Office Hours: You Should Go!

You are not alone!

http://inst.eecs.berkeley.edu/~cs61a/fal2/staff.html

### The Game of Hog

<table>
<thead>
<tr>
<th>Number of dice rolled</th>
<th>Expected score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.5</td>
</tr>
<tr>
<td>2</td>
<td>5.9</td>
</tr>
<tr>
<td>3</td>
<td>7.4</td>
</tr>
<tr>
<td>4</td>
<td>8.2</td>
</tr>
<tr>
<td>5</td>
<td>8.6</td>
</tr>
<tr>
<td>6</td>
<td>8.7</td>
</tr>
<tr>
<td>7</td>
<td>8.5</td>
</tr>
<tr>
<td>8</td>
<td>8.2</td>
</tr>
<tr>
<td>9</td>
<td>7.8</td>
</tr>
<tr>
<td>10</td>
<td>7.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number of dice rolled</th>
<th>Chance of scoring 10+</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16%</td>
</tr>
<tr>
<td>2</td>
<td>19%</td>
</tr>
<tr>
<td>3</td>
<td>23%</td>
</tr>
<tr>
<td>4</td>
<td>28%</td>
</tr>
<tr>
<td>5</td>
<td>33%</td>
</tr>
<tr>
<td>6</td>
<td>40%</td>
</tr>
<tr>
<td>7</td>
<td>48%</td>
</tr>
<tr>
<td>8</td>
<td>48%</td>
</tr>
<tr>
<td>9</td>
<td>16%</td>
</tr>
<tr>
<td>10</td>
<td>0%</td>
</tr>
</tbody>
</table>

### Environments Enable Higher-Order Functions

#### Higher-order function
A function that takes a function as an argument value or returns a function as a return value

- **Functions as arguments:**
  
  Our current environment model handles that already!

  We'll discuss an example today

- **Functions as return values:**
  
  We need to extend our model a little

  Functions need to know where they were defined

  Almost everything stays the same  

### Names Bound to Functional Arguments

#### Example:

```python
1 def apply_twice(f, x):
2     return f(f(x))

4 def square(x):
5     return x * x
7 result = apply_twice(square, 2)
```

### Non-Nested Functions Calls Have One Local Frame

#### Example:

```python
1 def f(x, y):
2     return g(x)
3     return x + y
4 def g(a):
5     result = f(a, 2)
```

- An environment is a sequence of frames
- An environment for a non-nested function (no def within def) consists of one local frame, followed by the global frame
Environment Diagrams for Nested Def Statements

- Every user-defined function has a parent frame
- The parent of a function is the frame in which it was defined
- Every local frame has a parent frame
- The parent of a frame is the parent of the function called

The Structure of Environments

A frame extends the environment that begins with its parent

How to Draw an Environment Diagram

When defining a function:
1. Create a function value with signature
   `<name>(<formal parameters>)`
2. For nested definitions, label the parent as the first frame of the current environment
3. Bind `<name>` to the function value in the first frame of the current environment

When calling a function:
1. Add a local frame labeled with the `<name>` of the function
2. If the function has a parent label, copy it to this frame
3. Bind the `<formal parameters>` to the arguments in this frame
4. Execute the body of the function in the environment that starts with this frame

The Environment for Function Composition

Lambda Expressions

```python
>>> ten = 10
>>> square = x * x
>>> square = lambda x: x * x
>>> square(4)
16
```

Lambda expressions are rare in Python, but important in general