Ruby methods use late binding: simple example:

```ruby
class A
  def even x
    puts "in even A"
    if x==0 then true else odd(x-1) end
  end
  def odd x
    puts "in odd A"
    if x==0 then false else even(x-1) end
  end
end

a1 = A.new.odd 7
puts "a1 is " + a1.to_s + "\n"

class B < A
  def even x
    # changes B's odd too! (helps)
    puts "in even B"
    x % 2 == 0
  end
end

a2 = B.new.odd 7
puts "a2 is " + a2.to_s + "\n"

class C < A
  def even x
    puts "in even C" # changes C's odd too! (hurts)
    false
  end
end

a3 = C.new.odd 7
puts "a3 is " + a3.to_s + "\n"
```