# Welcome to CS 61A

# Announcements

# About the Course

## The 61A Community

39 teaching assistants (TAs), known at Berkeley as GSIs or UGSIs:

- Teach lab & discussion sections
- Hold drop-in office hours
- Lots of other stuff: develop assignments, grade exams, etc.

Tutors:

- Hold drop-in office hours
- Teach 5-person mentoring sections
- Lots of other stuff: homework parties, mastery sections, etc.

Academic interns help answer individual questions during lab

1,450 fellow students make CS 61A unique

## Parts of the Course

**Lecture:** Videos posted to cs61a.org before each live lecture

**Lab section:** The most important part of this course (*next week*)

**Discussion section:** The most important part of this course (*this week*)

**Staff office hours:** The most important part of this course (*next week*)

**Online textbook:** http://composingprograms.com (*read it before class*)

Weekly homework assignments, three exams, & four programming projects

Lots of optional special events to help you complete all this work

**Everything is posted to cs61a.org**

# 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, e.g. operating systems

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

Graphics

Security

Networking

Programming Languages

Theory

Scientific Computing

...

## What is This Course About?

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 types of languages: Scheme & SQL

A challenging course that will demand a lot of you

python

λ

# Alternatives to CS 61A

# 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

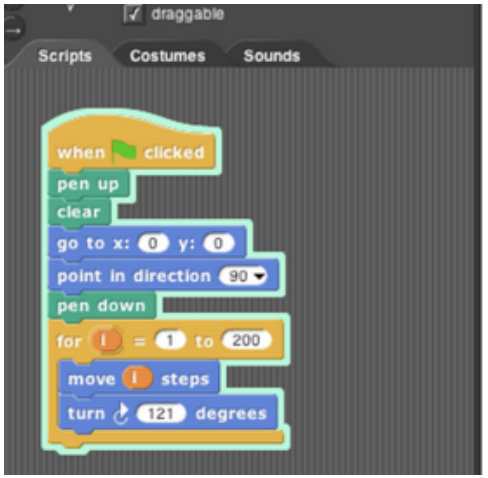
An introduction to fundamentals (& Python) that sets students up for success in CS 61A

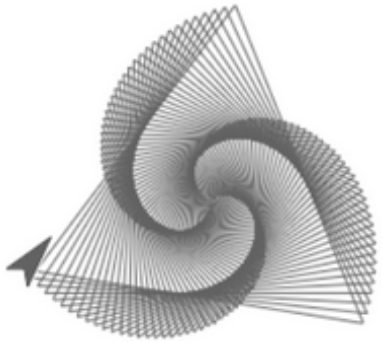
Spring 2020: Dan Garcia

11 open seats (as of Wed 1/22)

Monday & Wednesday 3–4 in 120 Latimer

More info: http://cs10.org/



# CS 88: Computational Structures in Data Science

Alternative to CS 61A with very similar content

- Data 8 overlaps with ~25% of CS 61A
- CS 88 overlaps with 50% of CS 61A or more

Both together cover ~75% of CS 61A, enough to skip CS 61A and go directly to CS 61B

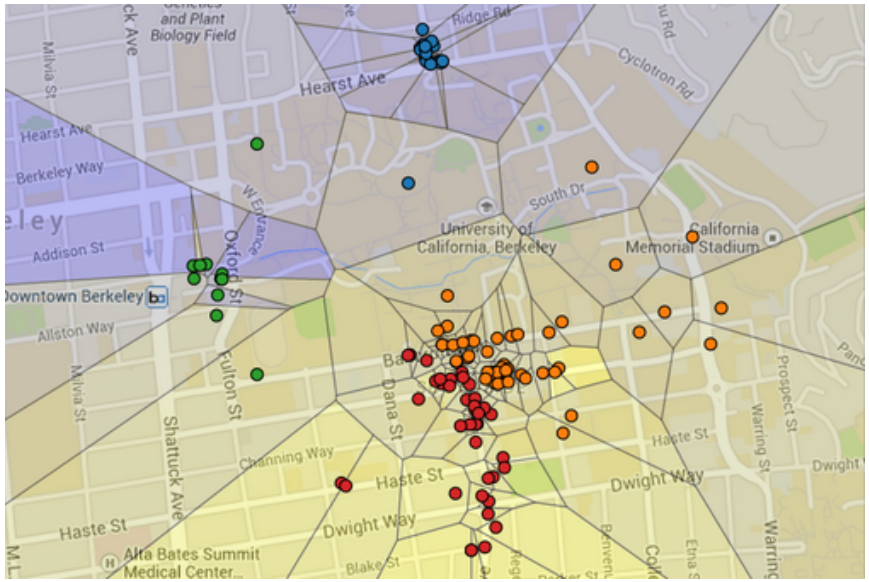
Some students take CS 61A after CS 88 for a very thorough introduction to programming

Spring 2020: 3 units and a revised syllabus

80 open seats (as of Wed 1/22)

Monday & Friday 1–2 in 10 Evans

More info: https://cs88-website.github.io/



# Course Policies

# Learning

# Community

# Course Staff

Details...

http://cs61a.org/articles/about.html

# Collaboration

**Asking questions is highly encouraged**

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

**The limits of collaboration**

- *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, but don't tell them what to type, and don't type for them
- Copying project solutions causes people to fail the course
- We really do catch people who violate the rules, and we're getting even better at it.

**Build good habits now**

# Expressions

## Types of expressions

not particular to computers

An expression describes a computation and evaluates to a value

$18 + 69$

$\frac{6}{23}$

$\log_2 1024$

"sine" takes too long :-

$\sin \pi$

$2^{100}$

$f(x)$ function

$\sqrt{3493161}$

$7 \bmod 2$

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

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

$|-1869|$

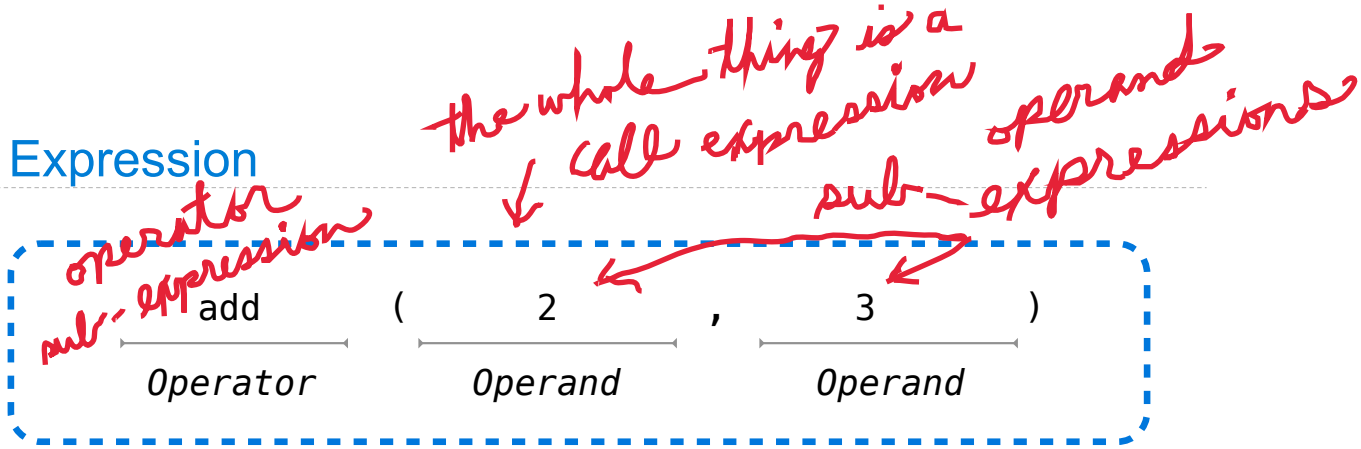
$\binom{69}{18}$

# Call Expressions in Python

All expressions can use function call notation

(Demo)

## Anatomy of a Call Expression



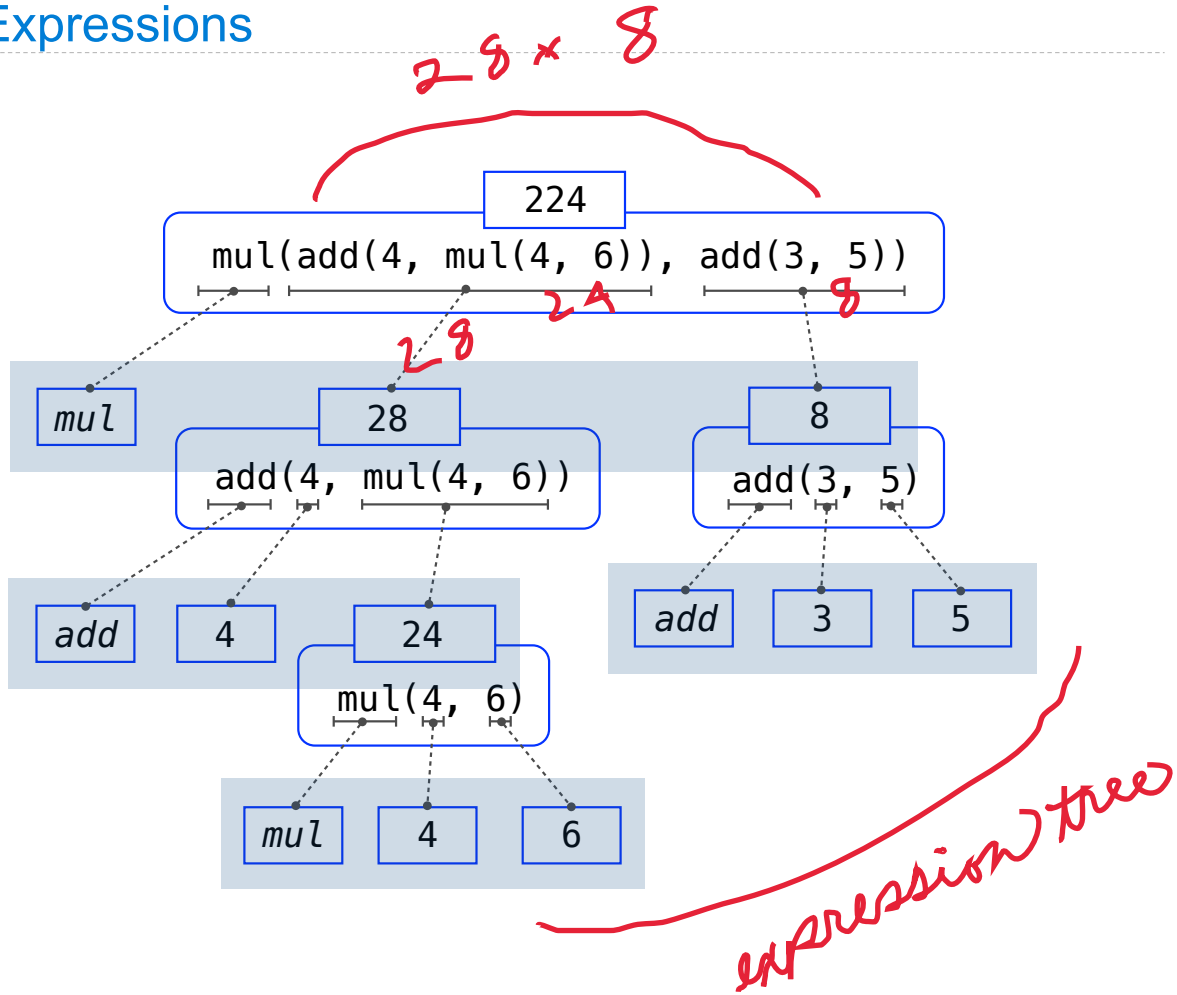

Operators and operands are also expressions

So they evaluate to values

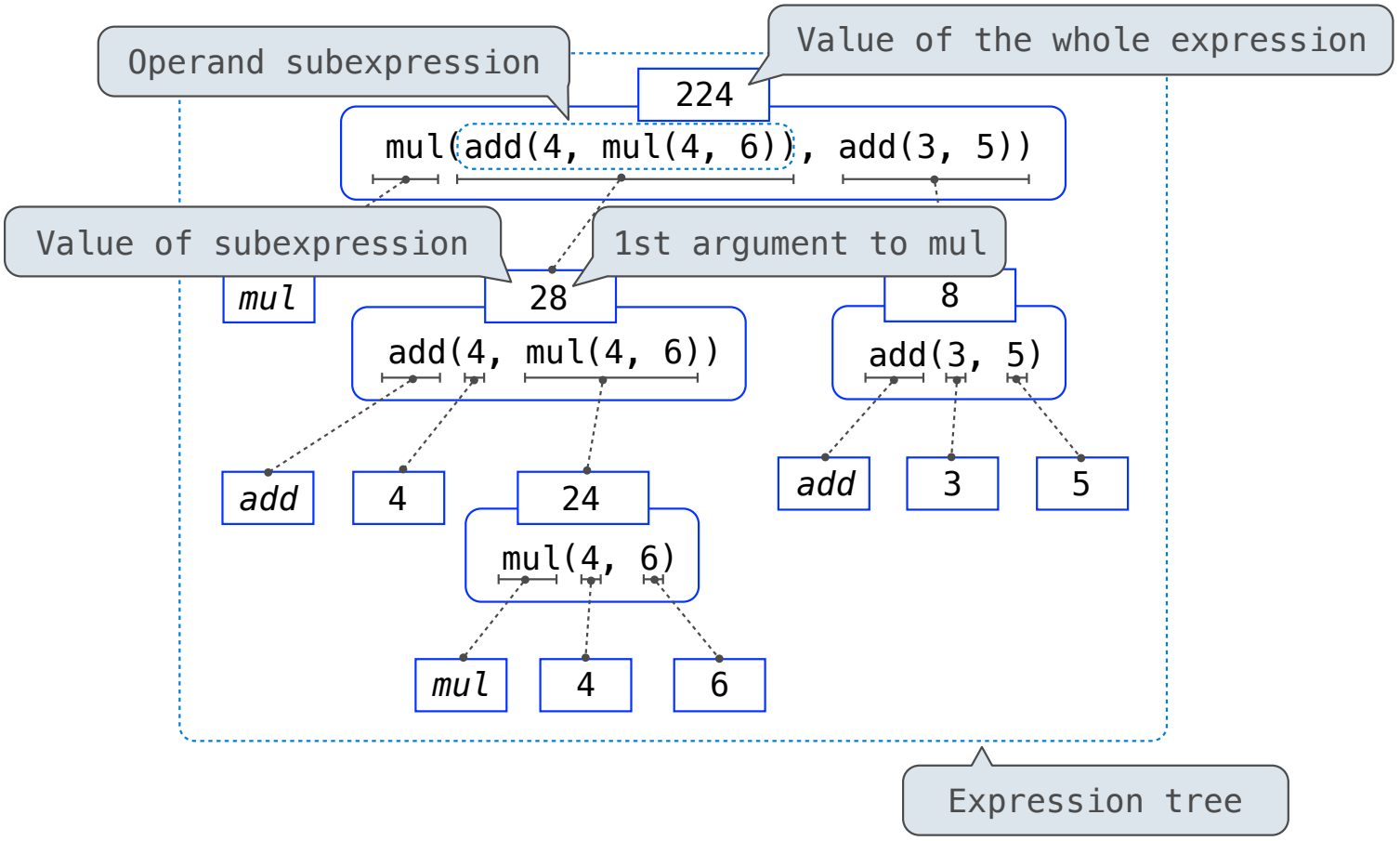
**Evaluation procedure for call expressions:**

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



# Functions, Values, Objects, Interpreters, and Data

(Demo)