Assignment 1 announcements

Look out for extra tests and a brief marking scheme **tonight**

Extra office hours
- David: Thurs 2:30-4:30, Friday 3:00-5:30
- Andrew: Mon 3-5, Tues 3-5 in **BA4261**
Object-oriented programming: principles and implementation
“OOP to me means only messaging, local retention and protection and hiding of state-process, and extreme late-binding of all things.”

–Alan Kay (inventor of Smalltalk)
“OOP to me means only **messaging**, local retention and protection and hiding of state-process, and extreme late-binding of all things.”

–Alan Kay (inventor of Smalltalk)
Sending messages

(aka “the most important punctuation mark in OOP”)
“OOP to me means only messaging, local retention and protection and hiding of state-process, and extreme late-binding of all things.”

–Alan Kay (inventor of Smalltalk)
Late binding (dynamic binding)

objects resolve messages at runtime
I want class.
**macro**: a transformation on *program syntax*

“syntax -> syntax” function
Pattern-based macros

(define-syntax syntax-name
  (syntax-rules ()
    ([pattern template] ...)))
Let’s implement class!
Announcements

Look out for extra tests and a brief marking scheme

Extra office hours
- David: Thurs 2:30-4:30, Friday 3:00-5:30
- Andrew: Mon 3-5, Tues 3-5 in BA4261

(NEW!) Labs will now be held exclusively in BA2200
Macro ellipses

In a pattern, operates like Kleene star: *

<attr> ...  
(⟨x⟩  ⟨y⟩) ...  
(⟨a⟩  ...) ...
Macro ellipses

In a template, operates like map

(list <attr> ...)
(+ <x> <y>) ...
(list (+ <a> ...) ...) ...)

Back to class!
The problem of this