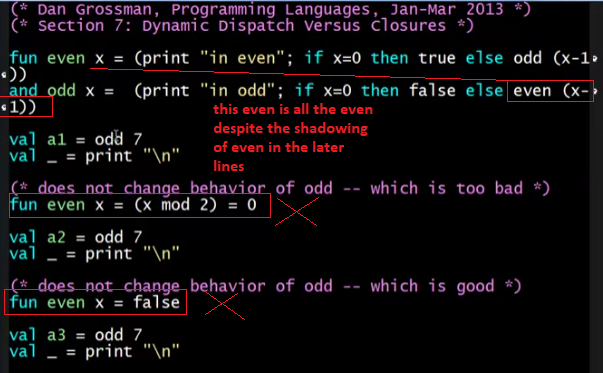
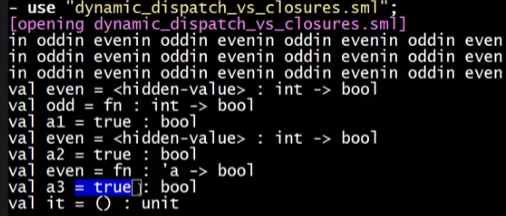
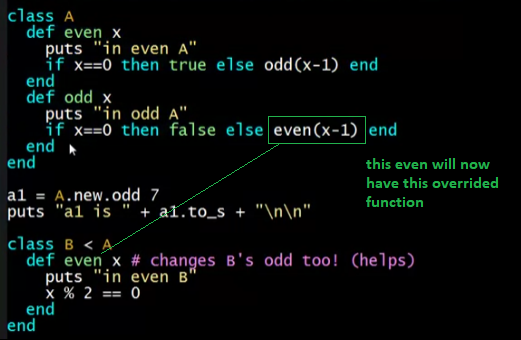
Closures in ML





Dynamic Dispatch in Ruby

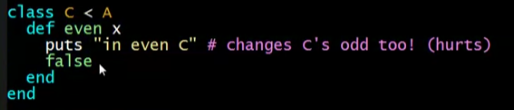




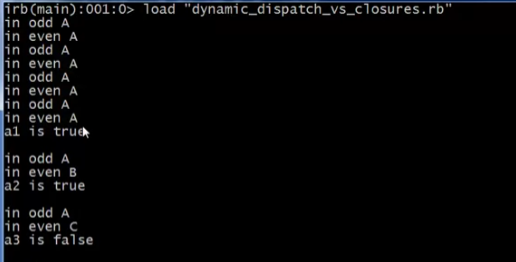
Isn't dynamic dispatch great? However, it is a double edged sword.

You may end up overriding a method that is relied on by other methods, without even knowing that it is important.

Class B has made both odd and even better, but class C, by breaking even, breaks both odd and even.







The OOP trade-off

* Any method that makes calls to overridable methods can have its behavior changed in subclasses even if it is not overridden
  + Maybe on purpose, maybe by mistake
  + Observable behavior includes calls-to-overridable methods
* So *harder* to reason about “the code you’re looking at”
  + Can avoid by disallowing overriding (less object-oriented)
    - “private” or “final” methods
* So *easier* for subclasses to affect behavior without copying code
  + Provided method in superclass is not modified later