

Scaling Introductory Programming Courses: Harnessing Both the Human and the Humane

Mehran Sahami

Computer Science Department
Stanford University



Outline

- The power of humans
 - CS198 section leading program at Stanford
- Scaling
 - Code in Place: scaling section leaders world-wide
- Adding the Humane
 - Embedding EthiCS

CS198 Section Leading Program

- Section Leaders are undergraduate TAs for CS1 and 2 (named CS106A and CS106B, respectively) at Stanford
- Competitive application process
 - 15-30% acceptance rate
 - Enroll in quarter-long workshop on teaching, grading, etc.
- Responsibilities
 - Teach weekly 50-minute section
 - Grade assignments/exams and hold interactive grading
 - Hold “helper hours” in campus computer cluster
 - Weekly staff meetings
 - Total commitment: ~15 hours/week
- Great experience for both students and section leaders
 - Student **rave** about section leaders in evaluations, emails, etc.

Principles Underlying Access to CS

- *Open access to CS courses and major*
- No enrollment limits on introductory classes
- Introductory classes are meant to be a “funnel” not a “filter”
 - Aim to making computing a welcoming option to all students
 - Don’t discriminate between CS/non-CS students
- We don’t create barriers (e.g., minimum GPA) for students to declare CS as a major
 - No limit on the number of CS majors
 - Such barriers disproportionately impact women and underrepresented minorities (alas, another talk, another day)
 - (Generally) uncapped enrollments in later CS courses

Today's first guiding question:

How can we **scale**
access to **human centric**
computer science education,
for **community service**?

Code in Place



Online Section Leading for Scalable Human-Centric Learning



Chris Piech



Ali Malik



Kylie Jue



Mehran Sahami

And many more!



Code in Place:



908 section leaders teach

10,428 students

1/2 of CS106A

As Community Service





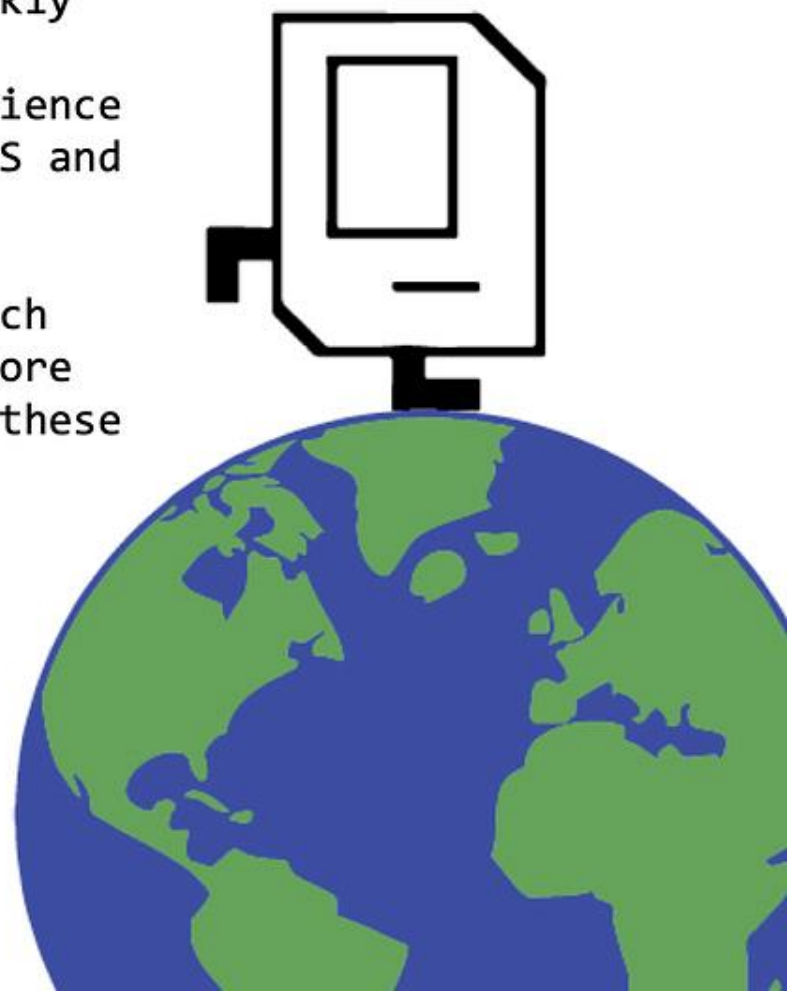
CS106A - Code in Place

def become_section_instructor():

- Lead groups of up to 15 students in weekly problem-solving sessions
- Gain communication and leadership experience
- Deeply solidify your understanding of CS and Python fundamentals
- Join a network of world-class computer science educators from Stanford, the tech industry, the teaching community, and more
- Help give back to the community during these unprecedented times

def become_student():

- Learn about program design, decomposition, encapsulation, abstraction, and testing
- Practice good programming style
- Gain experience with the Python programming language



Gather around and let me tell
you a story

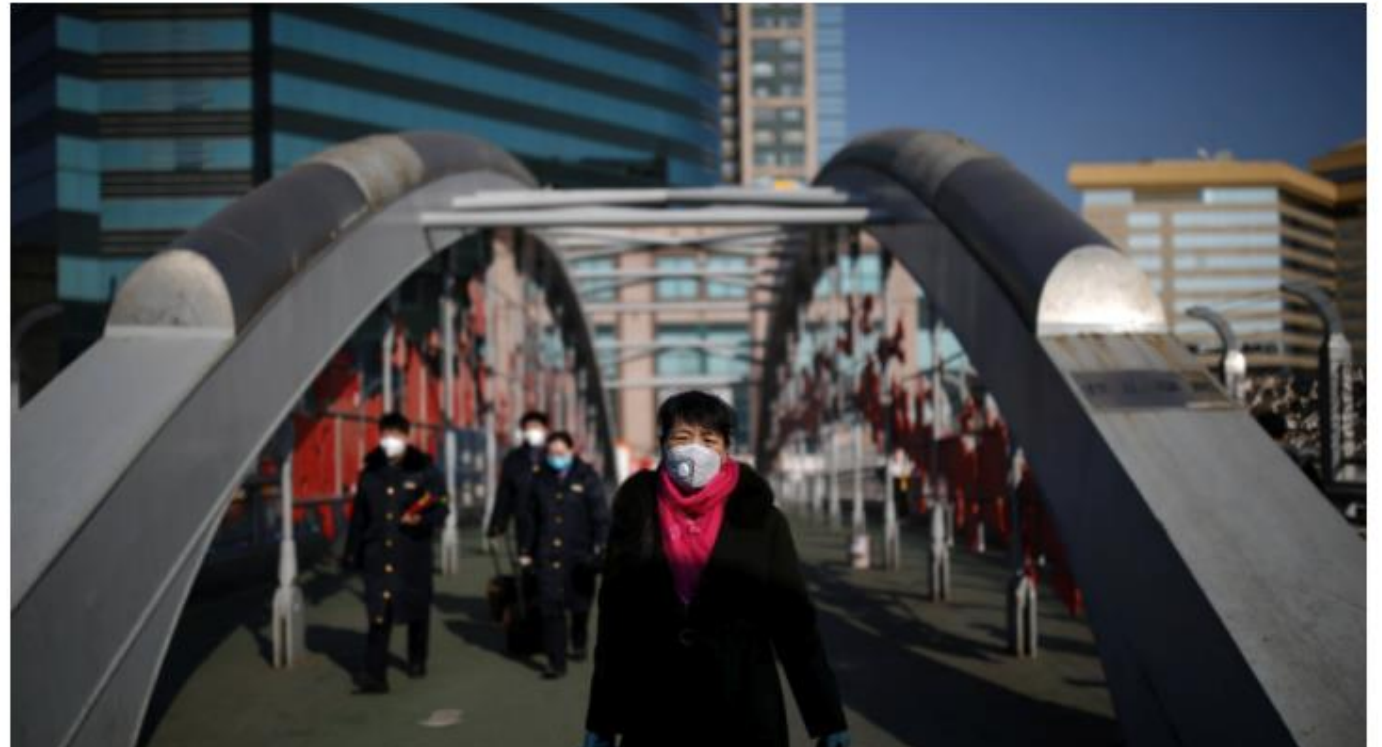


China Reports First Death From New Virus

The coronavirus, which surfaced in the city of Wuhan, has put the region on alert, but there is no evidence among humans.

W.H.O. Declares Global Emergency as Wuhan Coronavirus Spreads

The announcement came as nearly 10,000 cases have been reported worldwide.



Many Counties in California:
Shelter in Place!

What can we do to help?
Code in Place!

Our teaching community responded to
the call in huge numbers.
Over 100 volunteers (alum and current)
to help build.



109



The New York Times

The Year of the MOOC

By LAURA PAPPANO NOV. 2, 2012

coursera



UDACITY

Why wouldn't everyone want to
learn from the world experts?

Why wouldn't everyone want to
learn from the world experts?

Human touch?

Feedback on work?

How do we scale human-centered education?



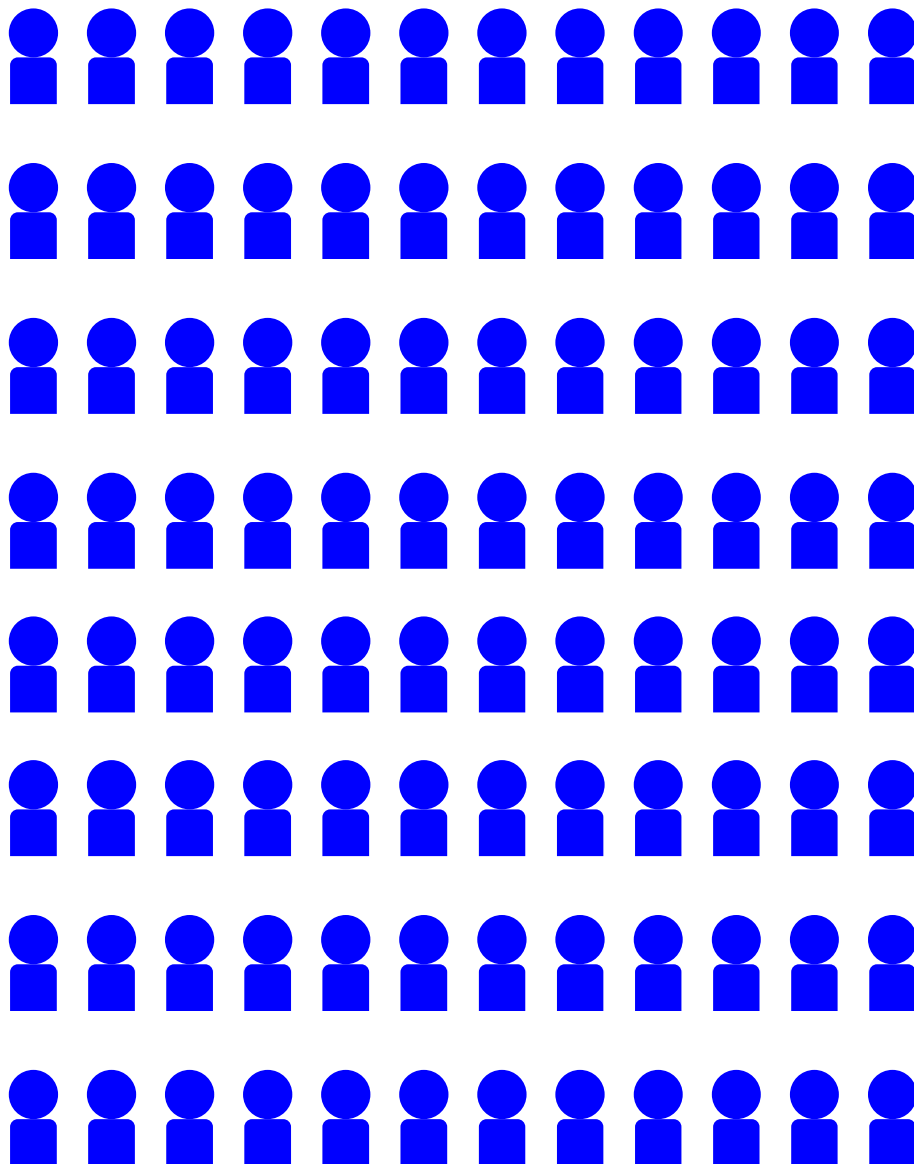
The magnitude of people
who **want to teach** is
roughly proportional to the
magnitude of people who
want to learn.

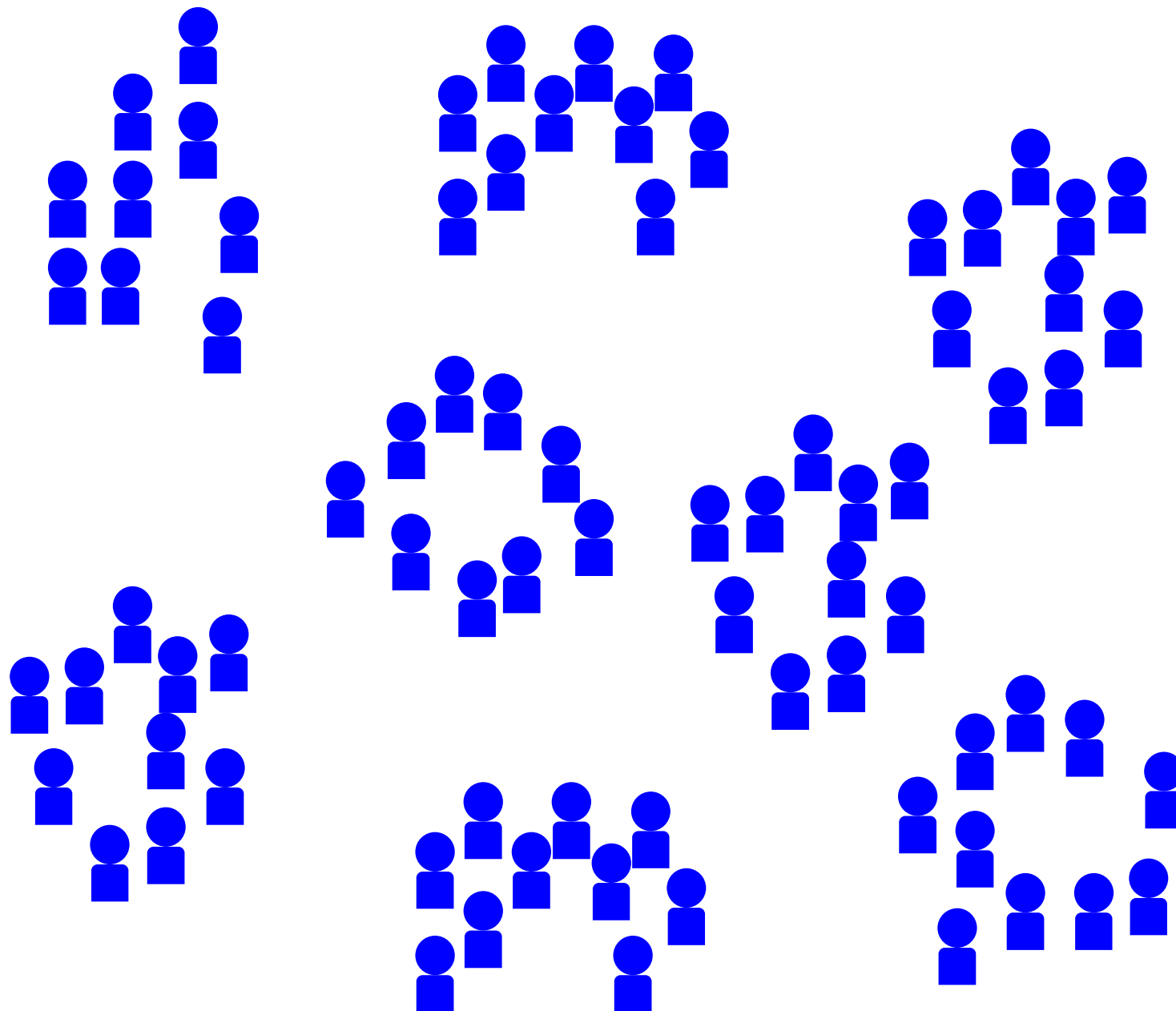
Teaching is **learning**

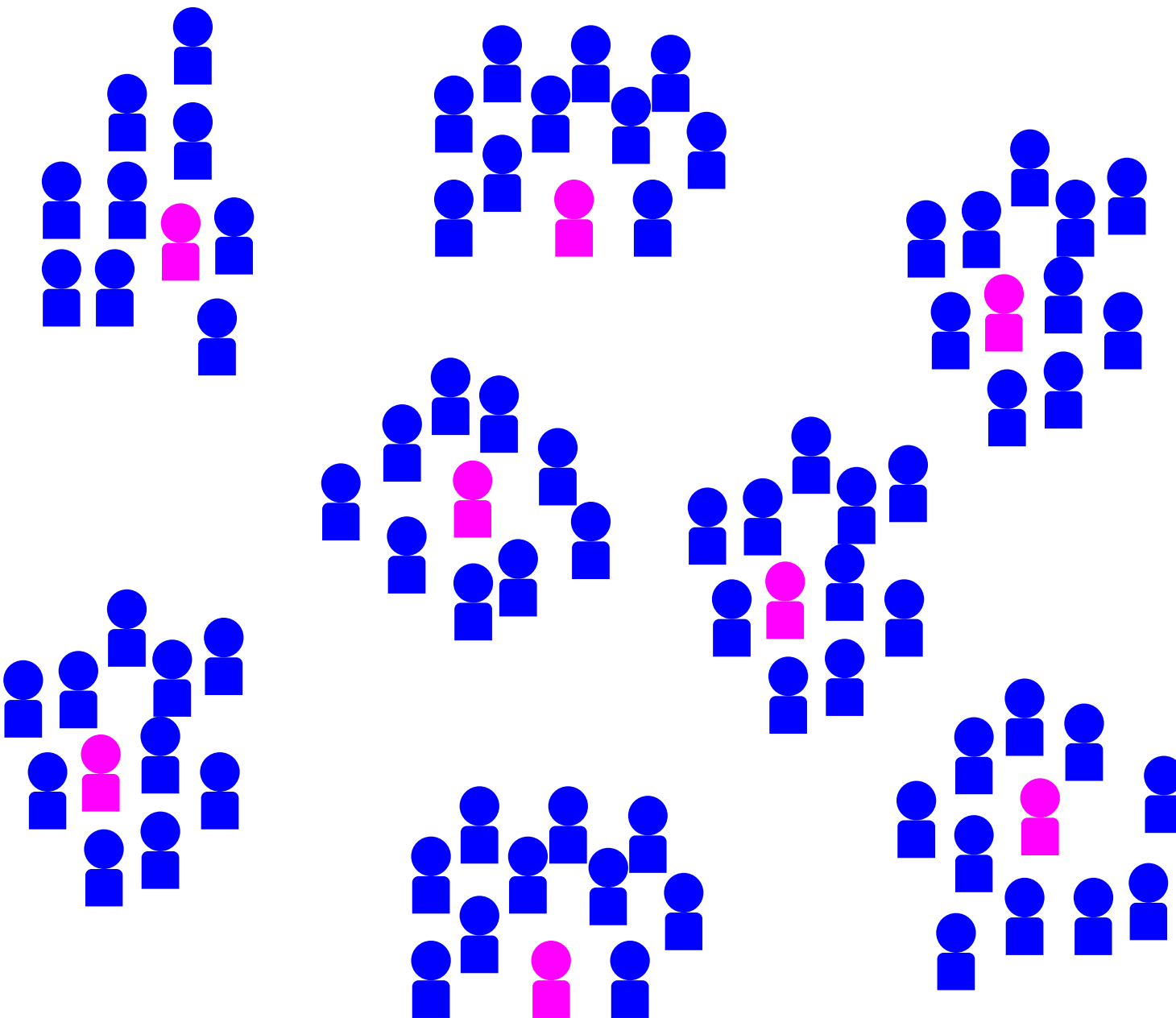
Teaching is **joyful**

The education community has barely scratched
the surface of the **potential** in this claim.









Who are these people?





Section Leaders

including Christelle Scharff!

Section Leaders

Strong grasp of
course material

Often a **first-time** teacher

Teachers from **wider
backgrounds** (industry,
geography etc)

Can be **anyone**! Past students,
industry professionals, etc.

The many benefits of section leaders

Benefits of section leaders

For the student

Section leaders can be closer in lived experience to students.
Can **speak their language** and make believable role models.

Students have high **social encouragement** to complete.

Course is robust. Army of section leaders are bought into making it a success.

Chance to include teachers from **wider backgrounds** (industry, geography etc)

Benefits of section leaders

For the section leader

Section leaders get
education about teaching
(communication + technical)

Teaching is an
intermediate job while
learning. Gives students
purpose.

Develop deeper
understanding of
concepts

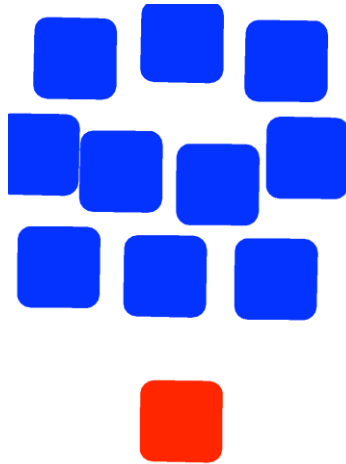
Potential for **networking**
and **job opportunities**

Strong sense of
responsibility

Code in Place SLs

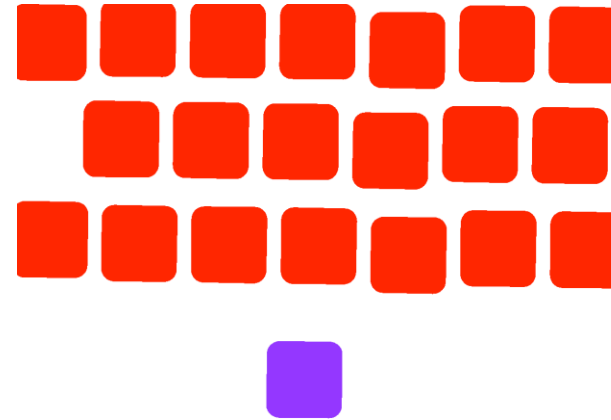
Idea: Hierarchical scale of human teaching

Students meet
once a week in
groups of 10



Led by a volunteer
“Section Leader”

Section leaders
are trained in
cohorts of 20



Led by a “Teaching
Leader”



Key to scaling
human teaching



SL Teaching Responsibilities

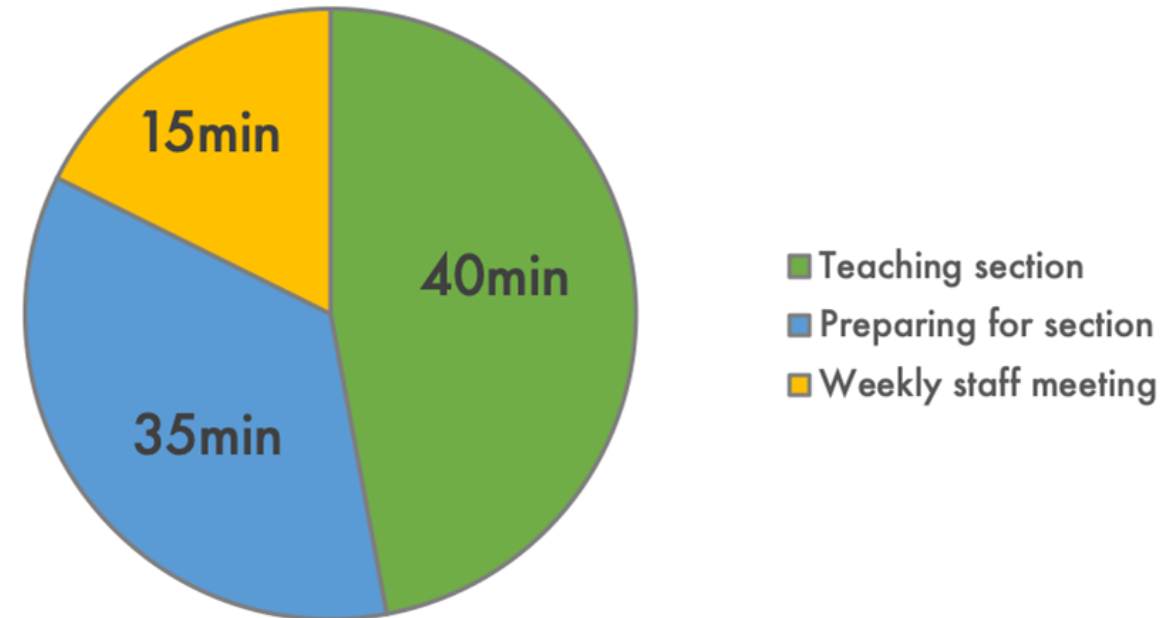
→ Training

Attend 2.5 hours of live teacher training and complete the corresponding readings/videos.

→ Teaching

1.5 hrs/week for 5 weeks (starting the week of April 13th):

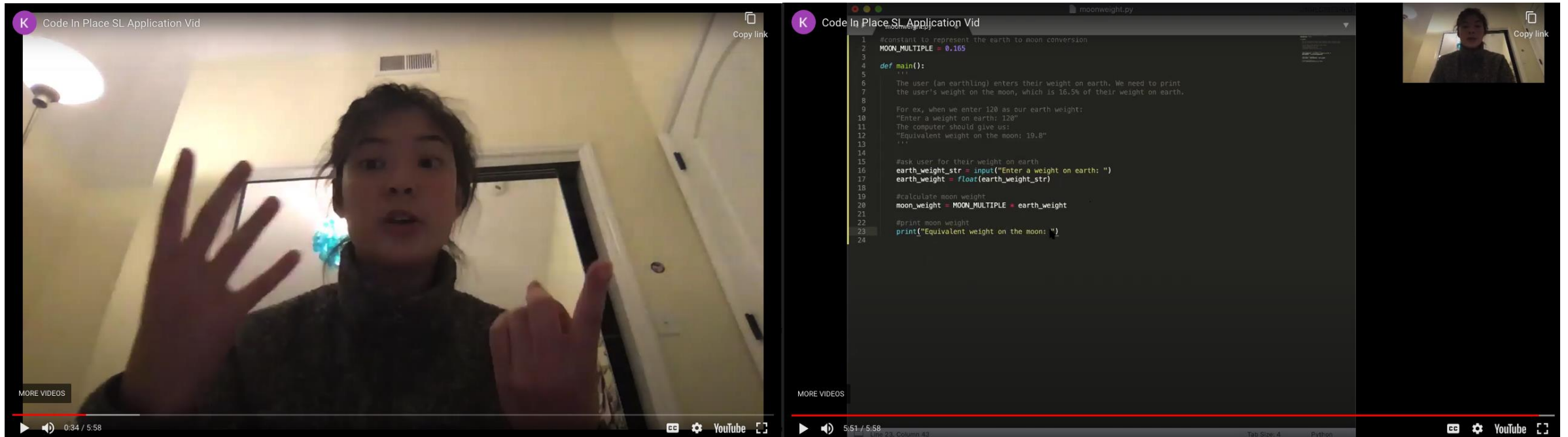
Distribution of your 90min weekly volunteer work



1000+ volunteer section leader
applied in 7 days...

How did they apply?

- Python debugging advice
- Record a 5-minute teaching demo



The image is a composite of two video frames. The left frame shows a person with dark hair, wearing a dark jacket, speaking and gesturing with their hands. The right frame shows a code editor window titled 'moonweight.py' with the following Python code:

```
1 #constant to represent the earth to moon conversion
2 MOON_MULTIPLE = 0.165
3
4 def main():
5     """
6     The user (an earthling) enters their weight on earth. We need to print
7     the user's weight on the moon, which is 16.5% of their weight on earth.
8
9     For ex, when we enter 120 as our earth weight:
10    "Enter a weight on earth: 120"
11    The computer should give us:
12    "Equivalent weight on the moon: 19.8"
13    """
14
15    #ask user for their weight on earth
16    earth_weight_str = input("Enter a weight on earth: ")
17    earth_weight = float(earth_weight_str)
18
19    #calculate moon weight
20    moon_weight = MOON_MULTIPLE * earth_weight
21
22    #print moon weight
23    print("Equivalent weight on the moon: ")
24
```

Both video frames have a 'Code In Place SL Application Vid' title and a 'Copy link' button. The left frame has a 'MORE VIDEOS' button and a progress bar showing 0:34 / 5:58. The right frame has a 'MORE VIDEOS' button and a progress bar showing 5:51 / 5:58.



Why did they volunteer?

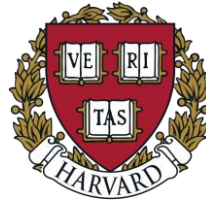
Reason for wanting to volunteer Section Lead	Percent of applicants
Give back through community service	86%
Improve my own teaching ability	72%
Be part of an experiment in online education	66%
I just love teaching programming	66%
Be part of a community of section leaders	57%



Who were they?



KOÇ ÜNİVERSİTESİ



Bug Bounty Hunters

Privacy in Social Media

Columbia Law School

Computer Science

Google Stanford

Lecturer

Math

Physics

Boeing

Microsoft Research India

Law Student Infosys Librarian

Professor Lyft CTO

Chan Zuckerberg Biohub

Technische Universität München

Psychology of Language

Computational and Systems Immunology

Mechanical Engineering

Electrical Engineering

Delhi Technological University

Healthcare Startup

University of Edinburgh

Professional Poker Player

University of Oregon

Robotics Engineers

Master's Student

Software Engineer

High School Teacher

PhD Student

Undergrad Student

Bilkent University, Turkey

And so much more...

How do we **train** section leaders from
such a **wide range of experiences**?

Modeling training after the section experience

- Small groups that enable active learning and discussion-based activities



Modeling training after the section experience

- Small groups that enable active learning and discussion-based activities
- Consistent “teaching leader” as a touchpoint for community and near-peer mentorship



Modeling training after the section experience

- Small groups that enable active learning and discussion-based activities
- Consistent “teaching leader” as a touchpoint for community and near-peer mentorship
- Curriculum designed for a diverse group of new and experienced teachers



Training Sessions

Welcome to Code in Place!	30 min	Before Code in Place starts with course instructors
Preparing for your first section	1 hour	Before their first section with their “teaching leader”
Leading a section for everyone	1 hour	After their first section with their “teaching leader”
Section Leader Learning Week	N/A	Optional workshops taught by staff and fellow section leaders



Training Sessions

Welcome to Code in Place!	30 min	Before Code in Place starts with course instructors
Preparing for your first section	1 hour	Before their first section with their “teaching leader”
Leading a section for everyone	1 hour	After their first section with their “teaching leader”
Section Leader Learning Week	N/A	Optional workshops taught by staff and fellow section leaders



Leading a section for everyone

Reflect on their
own learning



Reflect on their
teaching



Consider student
hypotheticals

Toolbox for Outstanding Section Leaders

Learning programming with effective instruction for ALL

- At Code in Place, we believe **teaching is about supporting student learning**.
- Learning to program is not easy – students are learning a completely new way of thinking, and we are there to support them in the process!
- Remember that once upon a time you were also a novice. Recall the beginning of your programming journey. Share with students concepts that you struggled with and early misconceptions you had. Learning is not a bug-free endeavor!

1) Pro tips for great teaching

Group management:

- Ensure **all students have a chance to have their voice heard** (they contribute and interact with the group), respecting their preferred mode of participation. For some students, this might be verbal, for others written (using the chat), and others might



Student Application



Code in Place App

Due midnight April 8th, 2020 Anywhere on Earth

Hello and welcome Chris Piech,

Signout

The time of COVID-19 has been difficult for many people around the world, in many different ways. As an act of community service, a group of computer science instructors is coming together to offer our teaching services free of charge for people who want to learn introductory coding. [About the course.](#)

To apply, please finish the tasks below by April 8th. We will review all completed applications by April 10th.

For beginners: We don't expect you to have any coding knowledge before taking this course! We expect the exercises to be a little tricky - but you can do it - check out the FAQs if you get stuck!

Not enrolled yet: We want to offer this experience to everyone, but our number of spaces may be limited by how many high-quality volunteer teachers we get. We will review every single application that was completed by April 8th, 2020.

1. Tell us about yourself

Fill out this form

2. Learn Karel basics

Read chapters 1 through 5

3. Do a few exercises:

A. Warmup

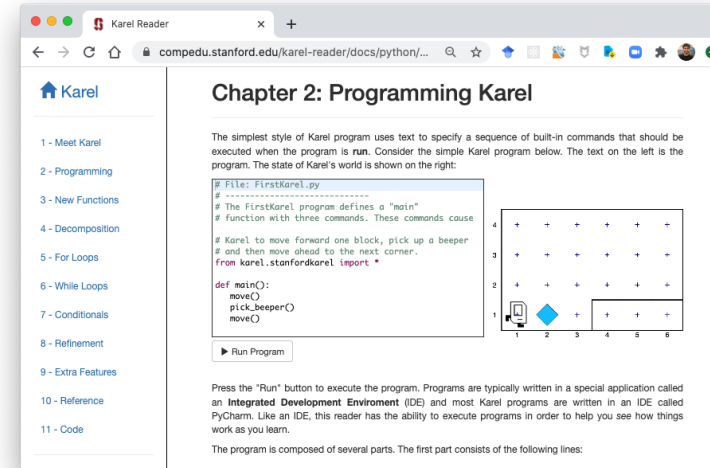
B. Shelter-in-Place

C. Piles

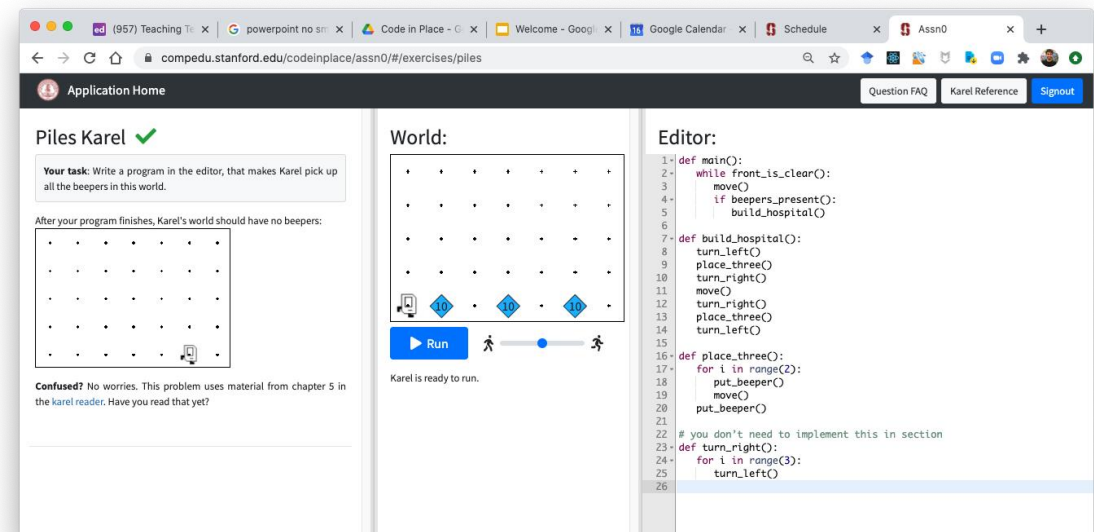
Just for fun! If you want more Karel:

Bonus: Frame Karel

Why are you taking the class? Demographics?
Experience? Time commitment? Background
statement. Honor code!

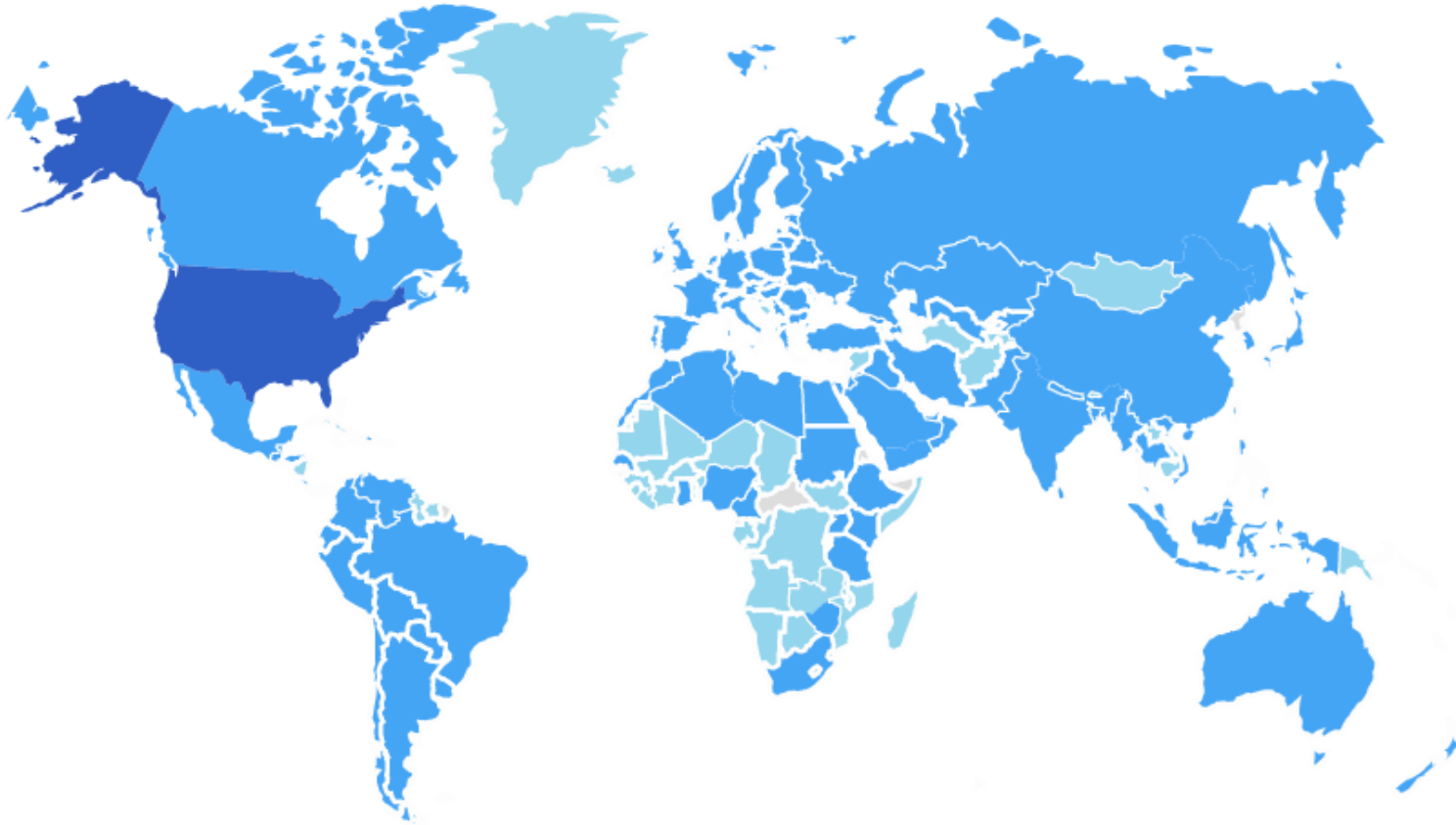


We log every
keystroke and look
at completion time

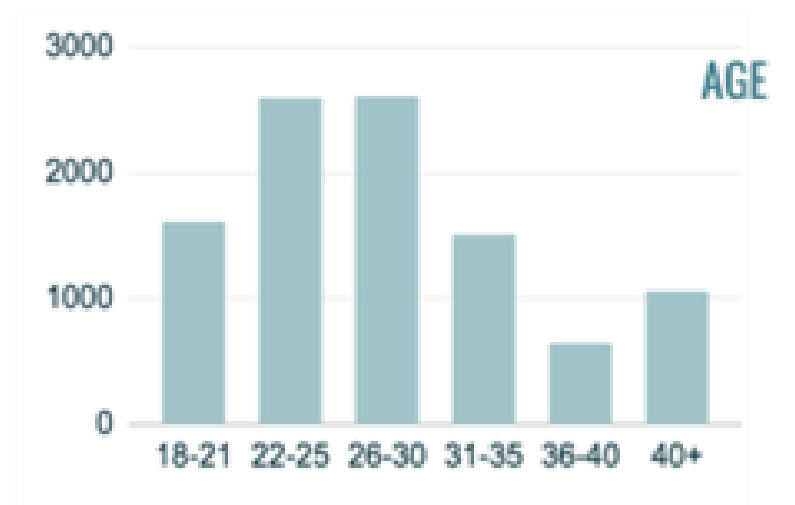
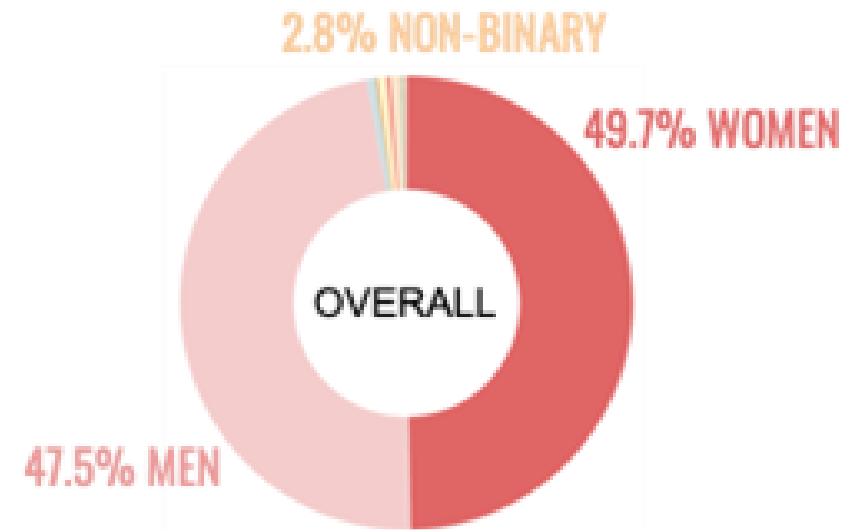


50K started
20K completed
10K admitted

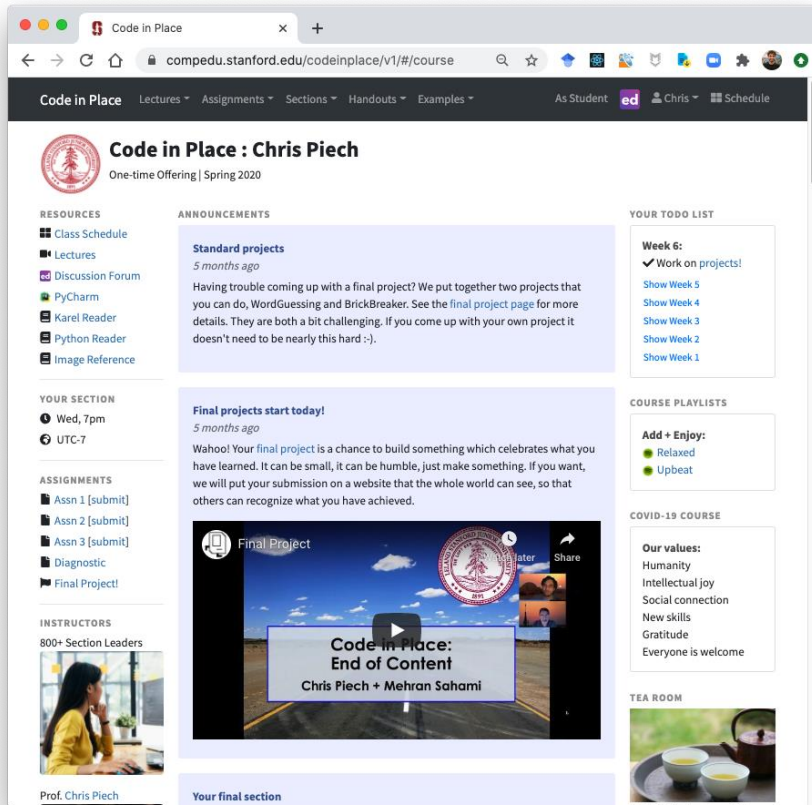
Student Demographics



40% of students had a job or living situation change because of COVID-19



Student Experience



The screenshot shows the Code in Place website for Chris Piech's course. The header includes navigation links like 'Lectures', 'Assignments', 'Sections', 'Handouts', and 'Examples'. The main content area features announcements such as 'Standard projects' and 'Final projects start today!'. A sidebar on the left lists resources like 'Class Schedule', 'Lectures', and 'Assignments'. The bottom section highlights the 'Final Project' and 'Your final section'.

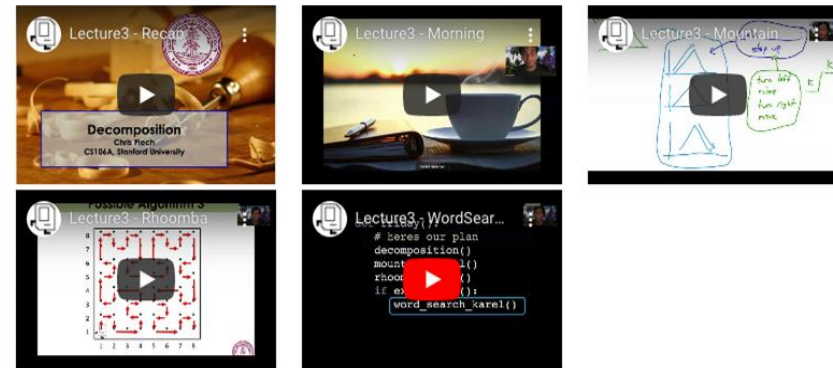
Course Website

Recorded videos

Lecture 3 - Decomposition

APRIL 17TH, 2020

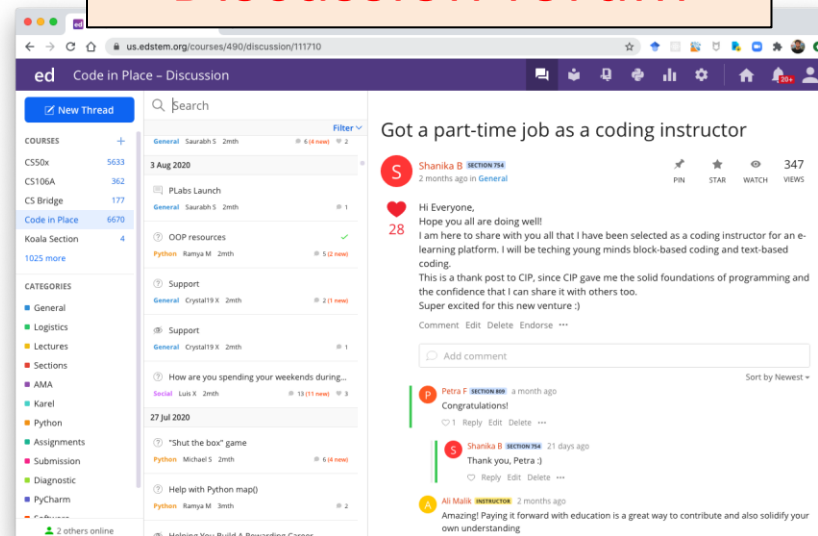
How can you solve large problems in Karel? Have a great weekend and see you on Monday!



Live sections
once a week



Discussion forum



The screenshot shows the Code in Place discussion forum. The forum has a search bar and a list of threads. A prominent thread titled 'Got a part-time job as a coding instructor' by Shanika B. is highlighted. The thread includes a post where the author shares their experience and a comment from Petra F. congratulating them.



IDE for Coding

Course Structure and Content

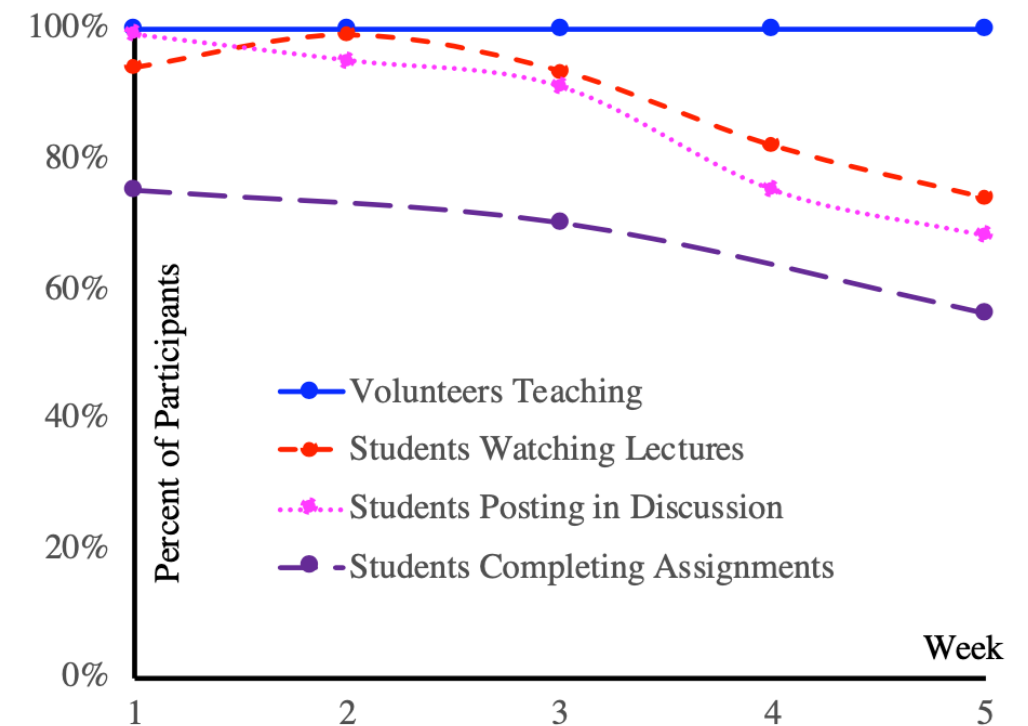
- Structure
 - Half of Stanford's CS106A (CS1)
 - Karel the Robot (1 week)
 - Python (4 weeks)
 - 3 hours of lecture content and 1 hour section each week
- Topics
 - Control flow
 - Variables/expressions
 - Functions and parameters
 - Images and graphics
 - Strings
 - Lists
 - Dictionaries

Wk	Monday	Wednesday	Section	Friday
1	APR 13TH 1. Welcome	APR 15TH 2. Control Flow	APR 15TH, 7PM Karel Section	APR 17TH 3. Refinement
2	APR 20TH 4. Variables in Python	APR 22ND 5. Expressions Assn 1 due	APR 22ND, 7PM Hello Python Section	APR 24TH 6. Flow in Python
3	APR 27TH 7. Functions Revisited	APR 29TH 8. More Parameters	APR 29TH, 7PM Functions Section	MAY 1ST 9. Images Assn 2 due
4	MAY 4TH Diagnostic	MAY 6TH 10. Graphics	MAY 6TH, 7PM Graphics Section	MAY 8TH 11. Animation
5	MAY 11TH 12. Lists	MAY 13TH 13. Text Processing Assn 3 due	MAY 13TH, 7PM Data Section	MAY 15TH 14. Dictionaries
6	MAY 18TH Final Project	MAY 20TH Final Project	NO SECTION	MAY 22ND Final Project

Outcomes

Participation

- 99.7% of section leaders completed class
 - Largest number of section leaders in any class that we are aware of
- 56% of students completed class
 - Submitted all programming assignments
 - Significantly higher than traditional MOOC completion rate (~5%)
- Sincere community of students
 - Positive, uplifting discussion comments
 - Student made cookies with class designs
 - "One of the best experiences of my life."
 - "This course saved my life... You were my only lifeline. I really cannot thank you enough!"



Section Leaders and Students Loved It

Group	Net Promoter (raw)
Students	+90.3 (9.7)
Section Leaders (SL)	+70.1 (9.2)
SL: First Timer	+76.3 (9.4)
SL: in University	+75.4 (9.5)

* Surveyed a random sample of students and section leaders who ***started*** the course

Additional Outcomes

- Students wanted a T-shirt to remember the class
 - Students created designs. Voted on a winner.
 - We turned it into a fundraiser. Raised \$15,000 for charity voted on by students.
 - Many students bought t-shirts for other students in class who could not afford them.
- 6 students reported getting full time coding jobs after the class
- 1 new CS teacher



Takeaways

Unique to Spring 2020?

Perhaps. But we think the answer may be **no**.

Actually Spring 2020 was a hard time:

17% of students had a change in **employment**

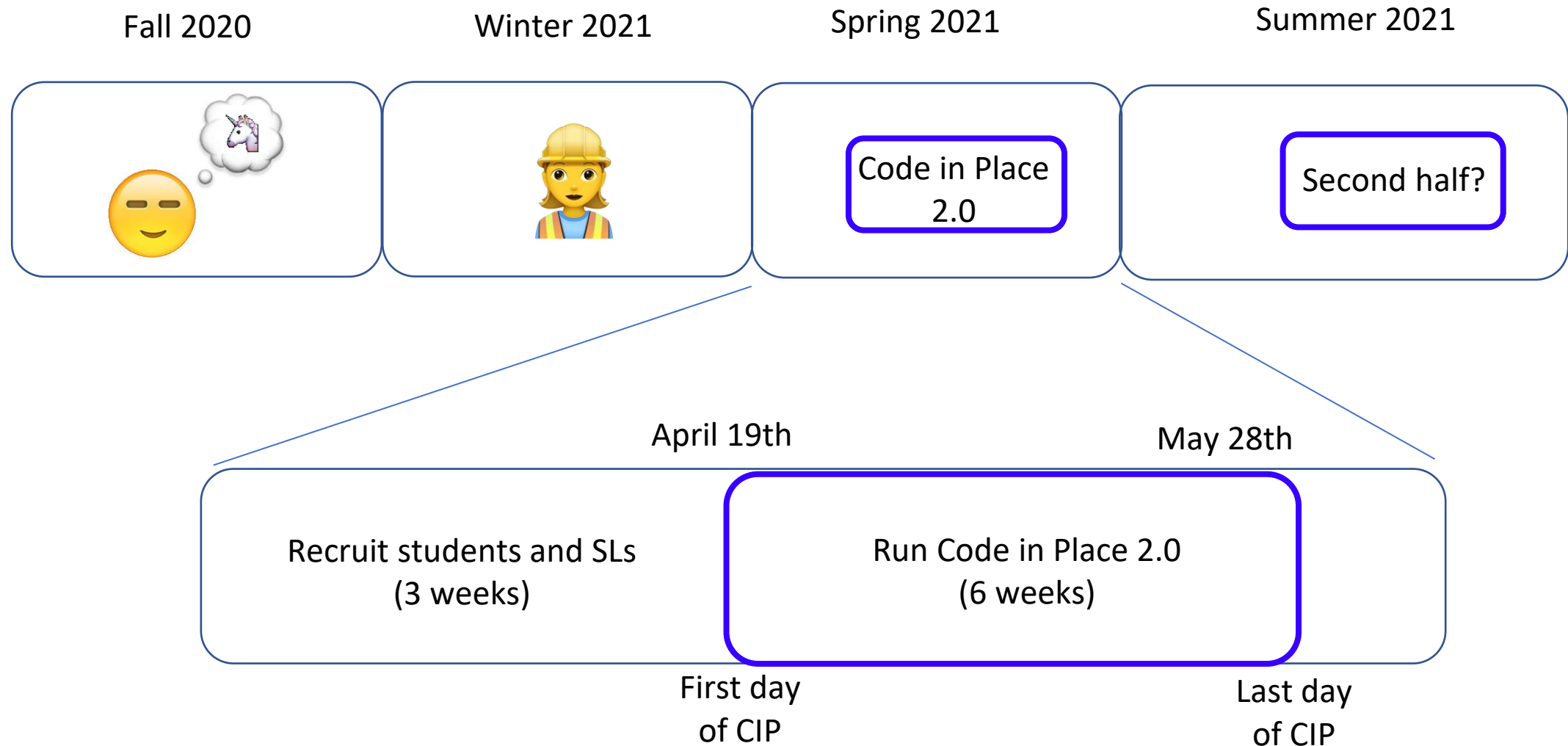
16% of students reported a change in **living situation**

Takeaways

- The scaled-section-leader idea worked in an online class.
- It is a time for community service education projects.
- Sense of responsibility drives completion.
- Especially great for getting folks their *first* teaching experience.

Can we do it again?

Code in Place 2.0 in Spring 2021?



What is the most important thing to
recreate?



Today's second guiding
question:

How can we get **CS students**
to better **appreciate** the
social consequences of
their work?

Embedded EthiCS



Kathleen Creel



Rob Reich



Mehran Sahami

And many more!

The Problem

- Teaching ethics is often not seen as a core part of a CS education
- CS faculty don't necessarily feel prepared to teach ethics
- Philosophy department courses in ethics are not specific to computing
- Even when tech ethics is taught, it is often relegated to a separate course
 - Not integrated into other coursework
 - Can't assume a particular background (e.g., machine learning)
 - Context may not be clear
 - Ethical implications are not taught adjacent to technology
- Creates a pattern that ethical evaluations only take place **after** technology is built

The Opportunity

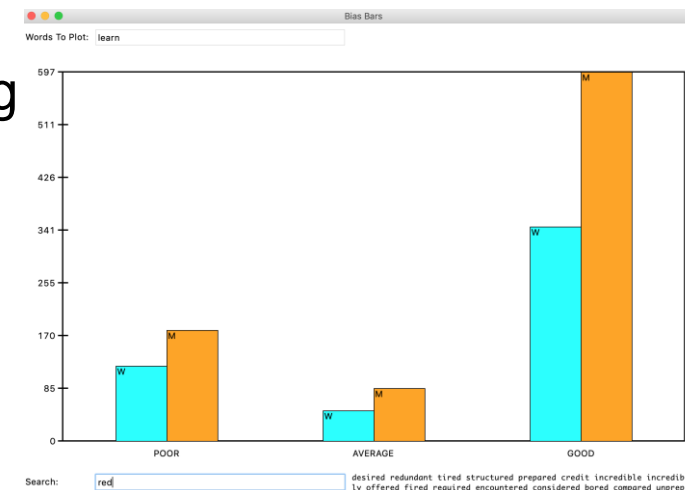
- Teaching ethics ***within*** existing computer science classes
- Bring in external expertise to develop ethics modules
 - Leverage expertise from philosophy
- Topics in ethics should be revisited in ***multiple*** courses
 - Can be integrated into other coursework
 - Allows for assuming specific background
 - Context becomes immediately clear
 - Ethical implications are taught alongside technology
- Creates a pattern that ethical evaluations should be on-going **while** technology is built

The Approach

- Embedded EthiCS
 - Stanford part of a wider consortium of universities, including Harvard, MIT, and Univ. of Toronto
- Hire post-docs with background in ethics and some understanding of technology
- Pair post-docs with grad students in CS
- Develop modules to be taught in existing classes
 - Faculty instructor for course works with team to determine material and fit
 - Post-doc gives (at least) one lecture
 - Also multiple mini-lectures
 - (At least) one assignment leverages material from that lecture
 - Can also add mini-assignments throughout
- Module designed so that faculty instructor can teach it again in the future

The Specifics in CS1

- Katie Creel is Embedded EthiCS Fellow working with us
 - PhD in History and Philosophy of Science
 - BA in CS and in Philosophy
- In first term, paired with Nick Bowman (MS student in CS)
- Module in CS1 course on gender bias in text data
 - Assignment for student was to read a file with course evaluations
 - Evaluations include text comment, gender of instructor, and rating
 - Students build program to plot ratings by gender for terms in text
 - CS1 topics: file reading, string processing, data structures (lists and dictionaries), simple graphics
 - Ethical issues: understanding gender bias in online data
 - Also have some mini-assignments to compute some basic statistics from the data to show gender bias in different ways



The Plan This Year

- Plan to add Embedded EthiCS modules in 5 course this year:
 - Programming Methodology (CS106A – CS1 course in Python)
 - Programming Abstractions (CS106B – CS2 course in C++)
 - Computer Organization and Systems (CS107 – CS3 course)
 - Introduction to Probability for Computer Scientists (CS109)
 - Design and Analysis of Algorithms (CS161)
 - Design for Behavior Change (CS247B – upper division HCI course)
- } Required core courses for all CS undergrads

The Bigger Picture

- Planning Embedded EthiCS modules in 5-6 additional courses next year
- Additionally, have full courses on tech ethics:
 - CS181: "Computers, Ethics, and Public Policy"
 - CS182: "Ethics, Public Policy, and Technological Change"
 - All students required to take on "Technology in Society" courses
 - Either CS181 and CS182 would satisfy that requirement

Thank you for your attention

Q&A and Discussion