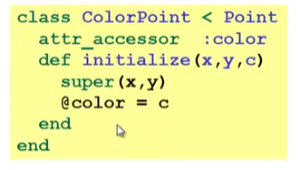
Example continued

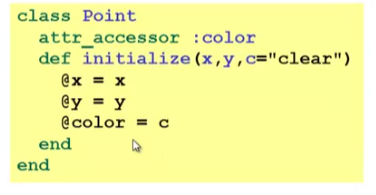
* Consider alternatives to:



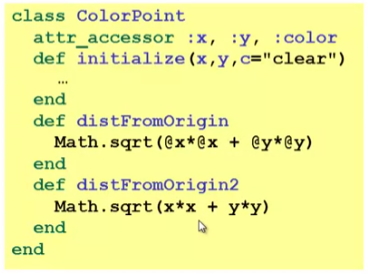
* Here subclassing is a good choice, but programmers often overuse subclassing in OOP languages

Why subclass

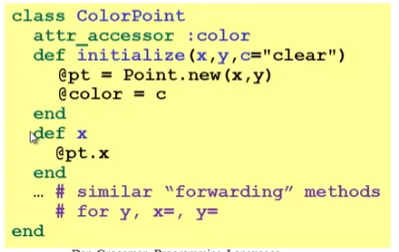
* Instead of creating ColorPoint, could add methods to Point
  + That could mess up other users and subclassers of Point
  + Requiring every Point in the system to have these methods even if they do not need them



* Instead of subclassing Point, could copy/paste the methods
  + Means the same thing if you don’t use methods like is\_a? and superclass, **but of course code reuse is nice**



* Instead of subclassing Point, could use a **Point instance variable**
  + Define methods to send same message to the Point
  + Often OOP programmers overuse subclassing
  + But for ColorPoint, subclassing makes sense: less work and can use a ColorPoint wherever code expects a Point



* + This will not return true in **is\_a? Point** even if they instantiate inside its body