Today

Objects

- Methods

P3

classes

\[ \text{class } C : \quad x = 3 \]

\[ \Rightarrow \quad \text{declassify temp} = \text{create class } \]

\[ \Rightarrow \quad C = \text{temp} \]

Objects are instances of classes

class C :: ....
\[ \text{O} = C(C) \]

class E ::
\[ y = 2 \]

class D (E)
\[ x = 1 \]
\[ \text{O}_1 = D(C) \]
\[ \text{O}_2 = D(C) \]

print D.x = 1
print O1.x = 1
print D.y = 2
print O1.y = 2
\[ \text{O}_1.x = 3 \]
\[ \text{O}_1.z = 5 \]

(Handwritten notes on the right side of the page, possibly statistical data or additional comments.)
class C:
    def __init__(self, o, n):
        print n

o = C(42)

Objects and Classes — are runtime thangys that live on the heap

Watch out for aliasing

o = C(42)
q = o
q.x = 1
print o.x

class C:
f = lambda o, x:

O = C()
o.f = lambda : O

print o.f()
class C:
    def move(s, dx):
        s.x = s.x + dx

O = C()
bnd = O.move
bnd has a bound method
bnd(25)

unb = C.move
unb has a unbound method
unb(0, 25)

a.z = unb
a.z(0, 25)
class C:
  x = 3
  if True:
    def foo(self, y):
      w = x
      return y+w
    \[ z = x + 9 \]
  else:
    def foo(self, y):
      return self.x + y
    print 'hello world!'

x = 1

\[ \text{class C:} \]
\[ \text{C.x} = 3 \]
\[ \text{if True:} \]
\[ \text{def } \_\_\_\text{foo}_\_\_\text{tmp}(\text{self}, y) \]
\[ w = x \]
\[ \text{return } y+w \]
\[ \text{C.x} = C.x + 9 \]
\[ \text{else:} \]
\[ \text{def } \_\_\_\text{foo}_\_\_\text{tmp}(\text{self}) \]
\[ \text{return } \text{self}.x + y \]
\[ \text{print } \text{C}.\text{foo} = \_\_\_\text{foo}_\_\_\text{tmp} \]
class C( B_1, ..., B_n ):
    body

emp = create_class(CB_i, ..., B_n)

newbody (body with appropriate attribute transformation)

emp.x

C = emp