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 fun word cloud of your classmates' interests and chat.

Lecture 1: Introduction
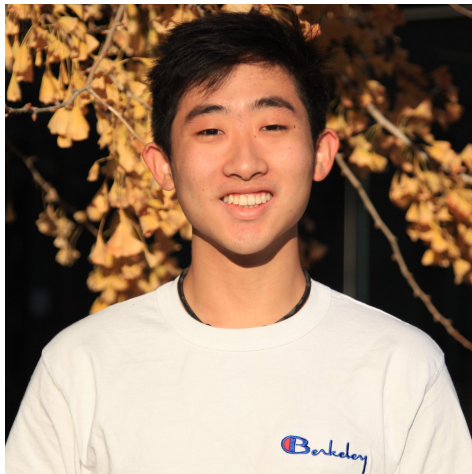
June 17th, 2024

Charlotte, Laryn, Raymond

Welcome to CS 61A!

Humans of CS 61A

Instructors

**Laryn Qi
(he/him/his)**
laryn@eecs.berkeley.edu

**Charlotte Le
(she/her/hers)**
charlottele@berkeley.edu

**Raymond Tan
(he/him/his)**
raymondtan@berkeley.edu

About Me

Undergrad @ Cal from 2020-2024

- B.A. in CS with Honors, Minor in Politics, Philosophy, and Law
- Research: AI (robotics and alignment)
- Industry: Slack/Salesforce, PlayStation/Sony
- CS 61A was my first exposure to programming

Office Hours

● Mon/Wed 1-2 PM in 781 Soda Hall

Tea Times

● 1:1s by appointment

Charlotte Le
✉ charlottele@berkeley.edu

About Me

Undergrad & Masters @ Cal from 2019-2024

- B.A. in CS & Music, M.S. in EECS
- Research: Building AI tools for CS Education
- Industry: Engineering @ AI Startup, Amazon

12th semester teaching CS 61A

My office hours start this week:

- M/Tu/Th 3-4pm in 781 Soda Hall (today!)

- After this week, they will be by-appointment (Calendly)

Laryn
✉ laryn@eecs.berkeley.edu

About Me

Undergrad @ Cal from 2020-2024

-   B.A in Computer Science

Previous industry experience @ Amazon, Capital One

-   Will be returning to Capital One full time this August!

7th semester teaching CS 61A

-   Favorite class at Berkeley!

My office hours start this week:
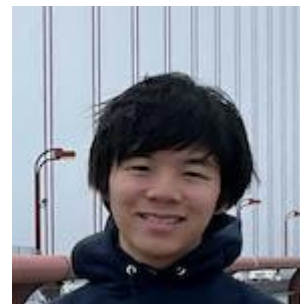
-   Tu/Thu @1-2pm in 781 Soda Hall

Raymond Tan

✉

raymondtan@berkeley.edu

TAs!

Tutors!

You!

**270+ students
31+ majors
UC Berkeley Students, Visiting
Students, High Schoolers**

Computer Science

What is Computer Science?

- What problems can be solved using computation?

- How do we solve those problems using computers?

- What techniques lead to effective solutions?

Systems

Artificial Intelligence ———————————

Graphics

Security

Networking

Programming Languages

Theory

...

Games

Computer Vision

Natural Language Processing

Robotics

Machine Learning

...
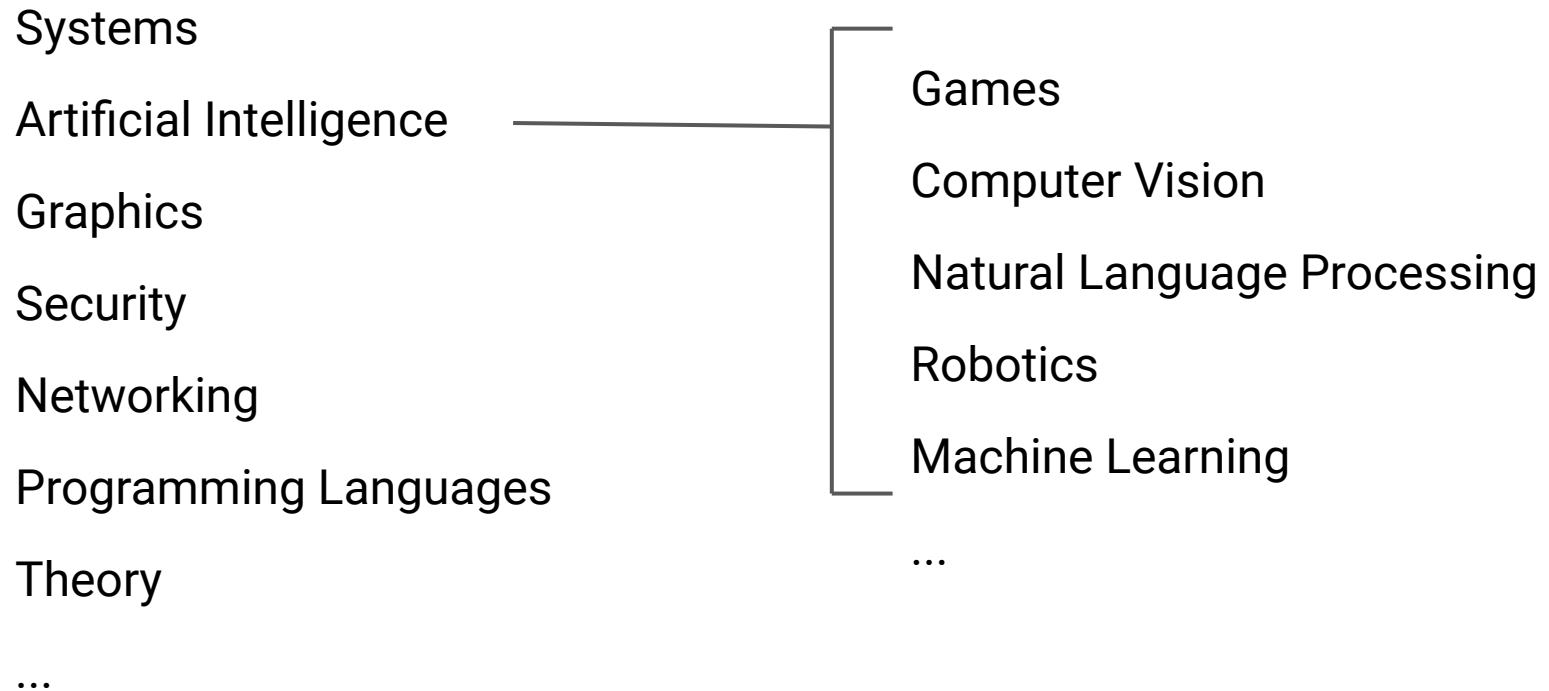
What is CS 61A?

- A course about managing complexity
  - Mastering abstraction
  - Programming paradigms

- An introduction to programming
  - Full understanding of Python fundamentals
  - Combining multiple ideas in large projects
  - How computers interpret programming languages

- Different languages: Scheme & SQL

- A challenging course that will demand a lot of you

Alternatives to CS 61A

CS10: **The Beauty and Joy of Computing**

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

cs10.org
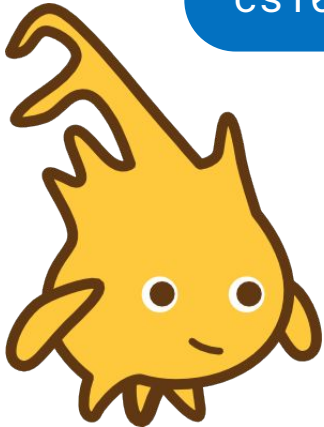
Data 8 **The Foundations of Data Science**

Fundamentals of computing, statistical inference, & machine learning applied to real-world data sets

data8.org

Data c88c **Computational Structures in Data Science**

Introduction to fundamentals, expands computational concepts and techniques of abstraction

c88c.org

Course Format

Course Format

Lecture	MTWTh 11am-12:30pm in Lewis 100
Discussion	the most important part of this course
Lab	the most important part of this course
Office hours	the most important part of this course
Tutoring	the most important part of this course
Textbook	composingprograms.com

- 8 programming **homeworks**
- 4 programming **projects**
- 1 **midterm exam** and 1 **final exam**
- Lots of course support and a great community

Lecture

- Time/Location: 11am-12:30pm in Lewis 100 on M/T/W/Th

- First introduction to the material taught by instructors

- Nothing you need to prepare for lecture, just be ready to listen and take notes!

- You will have opportunities to ask questions about the material

- Will be recorded so you can re-watch later

- Alternative: Pre-recorded lectures by Professor John DeNero

  - These will not have announcements and may not be up to date

Lab (Regular)

- Time/Location: Lots of different times on M/W, Soda Hall

- 90 minutes, twice a week

- Practice working with the material with brief review, good practice for homework and projects

- Assignment completed on your computer

- Lab section run by your TA

- You can work with other students on the problems and get help from your TA

- Can be done independently, but attending 9 lab sections is required

Discussion (Regular)

- Time/Location: Lots of different times/locations on T/Th

- 90 minutes, twice a week

- Practice working with the concepts more in the direction of what might seem on an exam

- Taught by your TA with more conceptual review & mini-lectures

- Nothing to turn in, just practice with the material

- Cannot be done independently, must come to section to get attendance credit

- Attending 10 discussions is required for full credit

- Sign up for Lab & Discussion on sections.cs61a.org – **open today starting at 2pm**!

Mega Section

- Alternative, self-guided approach **using the same assignments**

- **Asynchronous -** no weekly meetings with TA and other students

- Lab:

  - "Getting started" hint videos will be posted along with each lab assignment

  - Must complete and submit all required problems to get credit

- Discussion:

  - Full walkthrough videos will be posted along with each discussion on cs61a.org

  - Attendance credit automatically given

- Need to contact a staff member regarding Mega?

  - Aurelia Wang (**aureliawang@berkeley.edu**)

Should I choose regular or mega section?

- Students that like learning in a more traditional classroom environment should choose regular, while students that feel more comfortable learning at their own pace should choose mega

- Students with less/no prior experience often choose regular

- You can request a change to regular section in the first few weeks (**Deadline is end of Week 2, 6/28)** if you feel that Mega is not the right fit for you

  - Keep in mind that space in regular sections is limited

Office Hours

- Time/Location: Lots of different times/locations throughout the week

- This is a chance to get 1-on-1 help with the assignments during the week

- Come to any of our in-person office hour slots listed on cs61a.org/office-hours, and join the queue on oh.cs61a.org

- TAs/Tutors will spend about 10-15 minutes helping you out with your problem from lab, homework, or project

Small Group Tutoring

- Optional small group tutoring will be provided to students

- 1 hour, twice a week

- These sections will be taught by a tutor with 4-6 students in a section

- It will be a supplement to lab/discussion, not a replacement

- Great chance to practice the material and problem solving more

- Sections will either be focused on exam prep or content review

- Sign up information will be out later this week; starts next week

Lab Assignments

- Usually more introductory to the material.

- Attending lab section to work on the lab assignment is highly encouraged.

- Each lab is worth 1 point and you can receive a maximum of 10 points, but there will be more than 10 labs

- ~2 days to complete each lab.

Homework

- Often more difficult problems (conceptually or implementation-wise).

- You will get 1 homework drop

- Roughly 1 homework a week

Projects

- Contains many problems that work together to build a project (such as a game, or later, an interpreter).

- More difficult and longer than homeworks, labs

- ~1.5 weeks to complete each project with some checkpoints before the deadline

- You're allowed 1 partner for each project!

Optional Assignments

- If you have time and want an excuse to work on some fun project prompts, we will have some optional assignments

- Coding recursive art

Weekly Schedule

	Monday	Tuesday	Wednesday	Thursday
Morning		Complete lab		Complete lab
11:00 AM-12:30PM	Lecture	Lecture	Lecture	Lecture
Afternoon	Lab	Discussion	Lab	Discussion

Exams

Midterm:

- 64 points of your grade

- Roughly halfway through the summer session (7/11)

- ~ 2 hours

Final:

- 92 points of your grade.

- At the end of the summer session (8/8).

- Cumulative (as the course itself is also cumulative in its ideas).

- ~ 3 hours.

We have an **exam recovery** policy!

Exam Recovery

- Our exam recovery policy will allow you to **recover points on your midterm if you perform better on the final**
- Your grade percentage on the midterm will be replaced with a weighted average of the midterm and final grade percentages **if it is higher**, using the following formula:

```
new midterm % = max(midterm %, 0.6 * midterm % + 0.4 * final %)
```

*Example: Jake scores a 50% on the midterm, but studies really hard on the final and scores an 80%. Under the exam recovery policy, his adjusted midterm percentage would be calculated as 0.6 * 0.5 + 0.4 * 0.8 = 0.62 = 62%. Since this is higher than what Jake initially scored on the midterm, his original midterm percentage would be replaced with the adjusted one.*

Important Websites

cs61a.org
- Hub for all things related to CS 61A! Lectures, assignments, etc

Ed
- Ask questions and view announcements here

composingprograms.com
- Read the free, online textbook here

gradescope.com
- Submit all your coding assignments here

oh.cs61a.org
- Join the office hours queue here

Grading

- Grading bins, not curved (everyone could get an A)

- These will not change

A+	≥ 300	A	≥ 285	A-	≥ 270
B+	≥ 255	B	≥ 230	B-	≥ 210
C+	≥ 190	C	≥ 180	C-	≥ 175
D+	≥ 170	D	≥ 165	D-	≥ 160

Discussion: 3.3%

Lab: 6.7%

Midterm: 21.3%

Projects: 33.3%

Final: 30.7%
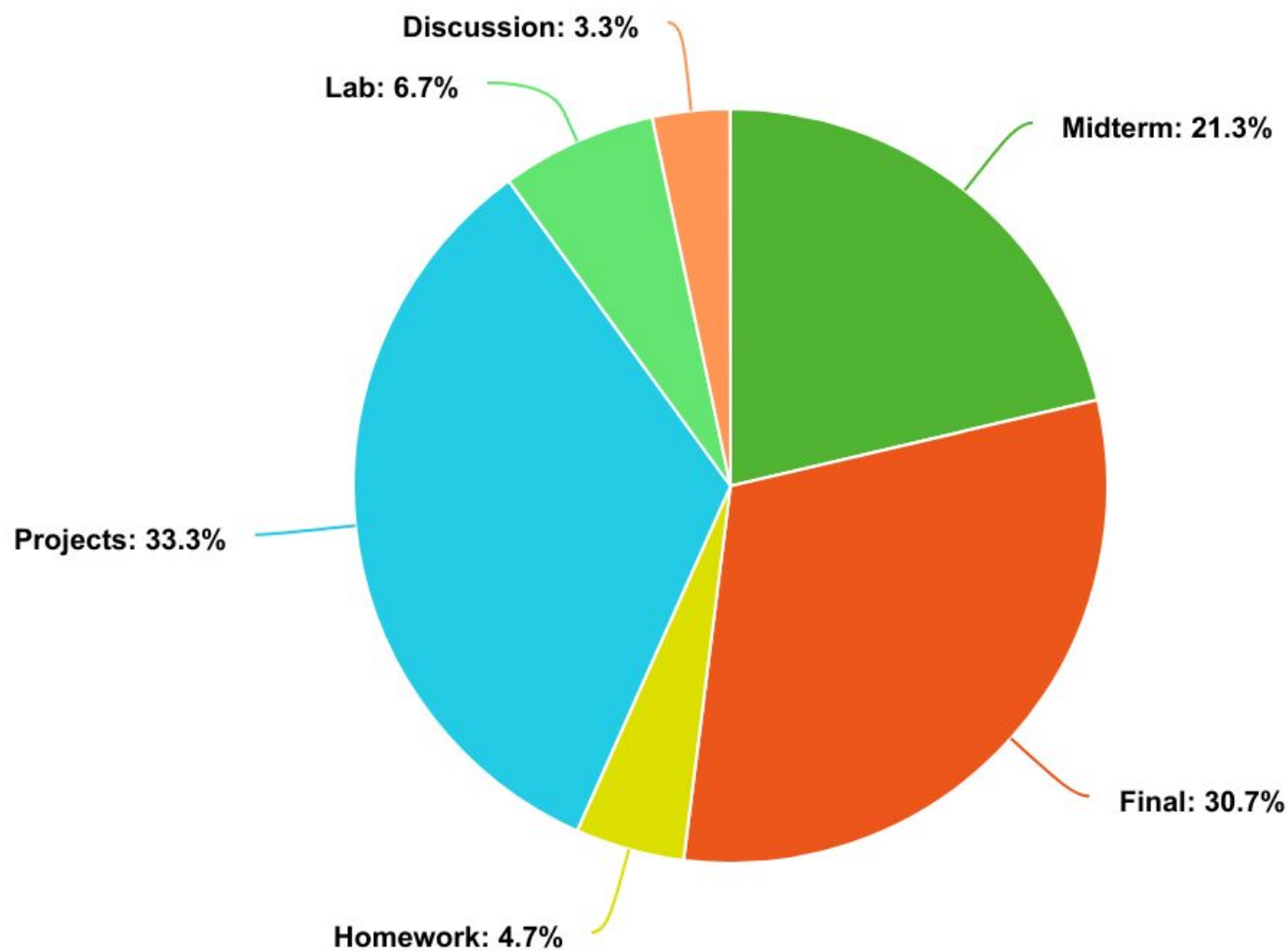
Homework: 4.7%

Midterm   Final   Homework   Projects   Lab   Discussion

meta-chart.com

Academic Misconduct

- We actively encourage collaboration in this class!

- Do:

  - Discuss concepts

  - Share approaches to problems

  - Problem solving, debugging, exam tips

  - Share lab code & share project code <u>with your project partner</u>

- Do not:

  - Share (send nor receive) code with students who are not your project partner or the internet

  - Collaborate on an exam → **F**

- Rather than copying someone else's work, ask for help in Office Hours or Ed!

Quick Break

Extensions and Accommodations

DSP Accommodations

- 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

- We will be flexible and reasonable with extensions.

- Extension requests need to be submitted before the deadline.

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

  - Linked on the "Contact" page of the course website

- Any extension for 24 hours past an assignment deadline will be granted automatically.

- However, remember that this class over the summer moves very quickly!

Finding Resources

- CS 61A Resources: cs61a.org/resources

    - Extra review materials

- Campus Resources: https://cs61a.org/articles/campus-res/,

    - Adjunct tutoring to basic needs services

Class Climate

In this section we will be briefly discussing sexual harassment and discrimination—if you would not feel comfortable being present for this part of lecture you may step out at any point in time

Sexual Harassment

- Often, people don't realize they are sexually harassing someone. The behavior still has the same impact and **impact is what matters**

- What is it?

  - Unwanted sexual or romantic advances:

    - Repeatedly asking someone out

    - Unwanted physical contact

    - Unwanted comments on someone's physical appearance

- See a full list in the [UCOP policy on SVSH](#)

Professional Boundaries

- While the summer session is happening, you should maintain professional boundaries with course staff (Tutors/TAs/AIs)

- Don't message them on social media or dating apps, don't give them physical compliments, don't flirt with them

- Just let them do their job

Reporting SVSH

- All faculty and staff members are Mandated Reporters. If we ever receive an incident report, we will need to make a report to OPHD—we cannot act as confidential resources

- Two goals to reporting

  - Getting you the support you need

  - Increasing our awareness of incidents so that we can try to make improvements

Where to Report

- [Office for the Prevention of Harassment and Discrimination](): Includes an online reporting form
- [Path to Care](): Includes 24/7 care line and advocate appointments
- [CS 61A Anonymous Feedback Form](): If you want to stay anonymous but make the instructors aware of something that happened in the course
- [EECS Anonymous Climate Form](): This will make the EECS department aware of any issues. You can also contact Susanne Kauer ([skauer@berkeley.edu]()) directly.

Racism & Sexism

- In previous semesters of CS 61A, students reported that other students made racist + sexist comments, such as by suggesting that they did not belong in CS

- We live in a society with a long history of racism and sexism and need to actively combat that in both our actions and language

Community Values

- 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 Mission:

  - "Diversity, equity, and inclusion are core values in the Department of Electrical Engineering and Computer Sciences. Our excellence can only be full realized by faculty, students, and staff who share our commitment to these values. EECS's mission is to serve the communities to which we belong, at local, national, and international levels, with a deep awareness of our ethical responsibilities to our profession and to society"

- We need to bring more people into CS, so that tech can create a better future for all

Resources

- Our staff has put together a set of campus resources that includes links for reporting and many other helpful links

Behavior We Want to See

- **Helping each other** understand concepts in the class, whether in section, Ed, or study groups, without expectation of anything in return

- **Saying "congratulations" to classmates** when they finish an assignment, receive a job offer, get into a club, etc.

- **Being a great partner while pair programming.** If driving, listen to what your partner suggests and consider their suggestions. If navigating, brainstorm ideas for how to tackle the problem, re-read the project description, check Ed, etc.

- **Recognize that we're all valuable members of the CS community and here to learn CS together!**

Expressions

$$20 + 24$$

$$2^{100}$$

$$\sin \pi$$

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

$$f(x)$$

$$\frac{20}{24}$$

$$\log x$$

$$\binom{n}{x}$$

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

$$\sqrt{2024}$$

$$-2024$$

An **expression** describes a computation and evaluates to a value.

Expressions in Python

- The Python Interpreter evaluates expressions and displays their value

```
>>> 1 + 2      # add 1 to 2, output the result

3

>>> 3 ** 2     # square 3, output the result

9
```

$$20 + 24$$

$$2^{100}$$

$$\sin \pi$$

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

$$f(x)$$

$$\log x$$

$$\sqrt{2024}$$

$$\binom{n}{x}$$

$$\frac{20}{24}$$

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

$$-2024$$

An **expression** describes a computation and evaluates to a value.

What is a computer program?

- We will use **programs** to do manipulation of **values**

- **Expressions** in programs evaluate to values

- These expressions can be represented by **function calls** in Python

Call Expressions

Evaluation procedure for **call expressions**

1. Evaluate the **operator**

2. Evaluate the **operands** from left to right

3. **Apply** the operator (a **function**) to the evaluated operands (**arguments**)

add (2 , 3)

Operator Operand Operand Apply!

Operators and operands are also expressions

So they also *evaluate to values*

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

Operator   Operand   Operand

Nested Call Expressions

- Humans evaluate inside-out

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

add(add(6,    24    ), mul(3, 5))

add(add(6,    24    ), mul(3, 5))

add(          30      , mul(3, 5))

add(          30      , mul(3, 5))

add(          30      ,    15     )

add(          30      ,    15     )

              45
```
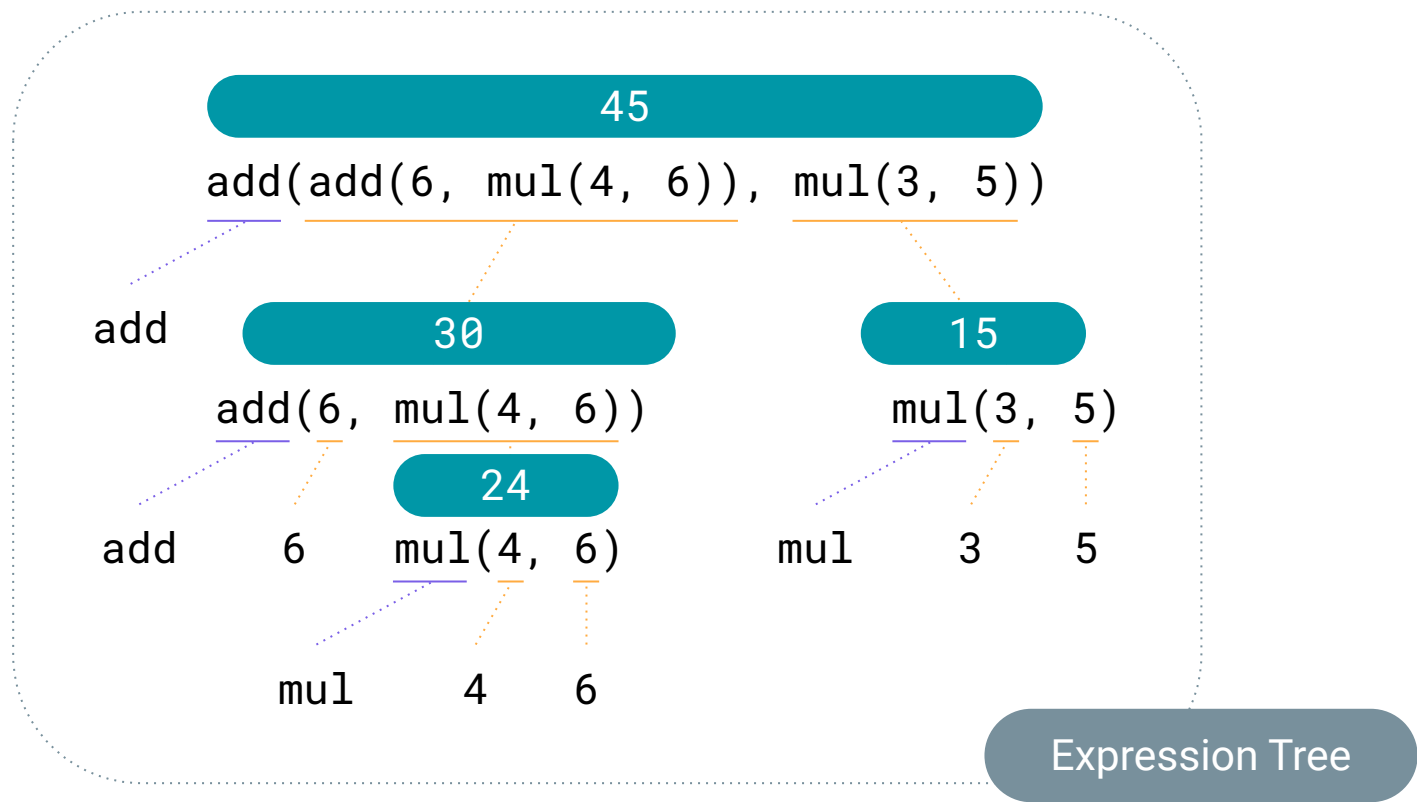
- We can jump ahead or skip around, but Python can't do that!

- How does the computer know which call to evaluate first?

Nested Call Expressions

1 Evaluate operator   2 Evaluate operands   3 Apply!

45
add(add(6, mul(4, 6)), mul(3, 5))

add   30   15
          add(6, mul(4, 6))   mul(3, 5)
                    24
add   6   mul(4, 6)   mul   3   5

          mul   4   6

Expression Tree

Values, Expressions, Objects, and Data

What's happens next?

- Today:

    - **Sign up for section today at 2pm on sections.cs61a.org**

    - Lab 00 is released *now* (required, due Mon 6/24)

- Tomorrow: Lecture 2, Discussion 00, Technical OH begins

- **No lecture on Wednesday** in honor of the campus holiday, Juneteenth

- Thursday: Lecture 3, Discussion 01

- No classes on Friday

- Ask any questions you have after lecture today or on Ed!

Let's have a great semester!