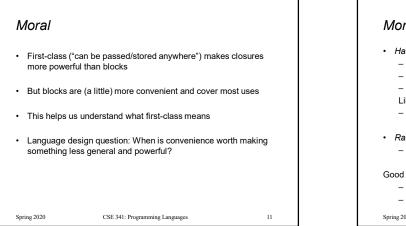
[4,6,8].each { |x| if i > x then puts (x+1) end }

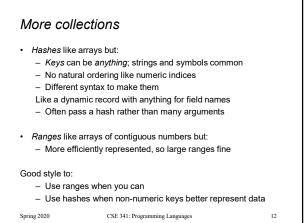
Spring 2020 CSE 341: Programming Languages 5
```

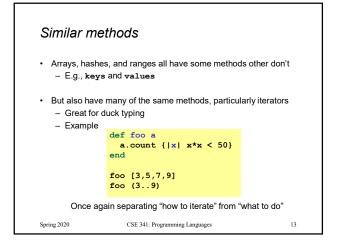


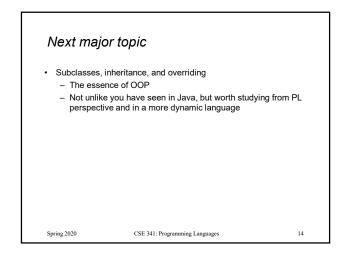


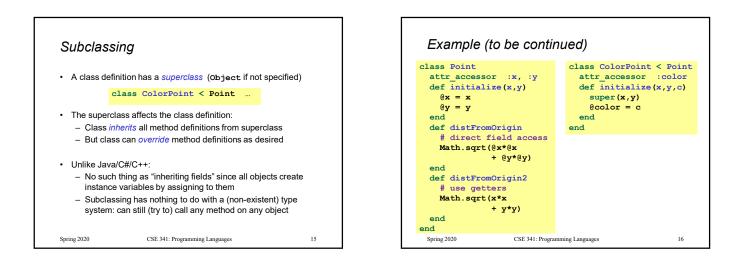


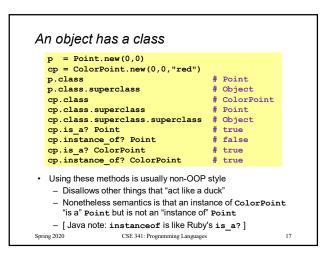


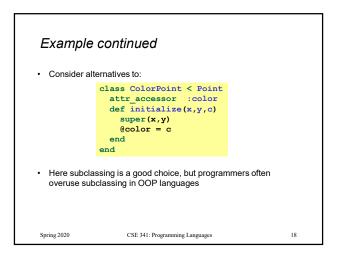












Why subclass • Instead of creating ColorPoint, could add methods to Point • That could mess up other users and subclassers of Point class Point attr_accessor :color def initialize(x,y,c="clear") @x = x @y = y @color = c end end

