61A Lecture 1

Wednesday, January 21, 2015

Welcome to Berkeley Computer Science!

BERKELEY EECS

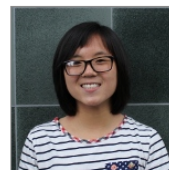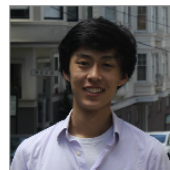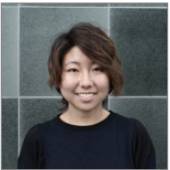Soda Hall

John DeNero

Spring 2015 office hours:

781 Soda
Wednesday 10am–12pm &
Friday by appointment:
http://denero.org/meet

The Course Staff

Teaching Assistants (GSIs/UGSIs) run discussion sections, labs, and office hours
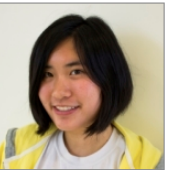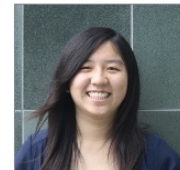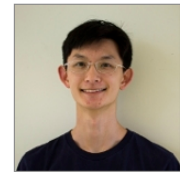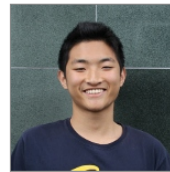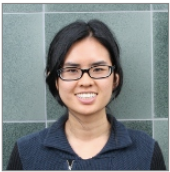
27 **Group Tutors** are your personal programming mentors

Over 300 **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

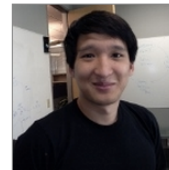**Lab:** The most important events in this course

Discussion: Also the most important events in this course

Office Hours: Also the most important events in this course [11–5 M–Th & 11–1 Friday]

Online textbook: http://composingprograms.com

Weekly homework assignments, three exams, three quizzes, & four programming projects

Lots of special events

An Introduction to Computer Science

What is Computer Science?

The study of ········
- What problems can be solved using computation,
- How to solve those problems, and
- What techniques lead to effective solutions

Systems

Artificial Intelligence ········
- Decision Making
- Robotics
- Natural Language Processing ········
  - Translation
  - Answering Questions
  - ...
- ...

Graphics

Security

Networking

Programming Languages

Theory

Scientific Computing

...
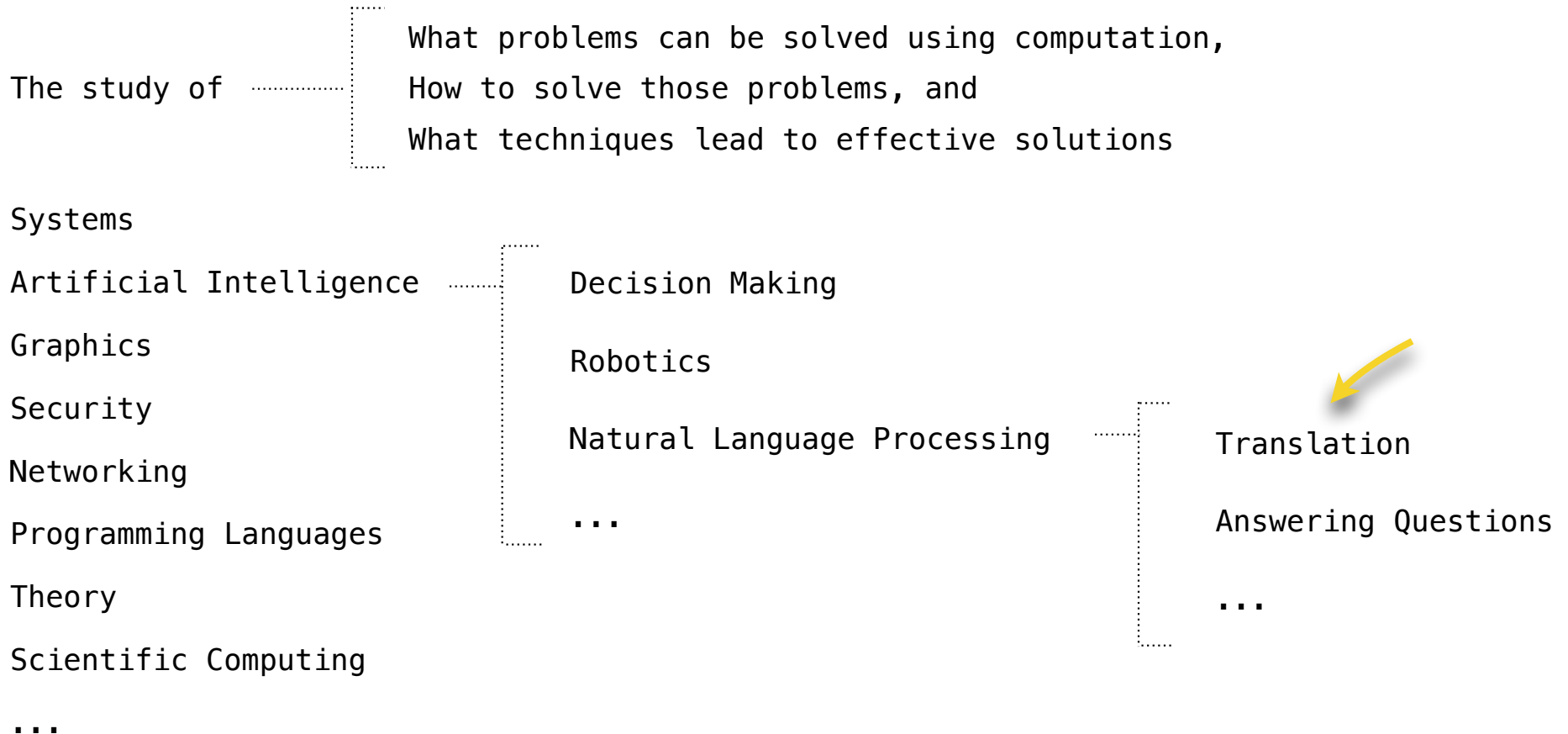
What is This Course About?

- A course about managing complexity

  - Mastering abstraction

  - Programming paradigms

  - Not just 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

CS 61AS: Self-Paced CS 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 use computers to check your work

Build good habits now

Expressions

An expression describes a computation and evaluates to a value

$18 + 69$

$\dfrac{6}{23}$

$\sin \pi$

$\log_2 1024$

2^{100}

$f(x)$

$\sqrt{3493161}$

$7 \bmod 2$

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

$\displaystyle\lim_{x \to \infty} \dfrac{1}{x}$

$|-1869|$
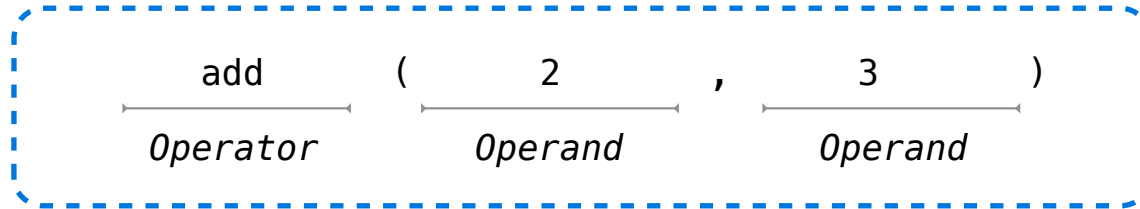
$\dbinom{69}{18}$

Call Expressions in Python

All expressions can use function call notation

(Demo)

Anatomy of a Call Expression

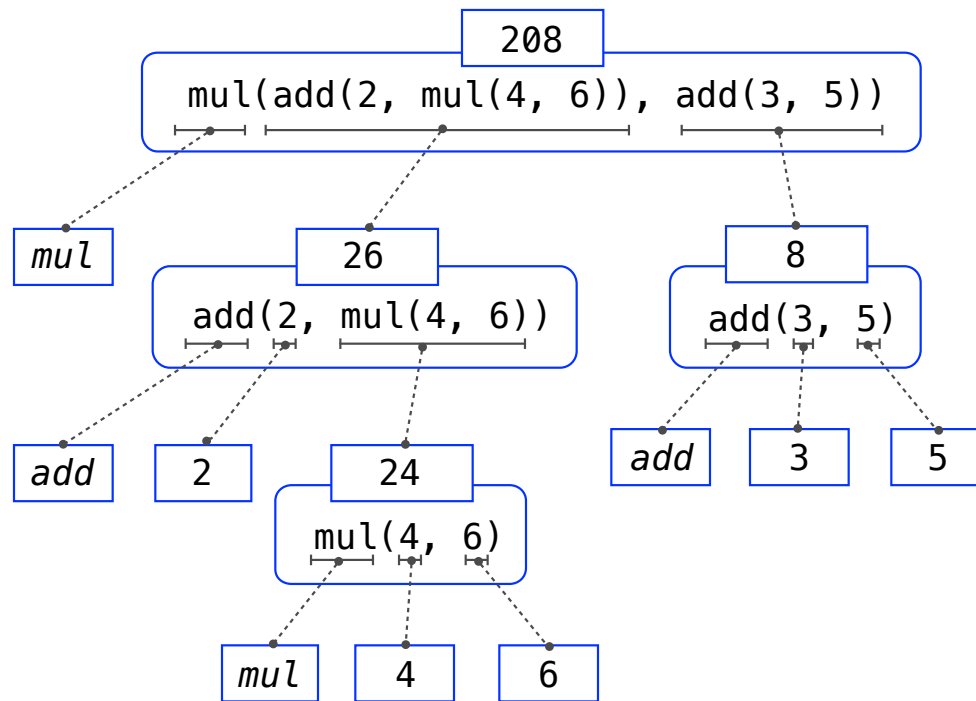| add	(	2	,	3	)
Operator		*Operand*		*Operand*	

Operators and operands are also expressions
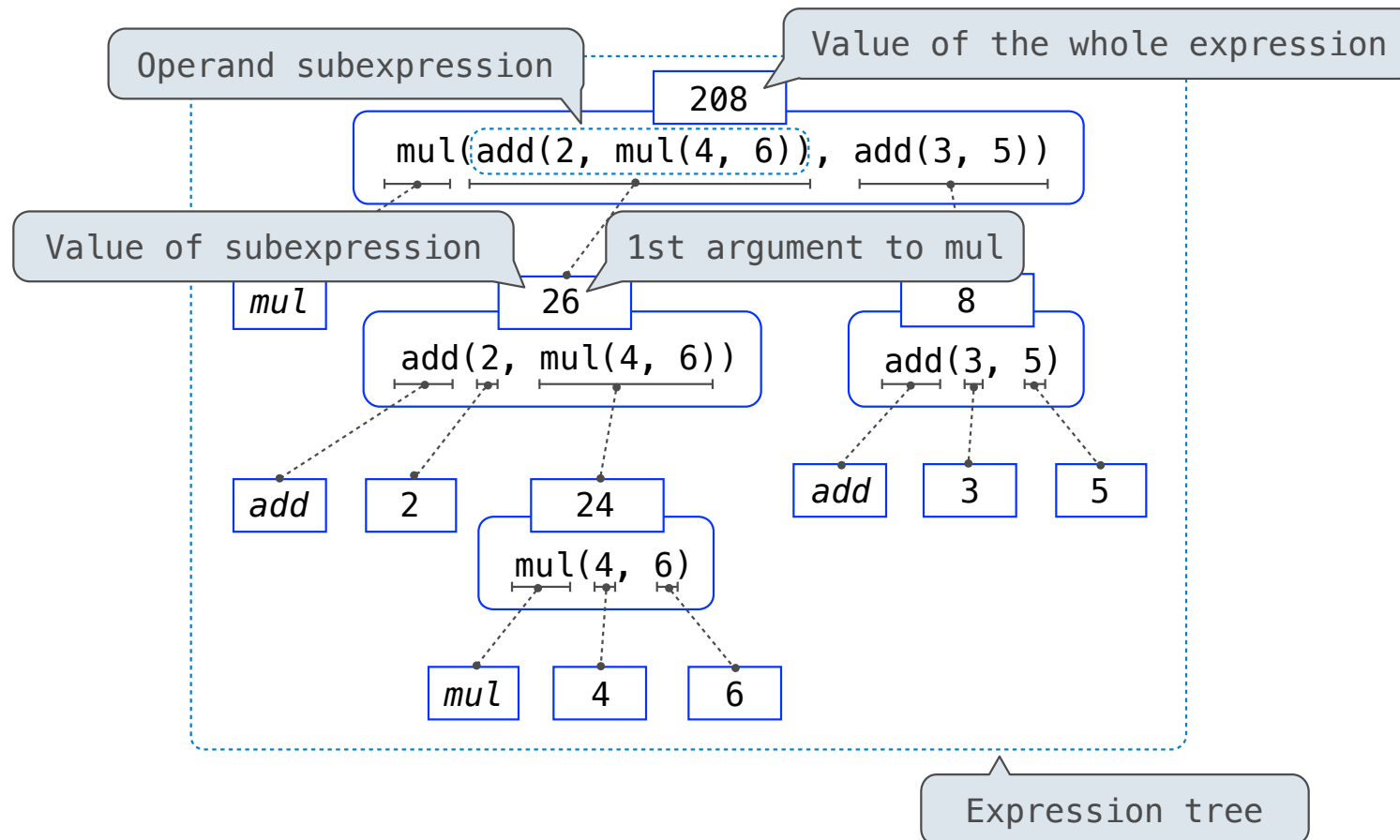
So they evaluate to values

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

Operand subexpression

Value of the whole expression

208

mul(add(2, mul(4, 6)), add(3, 5))

Value of subexpression

1st argument to mul

mul

26

add(2, mul(4, 6))

8

add(3, 5)

add   2   24

add   3   5

mul(4, 6)

mul   4   6

Expression tree

Functions, Objects, and Interpreters

(Demo)