Announcements

- Homework 1 due Wednesday 9/10 at 2pm. Late homework is not accepted!
- Homework parties on Monday 9/8 (Today!)
  - 3pm-4pm in Wozniak Lounge in Soda Hall (100 person capacity)
  - 6pm-8pm in 2050 Valley Life Sciences Building (408 person capacity)
- More sections for students without prior programming experience! http://cs61a.org
- Take-home quiz 1 starts Wednesday 9/10 at 3pm, due Thursday 9/11 at 11:59pm
  - Open-computer, but no external resources or friends
- Content Covered: Lectures through last Friday 9/5 (same topics as Homework 1)
- Project 1 due next Wednesday 9/17 at 11:59pm

The Fibonacci Sequence

\[ 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987 \]

The next Fibonacci number is the sum of the current one and its predecessor.

Discussion Question 1

- What does pyramid compute?

Designing Functions

Characteristics of Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>def square(x):</code></td>
<td><strong>&quot;Return X * X.&quot;</strong></td>
</tr>
<tr>
<td><code>return x**2</code></td>
<td><code>x</code> is a real number</td>
</tr>
<tr>
<td><code>def fib(n):</code></td>
<td><strong>&quot;Compute the nth Fibonacci number, for N &gt;= 1.&quot;</strong></td>
</tr>
<tr>
<td><code>pred, curr = 0, 1</code></td>
<td><code>n</code> is an integer greater than or equal to 1</td>
</tr>
<tr>
<td><code>while k &lt; n:</code></td>
<td></td>
</tr>
<tr>
<td><code>k = k + 1</code></td>
<td></td>
</tr>
</tbody>
</table>

A Guide to Designing Function

- Give each function exactly one job.
- Don't repeat yourself (DRY). Implement a process just once, but execute it many times.
- Define functions generally.
Generalization

Higher-Order Functions

Generalizing Patterns with Arguments

Regular geometric shapes relate length and area.

Shape:

Area:

Finding common structure allows for shared implementation

(Demo)

Generalizing Over Computational Processes

The common structure among functions may be a computational process, rather than a number.

(Demo)

Summation Example

Functions as Return Values

Locally Defined Functions

Call Expressions as Operator Expressions
The Purpose of Higher-Order Functions

Functions are first-class: Functions can be manipulated as values in our programming language.

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

Higher-order functions:
- Express general methods of computation
- Remove repetition from programs
- Separate concerns among functions

The Game of Hog (Demo)