Meeting 23: Objects

<table>
<thead>
<tr>
<th>IF A RESEARCHER SAYS A COOL NEW TECHNOLOGY SHOULD BE AVAILABLE TO CONSUMERS IN...</th>
<th>WHAT THEY MEAN IS...</th>
</tr>
</thead>
<tbody>
<tr>
<td>THE FOURTH QUARTER OF NEXT YEAR</td>
<td>THE PROJECT WILL BE CANCELED IN SIX MONTHS.</td>
</tr>
<tr>
<td>FIVE YEARS</td>
<td>I'VE SOLVED THE INTERESTING RESEARCH PROBLEMS. THE REST IS JUST BUSINESS, WHICH IS EASY, RIGHT?</td>
</tr>
<tr>
<td>TEN YEARS</td>
<td>WE HAVEN'T FINISHED INVENTING IT YET, BUT WHEN WE DO IT'LL BE AWESOME.</td>
</tr>
<tr>
<td>25+ YEARS</td>
<td>IT HAS NOT BEEN CONCLUSIVELY PROVEN IMPOSSIBLE.</td>
</tr>
<tr>
<td>WE'RE NOT REALLY LOOKING AT MARKET APPLICATIONS RIGHT NOW.</td>
<td>I LIKE BEING THE ONLY ONE WITH A HOVERCAR.</td>
</tr>
</tbody>
</table>

Announcements

- HW6 due 11/17
- Project status due 11/17 (in your project repo)

Class Scoping Puzzle
class D(C):
    print x
print C.x
print D.x

>>> x = 1
>>> class C:
    ...
    ... print x
    ... x = 2
    ... print x
    ...
    1
    2
>>> class D(C):
    ...
    ... print x
    ...
>>> print C.x
2
>>> print D.x
2

class C:
    - dynamic scoping
    x=2
    class C:
        ...
        if False:
            x=10
        print x
        ...

set f0:
    - static scoping
    x=12
    def f0():
        ... local x ...
        if False:
            x=10
        print x
        ...

f0()
Questions

1. Compiling classes
2. Conditions in loops
3. Class attributes
4. Strings in assembly

.data
.asciiz "foo"
.asciiz "bar"

.data (global variables)

.code

.code
class C:

class B:

\[ \text{topC} = \text{create\_class}(\ldots) \]

\[ \text{topB} = \text{create\_class}(\ldots) \]

\[ B = \text{topB} \]

\[ \Rightarrow \text{set\_attr}(\text{topC}, "B", \ldots) \]

\[ \begin{array}{|c|c|c|c|c|c|}
  \hline
  L_0 & \phi & \phi & e_1 & \phi & \phi \\
  \hline
  L_1 & \phi & e_2 & e_3 & e_2 & e_2 \\
  \hline
  L_2 & \phi & \phi & \phi & e_3 & e_3 \\
  \hline
  L_3 & \phi & \phi & e_2 & e_3 & e_3 \\
  \hline
  L_4 & e_3 & e_3 & e_3 & e_3 & e_3 \\
  \hline
\end{array} \]

\[ \text{while } C : \]

\[ \text{body} \]

\[ \text{object code} \]

def \text{metacode}(n, \text{While})::

\[ L_1 = L_4 \cup L_2 \ldots \cup \text{end} = \text{get\_attr}(L_1) = L_1 \]

\[ \text{while } \text{metacode} \]

\[ \Delta_3 = \ldots \ldots L_1 \ldots \]

\[ L_0 = \ldots L_1 \ldots \]

\[ L_2 = \text{liveness}(n, \text{body}, L_3) \]

\[ \overline{L_1} = L_1 \cup L_2 \]
while x > 3:
    body

Ladder = L_0 = L_1
L_1 = L_2 \cup \{ x \}
L_2 = L_{\text{before}}(S_{\text{body}}, L_3)
L_3 = L_1
L_4 = \text{Ladder}

and return L_0