

Welcome to CS 61A!

Berkeley Time: All lectures, sections, etc. will start 10 minutes past the official start time, as is Berkeley tradition.

In the meantime, check out this word cloud of CS 61A students' interests and meet the person next to you!



Humans of CS 61A

About Me

Laryn Qi

larynqi@berkeley.edu

Cal alum! Student from 2019–2024

- BA CS & Music, MS in EECS
- Research: AI tools in CS Education
- Teaching: 61A TA

3rd summer lecturing for 61A

Machine Learning @ Amazon

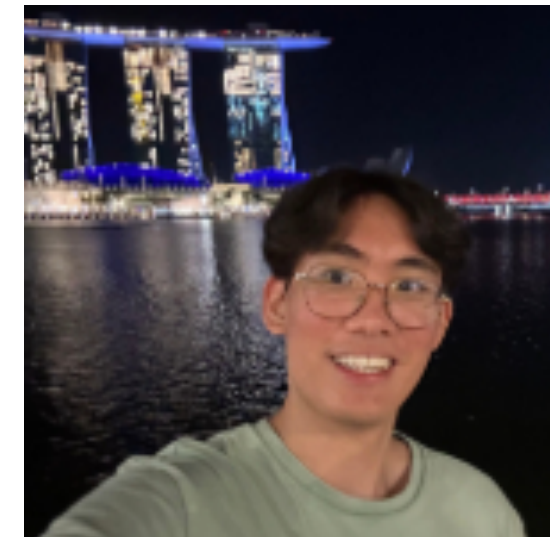
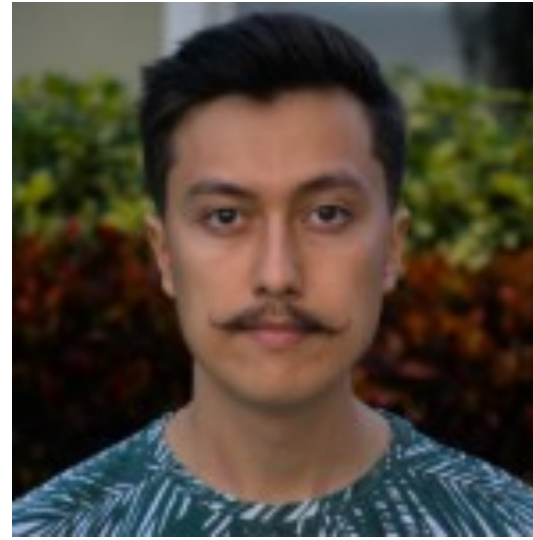
- Prime Video Recommendations

Interests: drums, snowboarding, Knicks

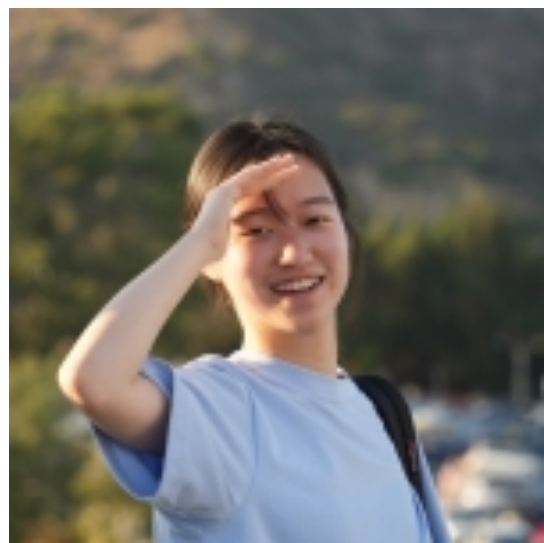
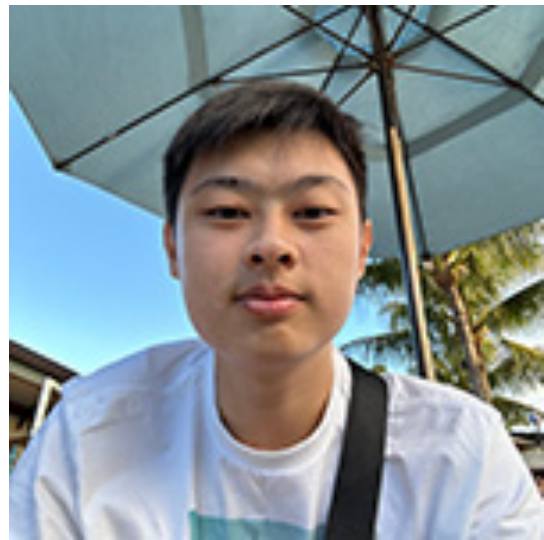
My office hours start this week:

- Tu & Wed: 11–12pm & 3–4pm in 795 Soda Hall
- After this week, they will be by-appointment (Calendly)

TAs!



Tutors!



About the Course

What is This Course About?

A course about managing complexity

Mastering abstraction

Isolating and solving problems

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

cs61a.org

How to Succeed

Lecture, Videos, and the Textbook

Videos posted to cs61a.org 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.

Lecture Mon, Tue, Wed, & Thu will review *the most important content* from the videos (but not all of it), work through examples, and discuss problem-solving strategies.

- Week 1: Hybrid – live lecture in Soda 306 + Zoom broadcast (recorded)
- After week 1: Zoom only (also recorded)

Student Advice from Fall 2024

"Watch videos before lecture"

"Watch the videos. Definitely helps with understanding the lecture beforehand, and the reason why I was so lost during the second half of the semester."

"Obviously watch the videos, try not to watch it at 2x speed (which is what I did and regret)...if you don't take time processing information, it will leave your brain by next morning"

"keep up with lectures, watch the videos, try to really understand everything along the way so it doesn't pile up at the end"

"Make sure to watch lecture videos before the lectures, so that lectures can be utilized for asking questions and further understanding."

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

Problem-Solving Practice

Solving problems becomes easier with **practice**.

Lab on Mon & Wed: attendance is required (unless you're in mega lab)

- Lab *assignment* must be completed correctly and submitted regardless of section type

Discussion on Tue & Thu: attendance is required (unless you're in mega discussion)

- Nothing to submit

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

- Homeworks are completed on your own
- Projects can be completed with a project partner

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

Discussion (Starts Tomorrow)

Unless you've elected the mega discussion...

- Begins Tuesday (**tomorrow**)
- ~20 students, led by 1 TA, 90 minutes

What happens during discussion section?

- You're given a worksheet full of conceptual review and example problems to solve together.
- Collaborate with your peers and TA to go through the worksheet
- Your TA will review important concepts from lecture, give leading questions, walk through solutions, and discuss problem solving strategies
- 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

Lab 0: Computer Setup (Starts This Week)

Lab 0 is posted. You can get started on it, but you will work through it with your TA in *lab section* on Wednesday

Lab section goal: dedicated time to complete the lab assignment with your lab partner and TA guidance

90 minutes of collaborative programming practice to set you up for success with homeworks and projects

Drop-in "Technical" office hours to get help with Lab 0 on Thursday.

<https://cs61a.org/office-hours/>

Weekly Schedule

| | Monday | Tuesday | Wednesday | Thursday |
|----------|---------------------------|---------------------------|---------------------------|---------------------------|
| Day | Lab | Discussion | Lab | Discussion |
| 5-6:30pm | Lecture | Lecture | Lecture | Lecture |
| Night | Watch next lecture videos | Watch next lecture videos | Watch next lecture videos | Watch next lecture videos |

Mega Section

No-attendance alternative, self-guided approach **using the same assignments**

Asynchronous – no daily meetings with a TA and other students

Lab:

- "Getting Started" hint videos will be posted along with each lab assignment
- Must solve all required problems correctly to get credit

Discussion:

- Full walkthrough videos will be posted along with each discussion worksheet on cs61a.org
- Your responsibility to go through the worksheet and watch the videos
- Attendance credit automatically given

Regular or Mega?

You should choose **regular** if you like learning in a traditional classroom environment with other students and a TA.

- The majority of students usually choose regular
- Students with less/no prior experience usually choose regular

You should choose **mega** if you are confident in learning at your own pace

Space in regular sections is limited by the number of TAs we have

Deadline to switch sections: 7/4 end-of-day (end of Week 2)

sections.cs61a.org

Asking Questions

?

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

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

cs61a@berkeley.edu: Goes to several staff members

Student Advice from Fall 2024

"Try to collaborate and with others and try to make friends within the class."

"Attend lab and make sure you understand how each program is structured"

"really utilize discussions. learning is easier with others!"

"Attend discussions and labs as it helps you discuss the content with other students"

"Go to discussion. I am so, so grateful for the fact that I had an active discussion group. Make the tough choice and commit yourself to spending that hour each week solving problems with other people just as confused as you are-- I guarantee that you'll look back on the decision and have no doubt that it was worthwhile."

"ASK OTHER PEOPLE FOR HELP WHEN NEEDED, DON'T BE AFRAID TO MAKE FRIENDS!!!"

So... Can I Take CS 61A Remotely?

Every part of the course has a remote-friendly option **except exams**

- Midterm: 7/14, 5–7pm (tentative)
- Final: 8/12, 11:30am–2:30pm (finalized)
- Exams will be administered **in-person only**

- Lectures: Video playlists, Zoom option + recordings
- Sections: Mega
- Assignments: Completed on your own and submitted online
- Questions: Ask on Ed

Grading

Grading

We use **grade bins** – not curved (everyone could get an A)

We are not planning to change these

| | | | | | |
|----|------------|------------|------------|------------|------------|
| | A | ≥ 285 | A- | ≥ 270 | |
| B+ | ≥ 255 | B | ≥ 230 | B- | ≥ 210 |
| C+ | ≥ 190 | C | ≥ 180 | C- | ≥ 175 |
| D+ | ≥ 170 | D | ≥ 165 | D- | ≥ 160 |

A+ will be given to those who perform exceptionally well in the course, at the discretion of the instructor (me)

Exams

Midterm

- 64 points (out of 300 total in the course)
- ~Halfway through the summer session (7/14)
- 2 hour paper exam – no devices

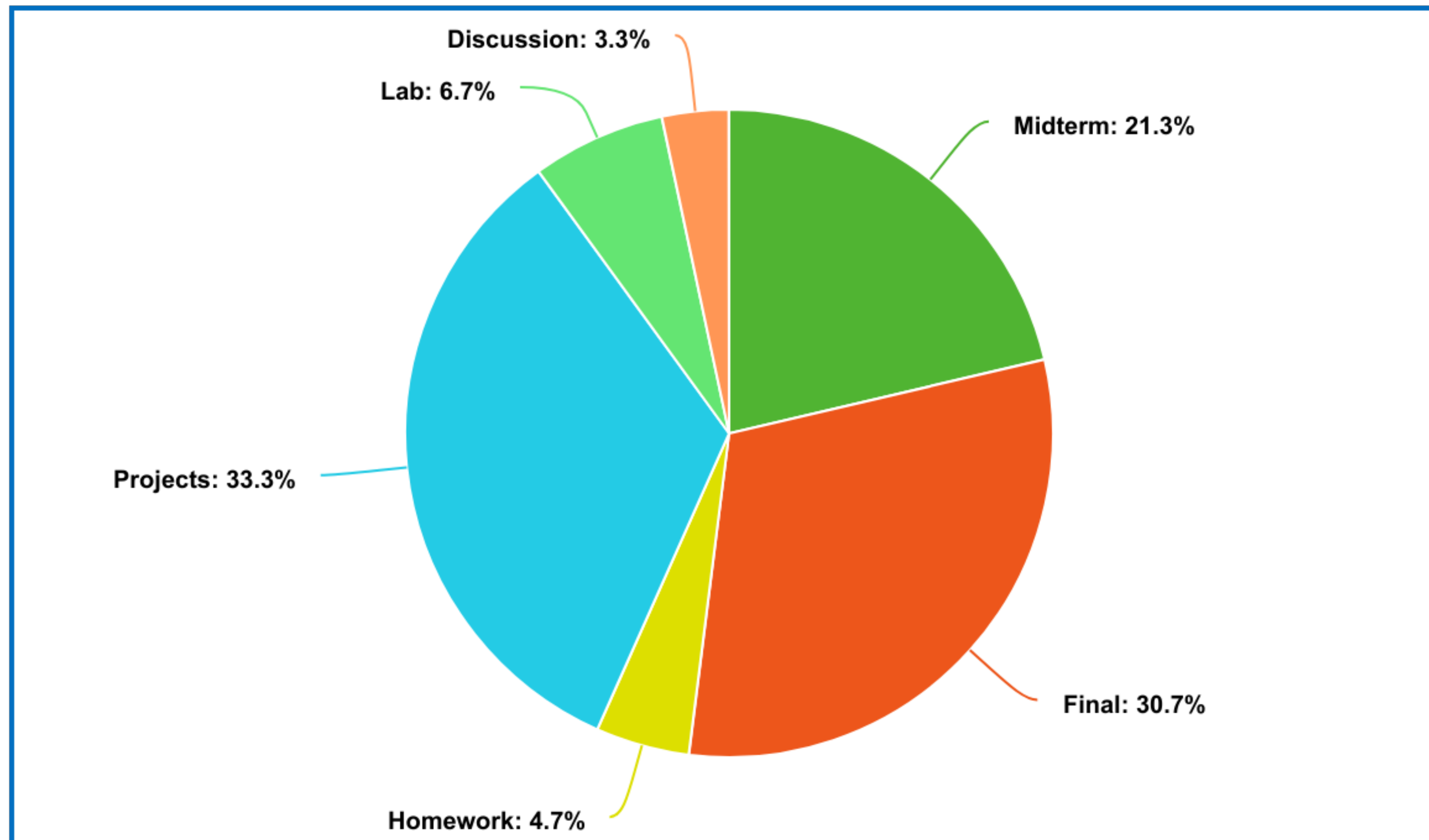
Final

- 92 points (out of 300 total in the course)
- End of the summer session (8/12)
- Cumulative, as the course curriculum is also cumulative in design
- 3 hour paper exam – no devices

We will have an **exam recovery policy**. Details TBA.

Past exams: <https://cs61a.org/resources/>

cs61a.org/articles/about-61a/



Midterm Final Homework Projects Lab Discussion

meta-chart.com

Important Websites

cs61a.org

- Hub for all things 61A – syllabus, lectures, announcements, assignments, schedule, etc.

Ed

- Ask questions to the staff and view important announcements
- Assignment help, logistical questions, etc.

composingprograms.com – Textbook

sections.cs61a.org – Section Scheduler

gradescope.com – Assignment Submission Portal

oh.cs61a.org – Office Hours Queue

Should you take CS 61A?

According to the Syllabus: 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 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.

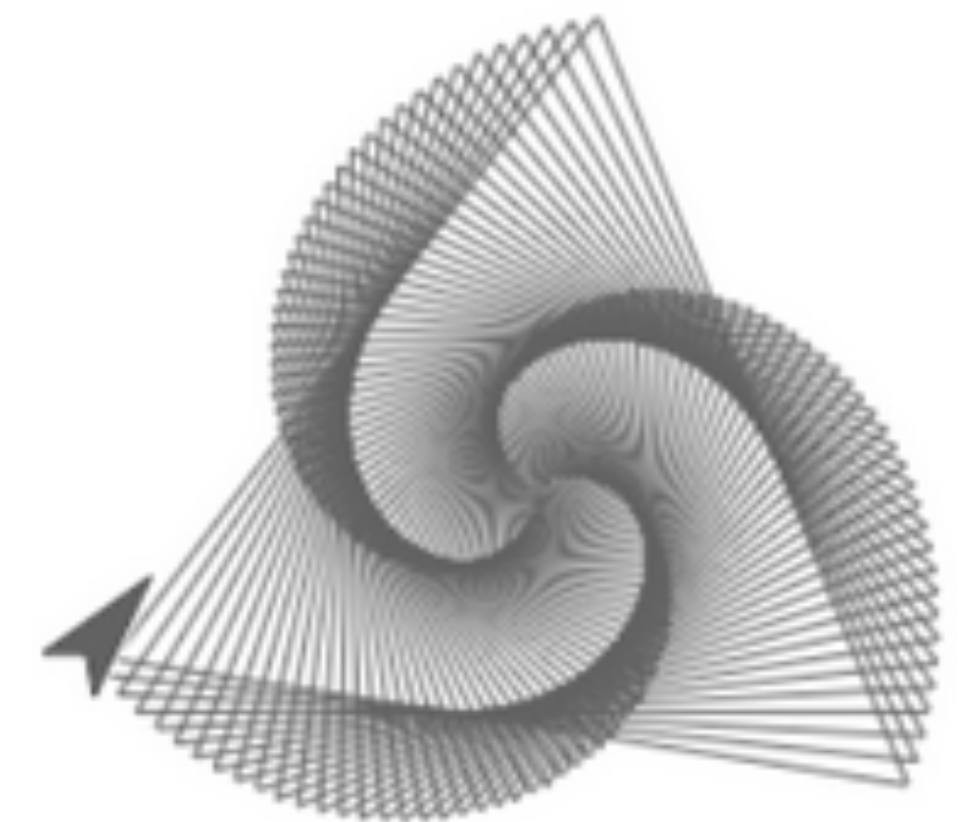
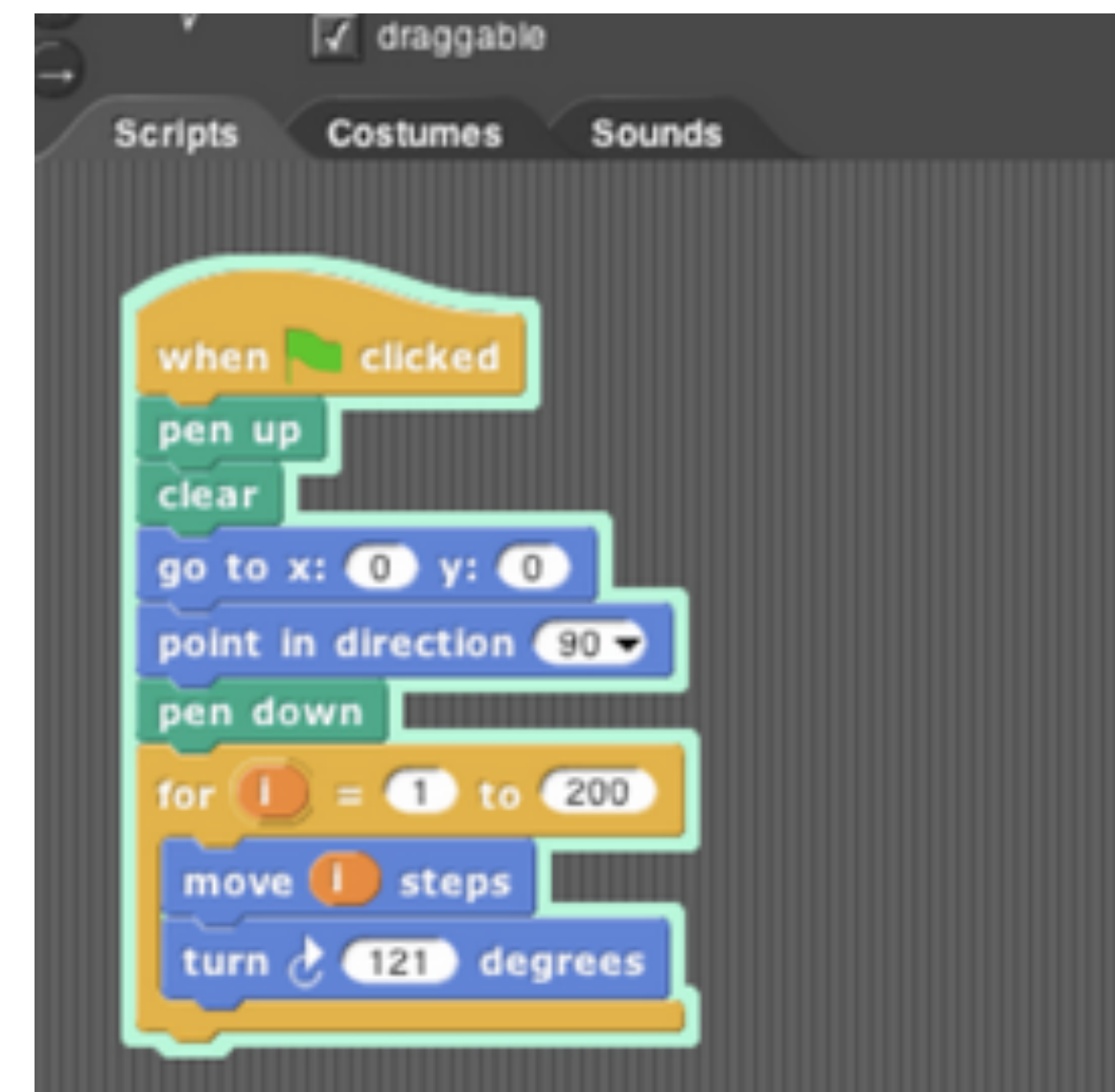
CS 10: The Beauty and Joy of Computing

Designed for students without prior experience

A programming environment created by Berkeley,
now used in courses around the world and online

An introduction to fundamentals (& Python)
that sets students up for success in CS 61A and
Data C88C

More info: <http://cs10.org/>



Data C88C (aka 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)
- Everything you need to know to continue on to CS 61B
- Omits the unit on how programs run other programs & a few advanced Python topics

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

The course is nearly full, but you can join the waitlist

Student Advice from Fall 2024

"If you have no coding experience this is really hard"

"Do not take this class if you don't have programming experience!"

"Make sure you have previous programming experience or understand the contents in AP CSA well before taking this class. It will be very challenging and time consuming if you are new to programming."

"As someone who came into this class with virtually no formal coding experience, remember that it doesn't mean you can not do well in the class. You just need to be prepared to put much more work in than others who do have experience."

"Need prior coding experience to succussed in this class."

"If you have no experience, it's doable, undeniably hard, but doable."

Break: 5 minutes

Course Policies

Learning Community

Community



Learning Community Course Staff

Details...

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

Collaboration

Working together is highly encouraged

- Discuss everything with each other; learn from your fellow students!
- 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 use ChatGPT or other similar tools to write code for you.
- Copying project solutions causes people to fail the course.

Build good habits now

Student Advice from Fall 2024

"LLMs are useful tools that you'll probably use on occasion as a software engineer or computer scientist, but you should be able to solve problems on your own without ChatGPT, as someone who doesn't need to rely on ChatGPT but can use it to augment their workflow will be significantly more productive and competent than someone who is reliant on ChatGPT."

"It'll be very tempting at points to just ask ChatGPT when you're running into a bug. For your sake, I strongly recommend against it... The skills you gain from solving the more simple problems in CS61A will be essential in your ability to solve tougher problems later on, and so reliance on ChatGPT can only take you so far. If you do ever use ChatGPT because you're rushing to meet a deadline, I strongly recommend coming back to the question later and thoroughly understanding it."

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."

go.cs61a.org/laryn-anon: 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.

Extensions and Accommodations

DSP

DSP students: make sure your specialist has sent us your accommodation letter

When we've received your accommodation letter, you'll receive a confirmation email

If you are not enrolled in DSP but believe you are eligible for accommodations, you can visit dsp.berkeley.edu

Extensions

Extension requests must be submitted before the assignment deadline

Request an extension here: go.cs61a.org/extensions

Any extension up to 24 hours will be approved

Any longer extensions require a strong justification and will be reviewed on a case-by-case basis

Any extension made by a student with a DSP accommodation up to 3 days will be approved

Remember, summer sessions moves quickly

Expressions

Types of expressions

An expression describes a computation and evaluates to a value

$$18 + 69$$

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

$$\sin \pi$$

$$\log_2 1024$$

$$2^{100}$$

$$f(x)$$

$$\sqrt{3493161}$$

$$7 \bmod 2$$

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

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

$$|-1869|$$

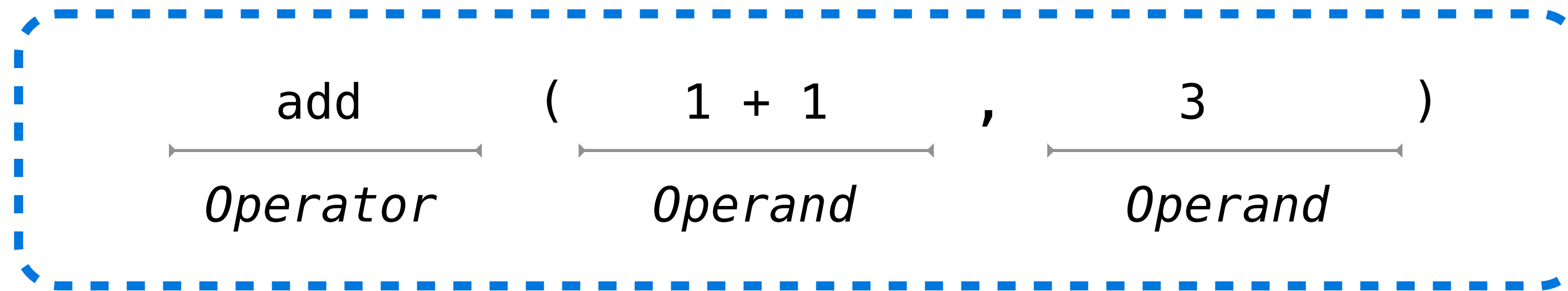
$$\binom{69}{18}$$

Call Expressions in Python

All expressions can use function call notation
(Demo)

Call Expressions

Anatomy of a Call Expression



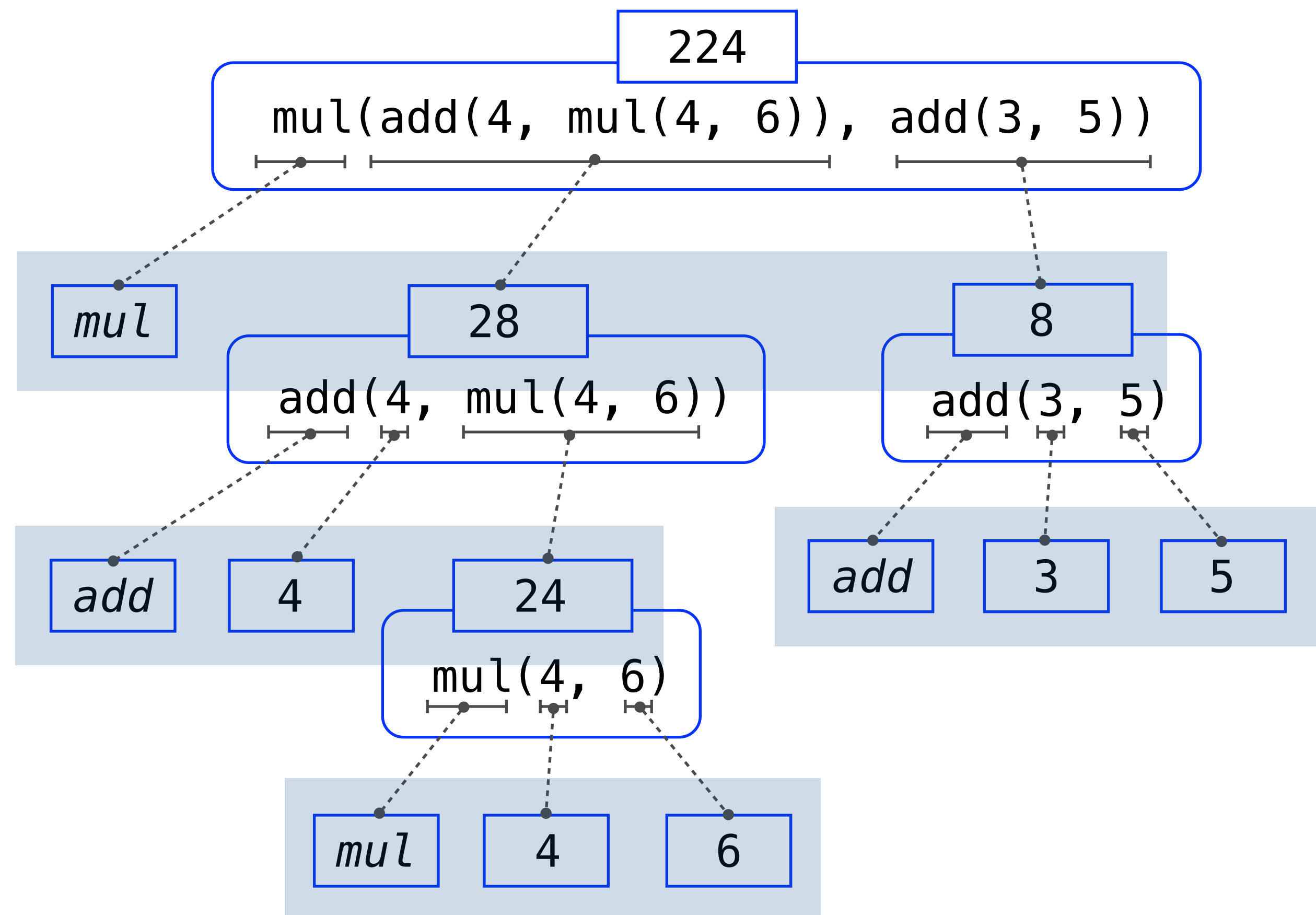
Operators and operands are also expressions

Expressions evaluate to values

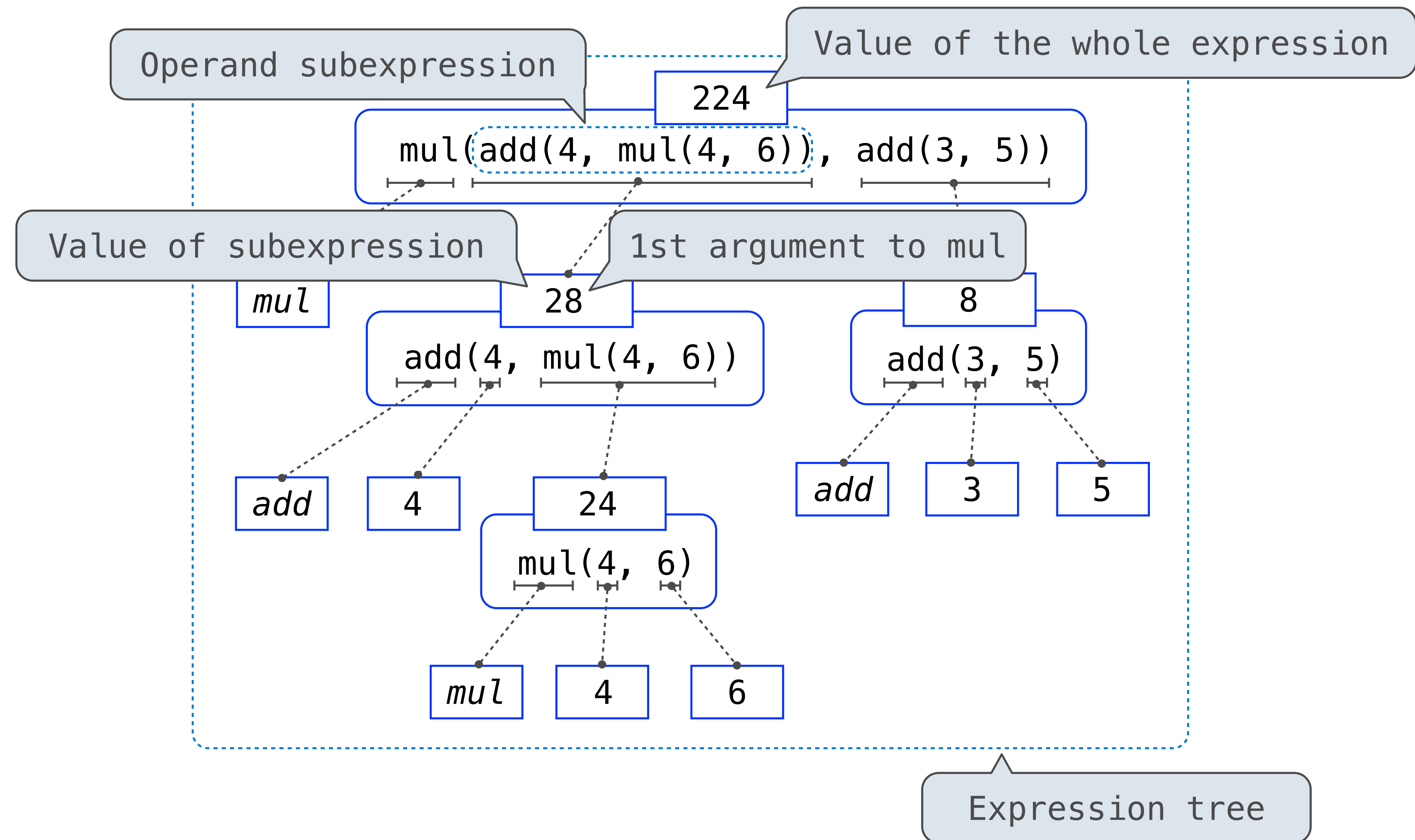
Evaluation procedure for call expressions:

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



Demo