

Lecture will begin at 1:10pm.

Looking for something to do?

The course website is cs61a.org/denero.html

Welcome to CS 61A!

About Me

John DeNero

denero@berkeley.edu

CS 61A instructor every Fall since 2011 (and some Spring semesters as well)

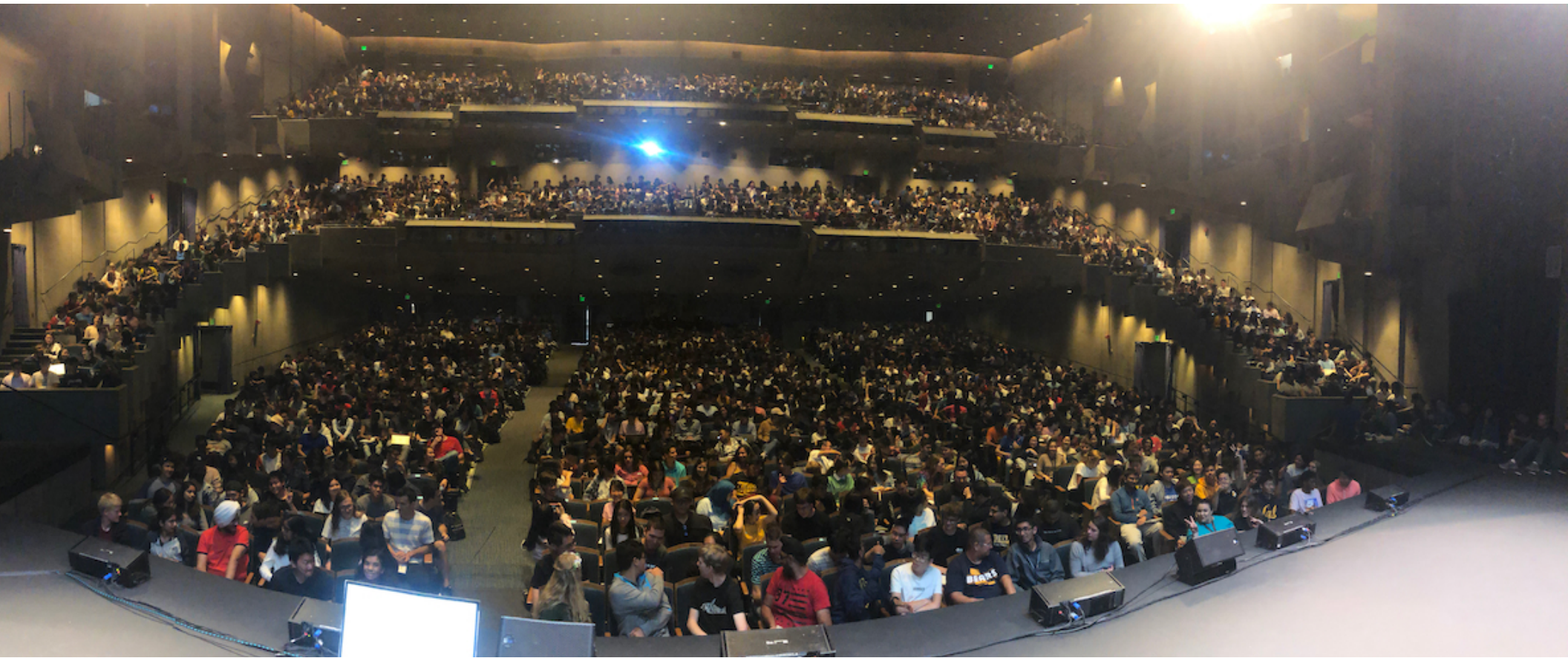
Before that I was a Cal PhD student: 2005–2010

Teaching Professor in EECS & Faculty Director of Data Science Undergraduate Studies

Research focused on machine translation and how people interact with AI systems

My office hours start next week (Aug 30):

- Monday & Wednesday 2:15–3pm in 110 Warren Hall
- If that time doesn't work, email me and we'll set up an appointment



About the Course

Lecture, Videos, and the Textbook

Videos posted to cs61a.org/denero.html are essential viewing **before** coming to lecture. All of the course content will be covered in the videos.

The **textbook**, composingprograms.com, is written to be concise and useful. Its content is very similar to the videos.

New this semester: **Lecture** Mon, Wed, & Fri will review *the most important content* from the videos, work through examples, and discuss problem-solving strategies.

Problem-Solving Practice

Solving problems is an effective way to learn how to solve problems.

Lab on Tuesday: attendance is recommended (but not required)

Discussion on Thursday: attendance is required (unless you're in the mega section)

These prepare you for weekly **homework** assignments & 4 larger programming **projects**

Drop-in one-on-one assignment help (called "**office hours**" at Cal) starts next week.

New this semester:

- I'll be teaching lab along with a great TA named Amy.
- The experimental discussion format I told you about...

What does a "discussion section" look like?

Expectation



<https://engineering.berkeley.edu/students/academic-support/>

Reality



<https://www.microsoft.com/en-us/research/blog/grassroots-data-science-education-uc-berkeley/>

Goal: Provide a great environment to learn how to solve problems through practice & *discussion*

I think this will work because I've seen small groups of Cal students do amazing things!

Small-Group Discussions

Unless you've elected the mega discussion...

- You should have a group number shared by the 6–7 students in your group, a room, and a discussion time. There will be 2 groups per room.

Coming soon: A Discord voice/chat channel just for your group and the course staff

What happens during discussion section?

- You're given a worksheet full of example problems to solve together & some instructions.
- The point is not just to solve those problems, but to learn how to solve similar problems.
 - Discussion problems aren't graded; you don't have to solve them all.
- At least one of you will open Discord so you all can talk/text with the course staff.
 - Message @discuss in your group's voice channel and Shafeeq, Steffi, or I will join.
 - A laptop or tablet is recommended but not required. Otherwise, bring paper & pencil.

Asking Questions

?

Ed: You can reach all staff (private posts) and all students (public posts)

denero@berkeley.edu: Don't be surprised if I ask you to post on Ed

jiyeonwoo@berkeley.edu: Jenna Woo is our course manager and handles logistics

What is This Course About?

A course about managing complexity

Mastering abstraction

Techniques for organizing complex programs

An introduction to programming

Full understanding of Python fundamentals

Large projects to demonstrate how to manage complexity

How computers interpret programming languages

Different types of languages: Python, Scheme, & SQL

Should you take CS 61A?

[According to the Syllabus: cs61a.org/articles/about/](https://cs61a.org/articles/about/)

There is no formal programming-related prerequisite for CS 61A, but...

- Taking the course without any prior programming experience is typically quite challenging.
- Most CS 61A students have had significant prior programming experience.
- Students who take the course without prior programming experience typically must spend more time to complete assignments and tend to receive lower final grades in the course.

Data 88C (Formerly CS 88): Computational Structures in Data Science

Based on CS 61A, but covers only 3 out of 4 units worth of the content:

- Two programming projects (instead of four) that are adapted from CS 61A projects
- Everything you need to know to continue on to CS 61B
- Omits the unit on how programs run other programs

Designed for students taking Data 8 (Foundations of Data Science), but is now independent

The course is full, but we're investigating expansion options.

Course Policies

Learning Community

Details...

<https://cs61a.org/articles/about/>

Collaboration

Working together is highly encouraged

- Discuss everything with each other; learn from your fellow students!
- Some projects can be completed with a partner

What constitutes academic misconduct?

- *Please* don't look at someone else's code!
Exceptions: lab, your project partner, or **after you already solved the problem.**
- *Please* don't tell other people the answers! You can point them to what is wrong and describe how to fix it or show them a related example.
- *Please* don't ask ChatGPT or other similar tools to write code for you.
- Copying project solutions causes people to fail the course.

Build good habits now

Let's Stop Harassment & Discrimination

Disparaging remarks targeting a particular gender, race, or ethnicity are not acceptable.

From the Berkeley Principles of Community:

"We affirm the dignity of all individuals and strive to uphold a just community in which discrimination and hate are not tolerated."

From the EECS department mission:

"Diversity, equity, and inclusion are core values in the Department of Electrical Engineering and Computer Sciences. Our excellence can only be fully realized by faculty, students, and staff who share our commitment to these values."

denero.org/feedback.html: If you want to stay anonymous but make me aware of something happening in the course.

EECS Student Climate & Incident Reporting Form: Informs the EECS department of any issues. You can also contact Susanne Kauer (skauer@berkeley.edu) directly.

The Best Approach to CS 61A

Help each other understand concepts in the class, whether in section, on Ed, or in study groups, without expectation of anything in return.

Be great project partners by listening to what your partner suggests and helping them understand the work you've done together.

Don't fall behind yourself, don't encourage others to fall behind, and ask for help when you need it.

Recognize that we're all valuable members of the CS community!

Expressions

Types of expressions

An expression describes a computation and evaluates to a value

$$18 + 69$$

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

$$\sin \pi$$

$$\log_2 1024$$

$$2^{100}$$

$$f(x)$$

$$\sqrt{3493161}$$

$$7 \bmod 2$$

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

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

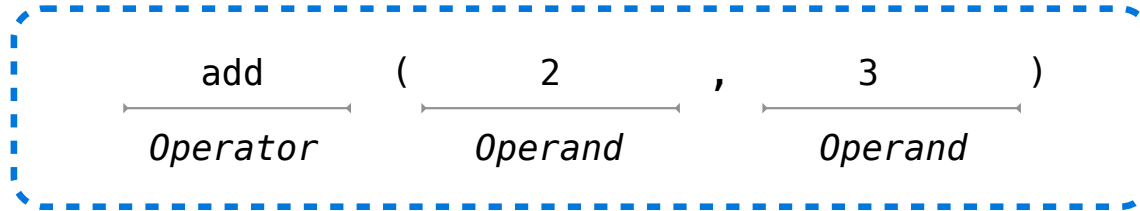
$$|-1869|$$

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

Call Expressions in Python

All expressions can use function call notation
(Demo)

Anatomy of a Call Expression



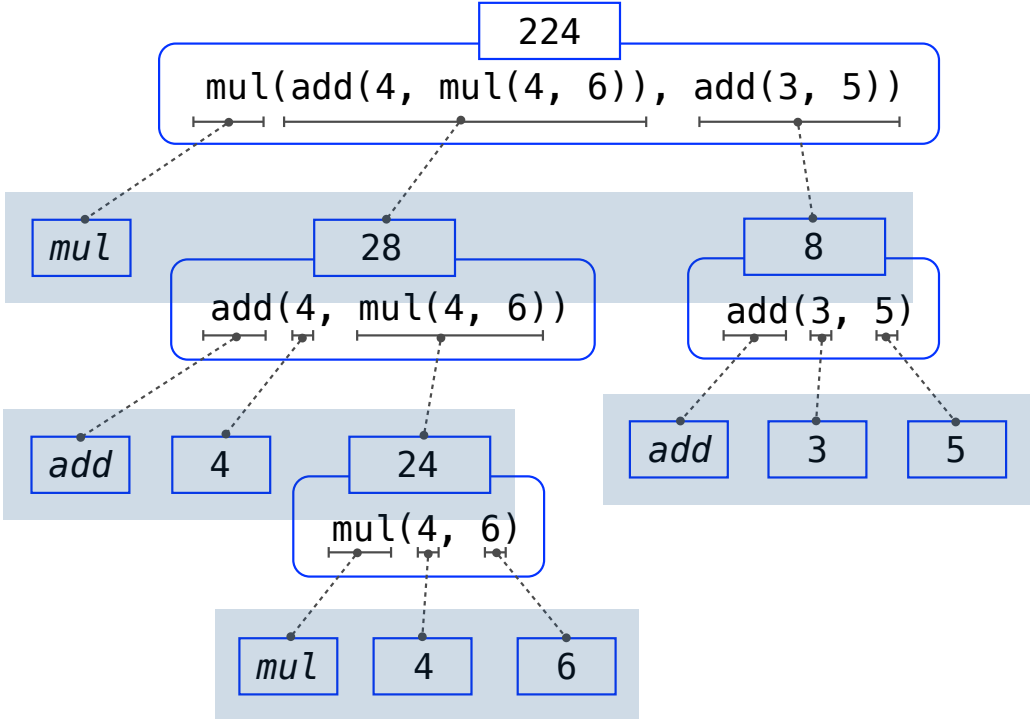
Operators and operands are also expressions

So they evaluate to values

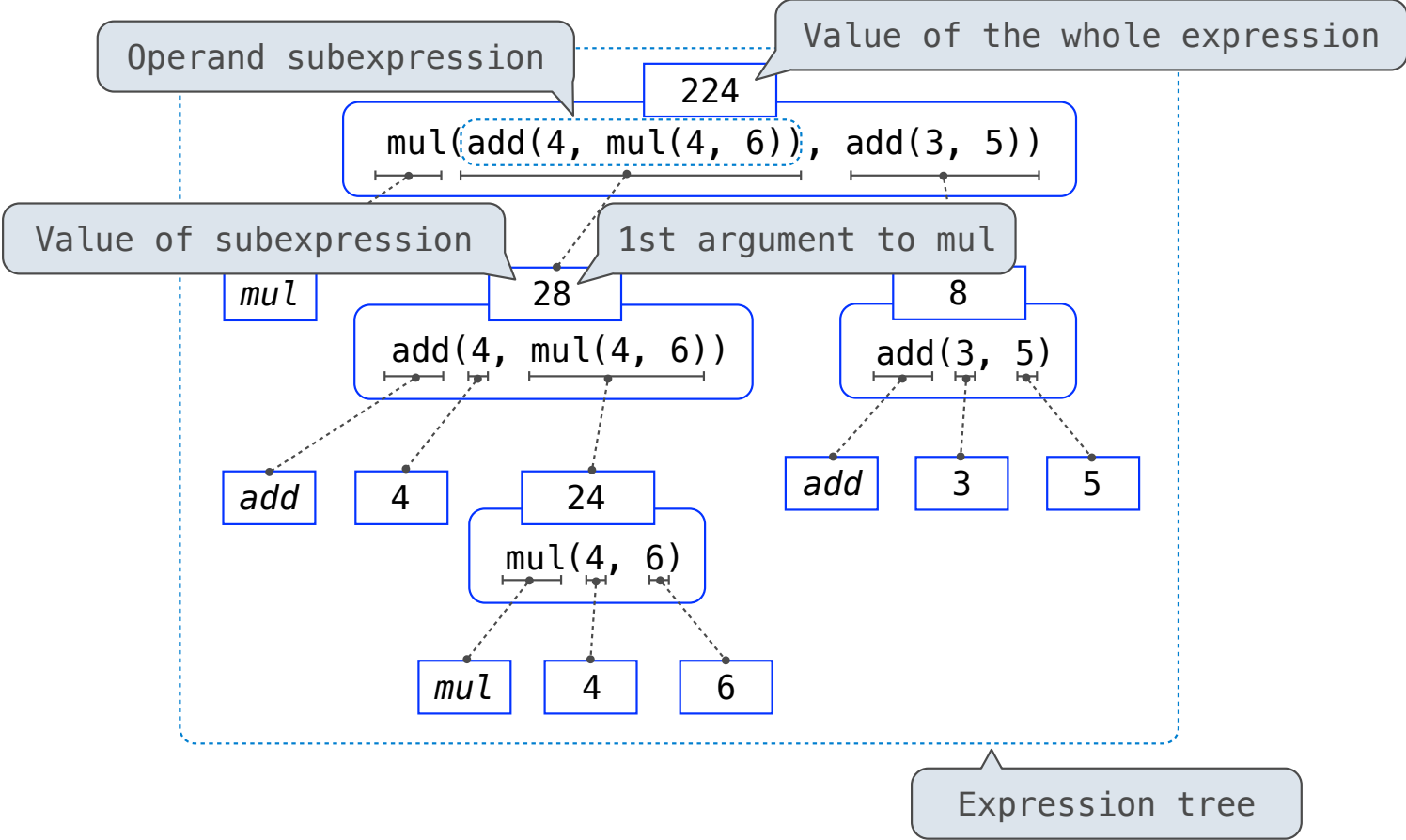
Evaluation procedure for call expressions:

1. Evaluate the operator and then the operand subexpressions
2. Apply the function that is the value of the operator to the arguments that are the values of the operands

Evaluating Nested Expressions



Evaluating Nested Expressions



Composing Programs

(Demo)