Welcome to Berkeley Computer Science!

61A Lecture 1
Friday, August 24, 2012

The Course Staff

The Course Staff

What is Computer Science?

Systems
Artificial Intelligence
Graphics
Security
Networking
Programming Languages
Theory
Scientific Computing
...

What is 61A?

What is 61A?

- A course about managing complexity
  - Mastering abstraction
  - Not about 1's and 0's
  - An introduction to Python
    - All the features we really need: introduced today
    - Understanding through implementation
    - Programs that run other programs: meta-evaluation
  - A challenging course that will demand a lot of you

Plone Conference. Photo courtesy of Kriszta Szita
Alternatives to 61A

CS 61AS: Self-paced 61A

CS 10: The Beauty and Joy of Computing

Course Policies

The purpose of this course is to help you learn

The staff is here to make you successful

All the details are online:
http://inst.eecs.berkeley.edu/~cs61A/fa12/about.html

Collaboration

• Discuss everything with each other
• EPA: Effort, participation, and altruism
• Homework can be completed with a partner
• Projects should be completed with a partner
• Find a project partner in your section!

The limits of collaboration

• One simple rule: don’t share code
• Copying project solutions is a serious offense!

Announcements

• Next week, both section and lab will meet in the lab rooms.
• Homework 1 is posted! All homework is graded on effort.
• If you are on the waitlist, still complete assignments!
• Midterms are on 9/19 and 10/24. Final exam is on 12/13.
• Read the lecture notes before you come to lecture!

Types of expressions

An expression describes a computation and evaluates to a value

18 + 69
6
23
f(x)
| − 1869|

Call Expressions in Python

All expressions can use function call notation

(Demo)
Anatomy of a Call Expression

\[
\begin{array}{ccc}
\text{add} & ( & 2 & , & 3 & ) \\
\text{Operator} & \text{Operand 0} & \text{Operand 1}
\end{array}
\]

Operators and operands are expressions
So they evaluate to values

Evaluation procedure for call expressions:
1. Evaluate the operator and operand subexpressions
2. Apply the function that is the value of the operator subexpression to the arguments that are the values of the operand subexpression

Evaluating Nested Expressions

Data, Functions, and Interpreters

**Data:** The things that programs fiddle with

- "The Art of Computer Programming"
- Donald Knuth
- Shakespeare's 37 plays
- (Knuth)

**Functions:** Rules for manipulating data

- Count the words in a line of text
- Add up numbers
- Pronounce someone's name

**Interpreter:** An implementation of the procedure for evaluation