Chapter 5: Objects and Classes

Classic example: Cartesian points

```
class Point(Object):
    def __init__(x, y):
        self.x = initx
        self.y = inity

    def move(dx, dy):
        x += dx;
        y += dy;

p = Point(10, 20)
p.move(3, -5)
```

```
class point extends object
  field x
  field y
  method initialize (initx, inity)
    begin
      set x = initx;
      set y = inity
    end
  method move (dx, dy)
    begin
      set x = +(x, dx);
      set y = +(y, dy)
    end

let p = new point(10, 20)
in send p move(3, 5)
```

5.2 Inheritance: Issues

Single (Java) vs. Multiple (C++) inheritance:

```
class foo extends object
  field x
  method initialize () set x = 1

class bar extends object
  field x
  method initialize () set x = 2

class foobar extends foo, bar
  method getx () x

myfoobar = new foobar ()
send myfoobar getx()
```

Shadowed declarations (like `let, lambda`):

```
class c1 extends object
  field y
  method initialize() 1
  method sety1 (v) set y = v
  method gety1 () y
class c2 extends c1
  field y
  method sety2 (v) set y = v
  method gety2 () y
let o2 = new c2()
  in begin
       send o2 sety1 (102);
       send o2 sety2 (999);
       list (send o2 gety1(), % returns 102
             send o2 gety2()) % returns 999
     end
```

- Statics vs. Dynamic Method Dispatch:

```
class c1 extends object
  method initialize() 1
  method m1 () 1
  method m2 () send self m1()
class c2 extends c1
  method m1 () 2
let o1 = new c1 ()
    o2 = new c2 ()
in list (send o1 m1(), % returns 1
         send o2 m1(),   % returns 2
         send o2 m2()) % returns ???
```

- Static: returns 1 (declaration context)
- Dynamic: returns 2 (calling context)
- We'll use dynamic; however...

```
class point extends object
  field x
  field y
  method initialize(initx, inity)
    begin
      set x = initx;
      set y = inity
    end
class colorpoint extends point
  field color
  method initialize (initx, inity, initcolor)
    begin
      set x = initx;
      set y = inity;
      set color = initcolor
    end
```

What's wrong here: what could we do better?

```
class point extends object
  field x
  field y
  method initialize(initx, inity)
    begin
      set x = initx;
      set y = inity
    end
class colorpoint extends point
  field color
  method initialize (initx, inity, initcolor)
    begin
      super initialize(initx, inity)
      set color = initcolor
    end
```

super call uses static method dispatch:

```
class c1 extends object
  method initialize () 1
  method m1 send self m2()
  method m2() 13
class c2 extends c1
  method m1 () 22
  method m2 () 23
  method m3 () super m1()
class c3 extends c2
  method m1 () 32
  method m2 () 33
let o3 = new c3 ()
in send o3 m3 () ; returns 33
```