

61A Lecture 1

Friday, August 29, 2014

Welcome to Berkeley Computer Science!



Fall 2014 office hours:

411 Soda

Tuesday 12pm–1pm

Wednesday 12pm–1pm

781 Soda by appointment

<http://denero.org/meet>

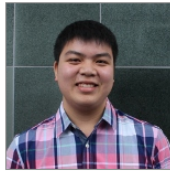


The Course Staff

Teaching Assistants (UGSIs/GSIs) run discussion sections, labs, and office hours.



Soumya Basu



Matthew Chow



Ajeya Cotra



Brian Hou



Andrew Huang



Robert Huang



Michelle Hwang



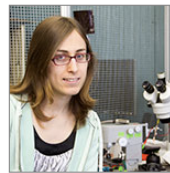
Mehdi Jamei



Joy Jeng



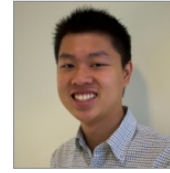
Chloe Lischinsky



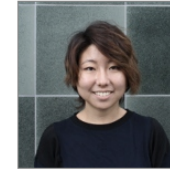
Kaylee Mann



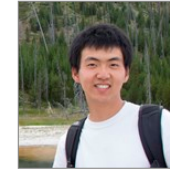
Beth Marrone



Allen Nguyen



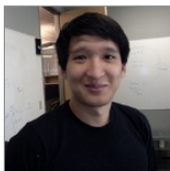
Youri Park



Jack Qiao



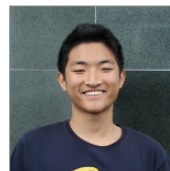
Sumukh Sridhara



Steven Tang



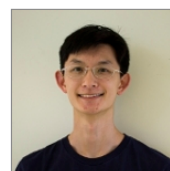
Michael Tao



Dickson Tsai



Iris Wang



Albert Wu



Chenyang Yuan



Marvin Zhang

18 Readers are your personal programming mentors.

Over 150 Lab Assistants ensure that you don't get stuck for too long.

Parts of the Course

Lecture: Videos posted to <http://cs61a.org> before each live lecture

Lab: The most important events in this course

Discussion: Also the most important events in this course

Office Hours: Also the most important events in this course [11–5 every day in 411 Soda]

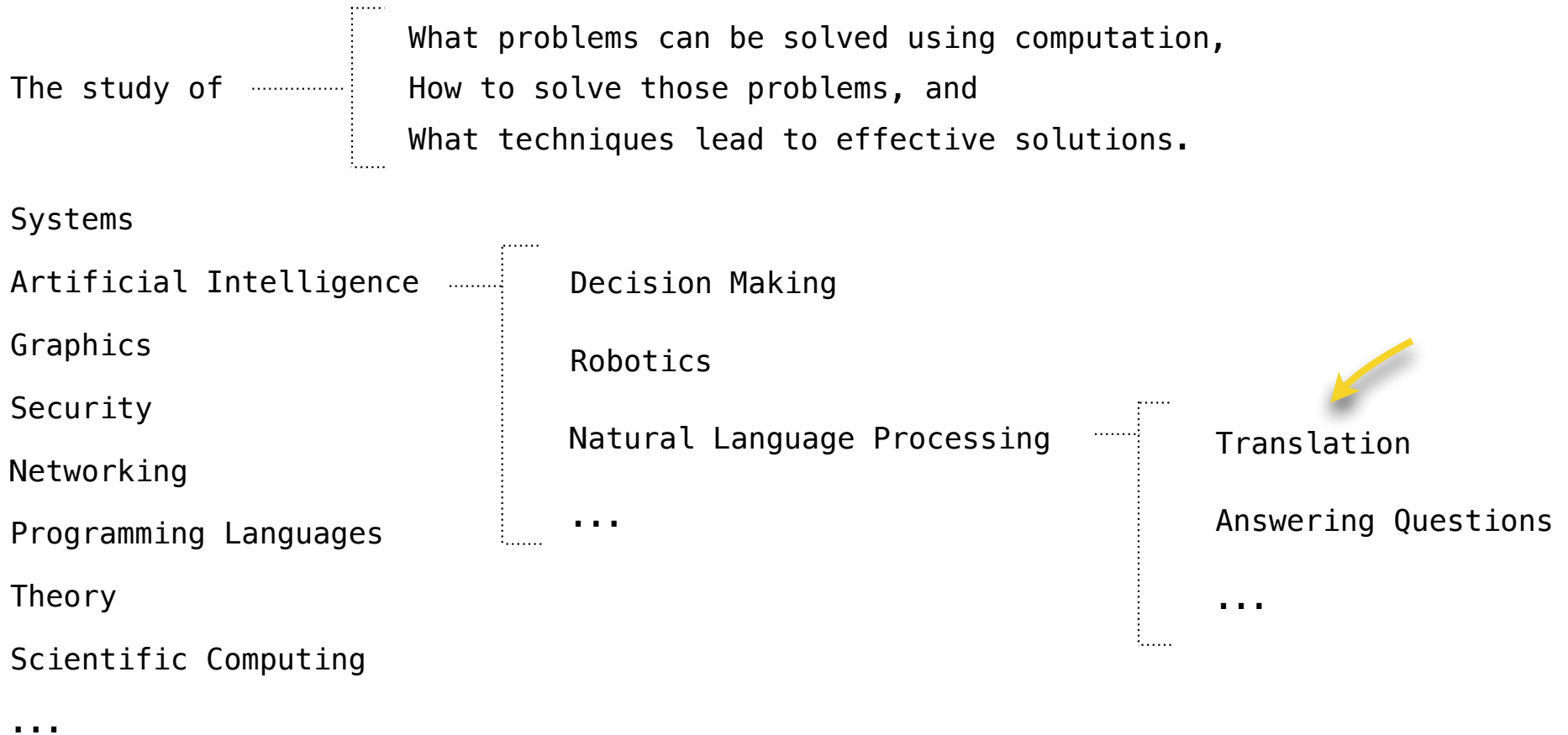
Online textbook: <http://composingprograms.com>

Weekly homework assignments, three exams, & four programming projects

Many special events

An Introduction to Computer Science

What is Computer Science?



What is This Course About?

- A course about managing complexity
 - Mastering abstraction
 - Programming paradigms
 - Not all about 0's and 1's
- An introduction to Python
 - Full understanding of language fundamentals
 - Learning through implementation
 - How computers interpret programming languages
- A challenging course that will demand a lot of you



Course Policies

Alternatives to This Course

CS 61AS: Self-Paced 61A

CS 10: The Beauty and Joy of Computing

Learning
Community
Course Staff

Details...

<http://cs61a.org/about.html>

Collaboration

Asking questions is highly encouraged

- Discuss everything with each other; learn from your fellow students!
- Homework can be completed with a partner
- Projects should be completed with a partner
- Choose a partner from your discussion section

The limits of collaboration

- One simple rule: Don't share your code, except with your partner
- Copying project solutions causes people to fail this course
- We really do catch people who violate the rules, because...
 - We also know how to search the web for solutions
 - We know how to use computers

Build good habits now

Expressions

Types of expressions

An expression describes a computation and evaluates to a value

$$18 + 69$$

$$\frac{6}{23}$$

$$\sin \pi$$

$$\log_2 1024$$

$$2^{100}$$

$$f(x)$$

$$\sqrt{3493161}$$

$$7 \bmod 2$$

$$\sum_{i=1}^{100} i$$

$$\lim_{x \rightarrow \infty} \frac{1}{x}$$

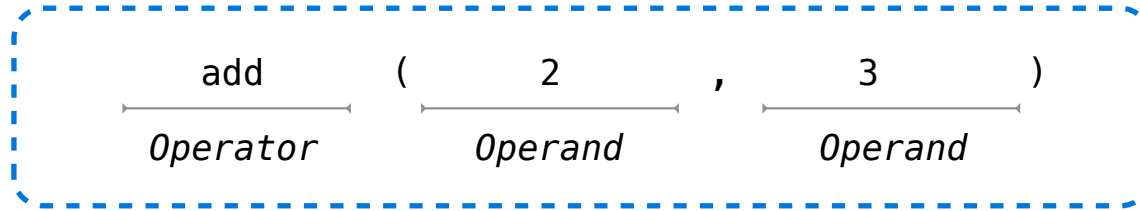
$$|-1869|$$

$$\binom{69}{18}$$

Call Expressions in Python

All expressions can use function call notation
(Demo)

Anatomy of a Call Expression



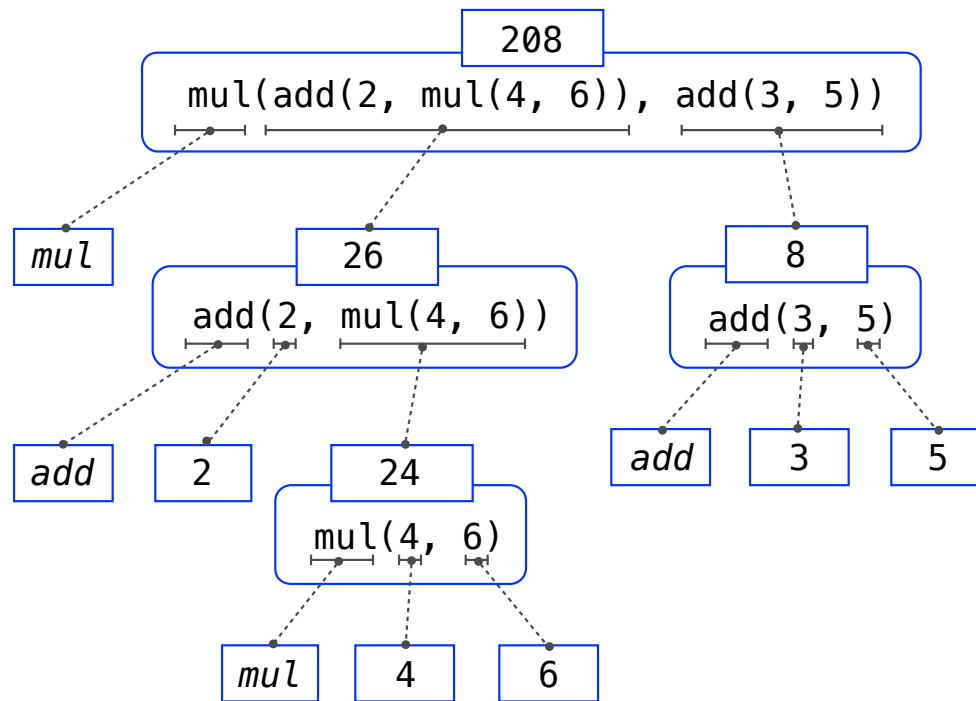
Operators and operands are also expressions

So they evaluate to values

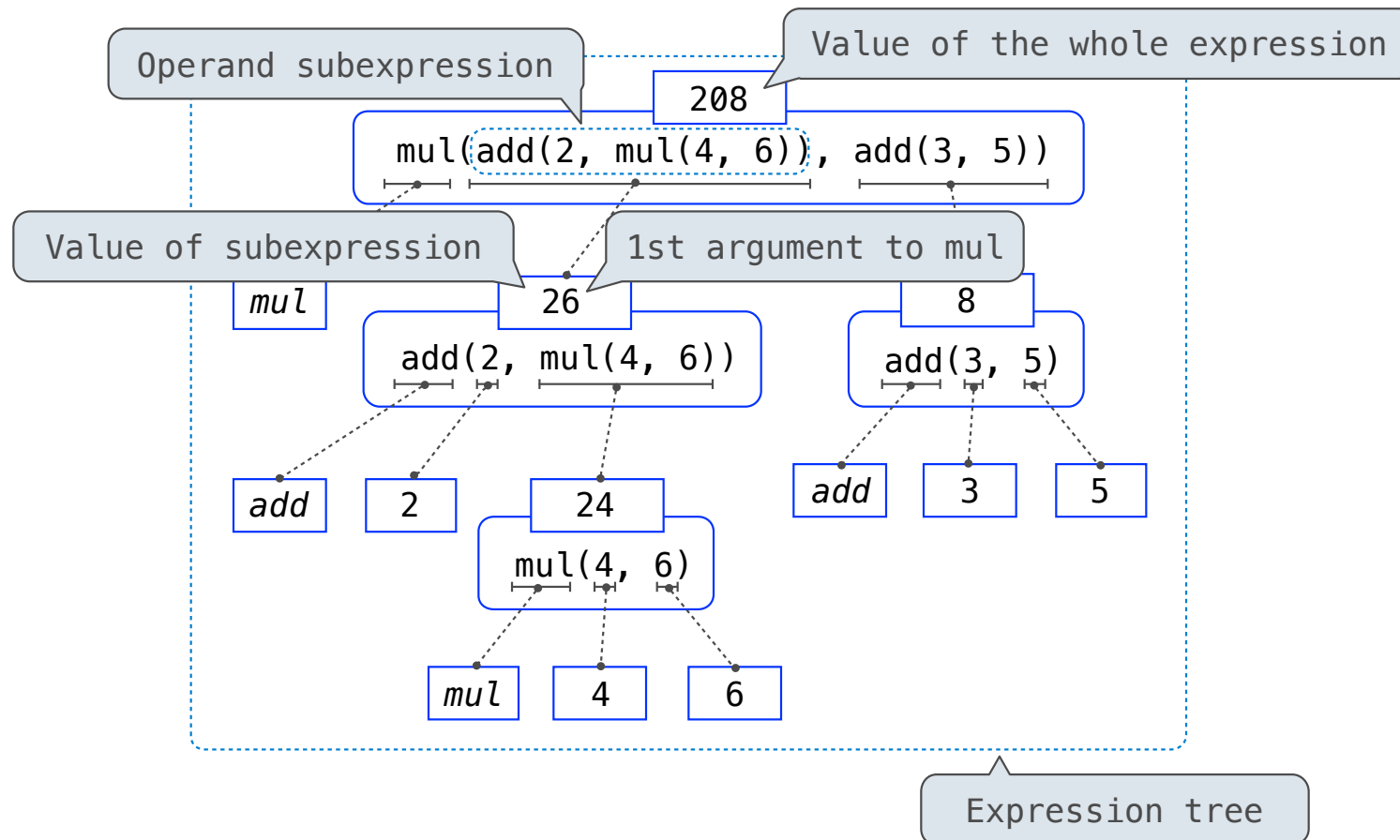
Evaluation procedure for call expressions:

1. Evaluate the operator and then the 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



Evaluating Nested Expressions



Functions, Objects, and Interpreters

(Demo)