Code in Place: Online Section Leading for Scalable Human-Centered Learning

Christopher Piech, Ali Malik, Kylie Jue, Mehran Sahami
Computer Science Department, Stanford University
{piech,malikali,kyliej,sahami}@cs.stanford.edu

ABSTRACT

Could it be the case that the number of people who want to teach computer science, and have the potential, is roughly proportional to the number of people who want to learn? During the time of COVID-19 we offered a free CS1 class to people around the world. Well-aware of the high drop-out rates reported in many massive open-access online courses (MOOCs), we augmented our course with a scalable, human-centered solution: *section leading*. Section leaders teach small, weekly interactive learning sessions. We hypothesize that the personalized attention adds a sense of responsibility for both student and teacher which drives learning. We recruited over 900 volunteer section leaders and more than 10,000 students in the class. To our knowledge this is the largest group of section leaders in a single CS1 course offering and the most small group interactions. The completion rate in our class was more than 10 times that usually reported for similar MOOCs. Additionally, 99% of the volunteer section leaders taught through the entire span of the course, showing the potential for large scale volunteer-driven education, and the benefit that teachers themselves derive. We also discovered the potential for replication of this model, as 34% of students in a representative-sample survey indicated they would serve as section leaders for a future offering of the course. This level of participation would be more than sufficient to field additional offerings of the course sustainably. We believe this is an intriguing case study of a model for significantly scaling human-centric CS education for all.

CCS CONCEPTS

• **Social and professional topics → CS1**.

KEYWORDS

CS1, online learning, teaching at scale

ACM Reference Format:
Christopher Piech, Ali Malik, Kylie Jue, Mehran Sahami. 2021. Code in Place: Online Section Leading for Scalable Human-Centered Learning. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21), March 13–20, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3408877.3432562

SIGCSE '21, March 17–20, 2021, Toronto, Canada

ACM ISBN 978-1-4503-8062-1/21/03…$15.00
https://doi.org/10.1145/3408877.3432562

1 INTRODUCTION

Demand for educational opportunities in computer science remains strong, especially given the large number of unfilled positions in the IT sector [17]. Over the past decade, Massive Open Online Courses (MOOCs) have seen a rapid rise as a potential means for virtually unlimited educational opportunity [9] especially for those who traditionally don't have access [4]. Although MOOCs have attracted millions of students across the globe [18], studies have revealed these courses suffer from signficant issues such as extremely low completion rates [16, 25]. For example, a comprehensive study of over 4 million course participants in MOOCs on the edX platform reported a 5.5% certification rate overall, despite more than half of students indicating an intention to earn certificates in the courses they enrolled in [2].

In the context of the COVID-19 pandemic, the need for educational opportunities in computing took on new urgency. With many workers finding themselves furloughed or unemployed, jobs in IT, which can often be done remotely, offered new potential employment opportunities. Moreover, many people who were sheltering in place might have had more time to dedicate towards learning. This confluence of factors motivated the need for a broadly available online introductory programming class that could more successfully retain students compared to standard MOOCs.

To address this problem, we created "Code in Place" (a play on the phrase "shelter in place"), an introductory online programming class in Python based on the CS1 course at Stanford University. Notably, we went beyond the traditional MOOC model by centering the courses around the idea of *section leaders*. These were skilled volunteers who had experience with the course material, who would meet weekly with small groups of roughly 10 students each to provide more personalized instruction and motivate students to continue in the course. The section leaders also participated in an online discussion forum to answer students' questions, provided guidance and feedback on assignments, and helped foster a positive community for all participants in the course.

The focus of the class was on creating a **human-centered** learning model built around a community that stresses learning for all, the importance of kindness, and peer support.

1.1 Contributions

Our work explores the potential of harnessing the power of section leaders and human-centric education in a scalable online educational setting. We hypothesize that the use of section leaders will lead to higher completion rates in an online course, and more novelly, that it is possible to create a model for section leading that scales to a course supporting thousands of students.

There are many challenges in creating such a human-centered learning experience, most notably recruiting and training large numbers of section leaders, and maintaining a healthy online community. The primary contributions presented in this work are:

(1) Showing that a section-leading model can be scaled far beyond existing practice online.
(2) Teaching (to the best of our knowledge) CS1 with the largest group of section leaders and most small group interactions.
(3) Assessing how well this approach worked in its first iteration.
(4) Positing a model for how this approach might be replicated sustainably in the future
(5) Providing open source materials for recruiting and training section leaders as well as lesson plans for weekly teaching to make this approach more replicable by others.

The structure of this paper is as follows. Section 2 presents an overview of the structure of the course, including details on the process for recruiting, selecting, and training section leaders to allow the course to scale, as well as student selection, material covered, and community development. Section 3 assesses the impacts of this course in terms of student and staff engagement, and provides a description of how this course model may be sustained in future iterations. Section 4 discusses the results and considerations for future offerings of such courses, as well highlights important limitations. We hope that the insights in this paper allow for others to replicate our positive learning experience.

1.2 Related Work

Our work continues on a long history of contributions towards high-quality open-access education for all.

Human-Centered Learning in MOOCs. Prior work has tried to address the problems of low course completion rates and personalized feedback in MOOCs by incorporating a more human-centered learning approach. Meet-ups have been suggested as a way to boost the human element of MOOCs [6]. However, meet-ups are usually not considered an integral part of a course and, in many cases, are only attended by a small fraction of students in a class. Recent research has suggested that social contexts in MOOCs matter in subtle ways [3]. Some educational programming tools have also tried to leverage the human element. For example, the PythonTutor platform allows users to help one another when they are stuck [7]. An extension of this work, CodeOpticon, even supports one-to-many code help [8]. While useful, such tools primarily focus on code-level rather than class-level support.

With regard to providing feedback on students' work, peer assessments [13] (and auto-graders [11]) have also been employed. While such methods can be effective, they cannot provide personalized guidance and support on course material in the way that a human well-versed in the subject can.

Section Leader Model. The use of section leaders—skilled undergraduates who have taken the course before—to scale introductory programming classes has a long history. Originally motivated by a desire to contain costs while scaling class size [21], the section leader model has since been recognized to provide other benefits, such as better fostering a learning community and providing a peer mentoring model [22]. As a result, the use of section leaders has slowly seen more widespread adoption in higher education [5]. Section leading is a form of "near peer mentoring" which has many well known benefits for both learner and teacher [14, 23].

Table 1: Stated reasons for applying to section lead

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%

Compared to peer feedback models, section leaders can provide more skilled guidance on course material and, like any good teacher, can also provide encouragement to students to continue in a course through challenges. Such human interaction can create a greater sense of accountability for students as they know that someone is paying attention to their struggles.

While the use of section leaders has grown in the traditional in-person learning setting, they have not seen much use in large MOOCs. Generally, the use of section leaders to teach live, small section meetings and provide feedback on student assignments is not seen as a scalable model for supporting thousands of students enrolling in free MOOCs.

2 THE CODE IN PLACE COURSE

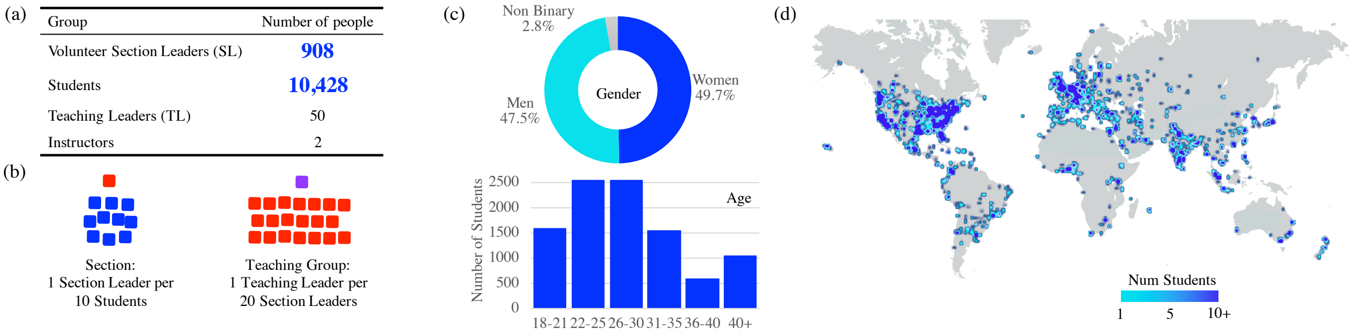
The intentions behind Code in Place were to (i) create a joyful learning experience for students, and (ii) provide an opportunity for community service for those who wanted to share their knowledge of programming during the time of COVID-19. We aimed to build a course which was a meaningful learning experience, while also fostering a positive and generous community.

2.1 Section Leaders at Scale

An important aspect of Code in Place was that it welcomed individuals from all walks of life who wanted to partake in teaching programming as volunteer section leaders. Their commitment was a responsibility to teach a group of roughly ten students once a week for an hour and also to contribute to the online community of learners.

Hiring volunteer section leaders. Perhaps the most formidable challenge in our effort was to hire sufficiently many experienced volunteer section leaders who would allow the course to scale to thousands of students. Based on prior experience, we wanted to maintain a 1:10 ratio of section leaders to students. To recruit section leaders, we created an online application which required the submission of three components: (1) a recorded 5-minute teaching video, (2) a python code debugging exercise, and (3) a short written application including demographic information. Section leading was clearly advertised as a volunteer (unpaid) experience, requiring 5 weeks of participation. The opportunity was advertised broadly through word of mouth, posts on a variety of mailing lists and discussion forums, and was substantially aided by an article in Scientific American [15].

For the teaching video component of the application, applicants were asked to prepare a 5-minute teaching sample on a given Python problem that covered variables, arithmetic, and user input. This teaching demonstration allowed the applicant to motivate the problem and to identify and teach the concepts they thought would be



Group	Number of people
Volunteer Section Leaders (SL)	908
Students	10,428
Teaching Leaders (TL)	50
Instructors	2

Figure 1: (a) Course size. (b) Example "Section" and "Teaching Group". (c) There were more female students in the course than male students and age was well distributed. (d) Students came from around the world.

most challenging for students, while also using the structure of our detailed lesson plan as a foundation.

The debugging exercise presented applicants with a sample student solution to a small programming problem. The student solution had three errors (of low, medium, and high complexity), and the applicants had to identify the bugs and write feedback to the hypothetical student. Great care was taken in developing the prompts for the teaching demo and the debugging problem, resulting in several iterations of testing and refinement before release.

In the space of only four days, we received 1,123 applications. Each section leader applicant was manually reviewed by at least one of 33 different hand-picked evaluators from the Code in Place team over a 12 hour time window[1].

Section leaders applied from 97 countries, were fluent in 64 different languages, and were located in 21 of the 24 time zones (excluding three around the international date line). 8% of the section leaders were current computer science teachers, 39% were university students, and 49% worked in industry as programmers. Applicants represented a diverse spread of institutions, including MIT, Stanford, and Harvard (US), Oxford (UK), Koç University (Turkey), as well as a variety of industry and academic institutions. Applicants were surveyed about their reasons for applying to section lead (Table 1). The most common response was a desire to give back through community service (86% of applicants) followed by a desired to improve teaching skill (72%); applicants saw this as an opportunity to both contribute and learn.

From the over 1,100 applications received, more than 80% of applicants were determined to be above the required standard and were offered a position. In total, 904 section leaders participated.

Training section leaders. While some section leaders had prior teaching experience, many did not. As a result we developed a succinct, community-focused training program for all our volunteers. Since human-centered learning was one of our central tenets, we had to ensure that, on top of solidifying course content, sections fostered a strong sense of community and belonging. In order to align section leaders to these values, we modeled section leader training directly after the structure of sections themselves (see Fig. 1b). Specifically, section leaders were organized into small-groups lead by a highly experienced teaching leader.

Section leaders were asked to attend three training sessions in total: a 30-minute welcome hosted by the course instructors, a 60-minute workshop to prepare for their first section, and a 60-minute workshop after their first section to reflect on how to create an inclusive section culture. They were also welcomed to attend "Section Leader Learning Week", a series of optional 45-60 minute workshops, for additional professional development opportunities after their teaching responsibilities had concluded.

Training focused on good teaching practices as well as instilling the importance of engaged problem-solving in sections. We immersed the section leaders in active learning techniques during the small group workshops by having them critique their small group leader's example lesson and to collaboratively discuss strategies for approaching their first sections. In the final, required training workshop, we used scenario-based training to have section leaders tackle potential situations they might encounter when teaching globally diverse students.

To cater to to diverse teaching backgrounds and emphasize the importance of an inclusive section culture, we chose to ground the training curriculum in collaborative discussions that built off section leaders' existing experiences. All section leaders had the ability to communicate and collaborate with one another via a discussion forum that only included teaching staff, and they used the platform to share teaching tips and ask one another questions about the material for each week. To further elevate the diversity of expertise among the section leaders, we also provided section leaders with the opportunity of volunteering to teach workshops to their peers during "Section Leader Learning Week."

2.2 Student Selection

Application. To manage the student-to-section leader ratio, we had students apply to take part in the course. Applicants had to fill out a short form, read about 5 pages of a textbook, and complete 3 short challenges based on these readings using an online IDE. These challenges required no other prerequisite knowledge. The intention of the reading and challenge questions were to give students a sense of the content and time commitment of the course.

[1] The application and evaluation periods were condensed due to various constraints at the time. Such short application/evaluation periods are not intrinsic requirements nor recommendations for future iterations.

The overall time to complete the application was approximately 1 hour and over 20,000 students completed a full application. When evaluating applications, we looked to see if students (1) used the concepts in the reading and (2) copied and pasted the solutions. By requiring an application, we made sure that the students passed a minimum bar of engagement. Due to certification requirements for interacting with minors, we limited the course to adults.

Students. Code in place accepted 10,428 students from around the world. Of those, 49.7% were women and 47.5% were men. We also had a wide distribution of ages: 15% were under 21, 9.7% were above 40 and the rest were in between. See Figure 1 for more details.

2.3 Course Components

The course ran for five weeks from mid-April to late-May 2020. It was marketed simply as an experience to learn. Notably, we explicitly stated that we would not provide certificates of completion. This latter point was actually a logistical requirement for our home institution. However, it also allowed us to set expectations that this class was about the value of learning programming, not a mechanism to certification. The course included many components to keep students engaged.

Course website. The central hub for the course course was a website where students logged in and were presented with a personalized home page. Their page had a link to their section, as well as course content such as lectures, worked examples, an online textbook [19], and relevant handouts.

Recorded lectures. To provide a cadence to the class, course lecture videos were released three times a week (on Mondays, Wednesdays and Fridays). For each release, approximately 50-60 minutes of lecture content was made available. This content was broken into 10-minute chunks to make it more digestible online. The content was largely recordings of the live (online) CS1 class we were teaching at our university with some slight edits. Overall there was over 14 hours of content split between 74 videos. The content covered the basics of python including variables, control flow, data structures such as lists and dictionaries, as well as images and graphics (including animation).

Weekly sections. Once a week students would meet online in a group of ten with their designated section leader for a 40-minute section. The section leader would lead an interactive learning experience where the students would practice the concepts covered in class that week. Section leaders were provided with detailed lesson plans both to lower their time commitment for preparation as well as to ensure more uniformity and high quality among sections. Section leaders were also encouraged to share their reflections on how their section went with one another to further develop their own teaching skills. Students were assigned to sections first based on time-zone preferences and second based on age similarity to their section leader (to foster a "near peer" teaching model).

Discussion Forum. A key component of the course was a series of richly-featured discussion forums hosted by EdStem.org. The course had a primary forum for all students and section leaders, where students could ask questions that would be answered by the staff and other students, and where staff would post announcements. Each student was also in a smaller discussion forum with their section (10 other students and their section leader). Section leaders had their own discussion forum where they could ask teaching questions and discuss section issues. Students and section leader names were partially anonymized to preserve privacy and email addresses were never required to be shared.

Assignments. Students were given three large assignments which were based on the first three assignments of our university's CS1 course. Students could either complete their assignments in a traditional offline IDE (PyCharm) or through an online interpreter provided by EdStem. The assignments covered (1) an introduction to programming using Karel The Robot (using Python syntax), (2) Python console programs which built up to a program named KhansoleAcademy, which quizzes the user with randomly generated arithmetic questions, and (3) an image processing assignment where students implement image filters and a forest fire detector using pixel-level analysis of images. In each assignment, students were encouraged to go beyond the minimal requirements and add their own creative extensions.

Optional Diagnostic and Final Project. Beyond the three required assignments, students were also given the option to take a diagnostic assessment (i.e., test) mid-way through the course to get a better sense of how well they were understanding the material. This assessment was auto-graded. Additionally, students who wanted a greater challenge at the end of the course were given guidelines for developing open-ended programming projects, with a number of suggestions provided from a standard list. Since the diagnostic and final project were purely optional, we don't include results based on engagement with those options in this paper.

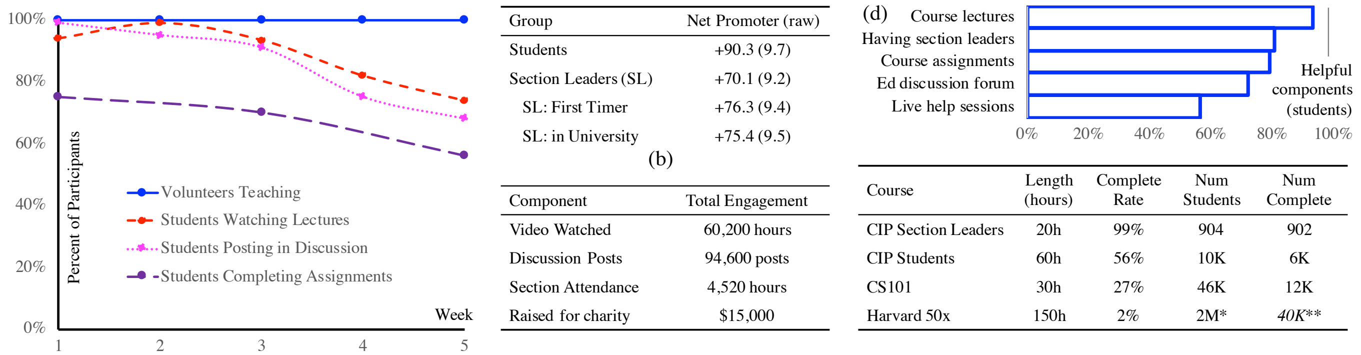
2.4 Course Context: COVID-19

This course was built in response to an exceptional moment in history, when shelter in place mandates went into effect around the world in response to the COVID-19 pandemic. As a result, this was a time of turmoil for many people. In an early course survey, students revealed the impact of this moment on their lives. 87% of students reported a significant life event taking place during the course. Of those, 17% reported a change in employment and 16% reported a change in living situation. The recognition of such upheaval drove the tone for our class. From the first lecture we set course values: Humanity, Intellectual Joy, Social Connection, Gratitude, and "Everyone is Welcome." We kept these values visible on our course webpage. We wanted to instill in both our staff and our students that we are a collective team working for one another.

While we presently turn to measuring more quantifiable impacts of the course, we do note this unique context of the course creates a possible confound for the replicability of our results. As an experience report, however, we primarily aim to highlight the results of the current class offering. Nevertheless, we do believe that many of the results reported here would in large part be applicable in different course contexts as well. While future offerings may or may not occur during the COVID-19 pandemic, the impact of this moment will be felt far into the future.

3 IMPACT

Throughout the course we kept track of student engagement in assignments, lectures, and on the discussion forum. In addition, we offered an end-of-course survey to all students and section



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)

(b)

Component	Total Engagement
Video Watched	60,200 hours
Discussion Posts	94,600 posts
Section Attendance	4,520 hours
Raised for charity	$15,000

(c)

Course	Length (hours)	Complete Rate	Num Students	Num Complete
CIP Section Leaders	20h	99%	904	902
CIP Students	60h	56%	10K	6K
CS101	30h	27%	46K	12K
Harvard 50x	150h	2%	2M*	*40K***

(e)

Figure 2: (a) Engagement over time (b) Would you recommend this experience? Where +70 is "exceptional" (c) Σ engagement (d) percent of students who considered different course components helpful (e) comparison of completion rates to other courses [10, 12]. CIP is Code in Place. *Harvard 50X enrollment is over an 8 year period. **This estimate is a projection.

leaders. We also chose a random subset of 100 section leaders and 250 students (including those who did not complete the course) to directly target with extra motivation and follow-up messaging. The response from this random sample was high, with 89% of section leaders and 92% of students responding. The high, but incomplete response leaves some room for sample bias.

3.1 Student and Teacher Opinions

Both students and section leaders had a high opinion of the experience. The overall evaluation score students gave the courses was measured at $\mu = 4.9$, $\sigma = 0.34$ on a 5-point scale where 4 is "good" and 5 is "excellent". In addition we asked respondents whether they would recommend the experience to a prospective student or a prospective section leader with a similar background to themselves, on a scale of 0 to 10—the standard Net Promoter scale [24]. Students raw recommendation value averaged 9.7, while section leaders averaged 9.2. This corresponded to a Net Promoter Score (NPS) of +90.3 for students and +70.1 for section leaders. Generally an NPS of +50 is deemed excellent, and anything over +70 is exceptional [1]. When we break down the NPS between different subsets of section leaders we find that first time teachers and university students enjoyed the experience the most (NPS of 76.3 and 75.4 respectively), section leaders from "industry" were similarly high (NPS = 72.6) and the NPS was lowest for professional computer science teachers (NPS = 64.1). Students were asked which course components were helpful: the highest rated components in order were lectures (93% of students), section leaders (81%), assignments (79%), and the discussion forum (72%). This high rating of section leaders was commensurate with student responses when asked to rate the "overall quality of my section leader", yielding $\mu = 4.5, \sigma = 0.8$ where 4 is "good" and 5 is "excellent". See Figure 2(b).

3.2 Completion and Engagement

From the start, we were interested in making sure that both students and section leaders completed the Code in Place experience – this was especially important to us since we needed to maintain an appropriate ratio of section leaders to students.

Section Leaders. Perhaps the most surprising result was the incredibly high completion rate among the volunteer section leaders. Out of 904 original section leaders only two dropped out: one because their life became unexpectedly busy and one because they fell ill with COVID-19. This translates to a 99.8% completion rate among the volunteer section leaders over the entirety of the course. To make-up for section leaders who couldn't complete the course, other volunteer section leaders stepped in to teach multiple sections. In preparation for more potential dropouts among section leaders we limited all section leaders to only teach one section (10 students) at the start.

Students. Engagement in the course was surprisingly high. The students (over 10,000 of them) watched over 60,200 hours of lecture (6.02 hours / student) and made 94,600 posts to the discussion forum (an average of over 9 posts per student). For comparison, there were under 3,000 posts to Harvard CS50x's forum during the same time period. We note that about one third of posts were from students own section discussion forums. Over 56% of the original class submitted all three assignments (our definition of completing the course). Given that approximately 80% of students were watching videos and posting in the discussion forum it seems that there were students who either were auditing or not submitting their work. Most of the students who didn't complete the course didn't submit the first assignment. In contrast, 75% of students who showed real engagement with the course by submitting the first assignment went on to also submit the last assignment. Figure 2 (a, c, e) provides more details on course engagement.

Historical completion rates reported in similar MOOCs with a comparable number of hours of instruction are generally low. Harvard's CS50x reports a 2% completion rate [10] while Stanford's CS101 is 27% [12]. We note that comparisons of completion rates need to be interpreted cautiously. These other courses allowed for unlimited enrollment, had much higher numbers of initial students, and presumably had many students who were sampling. Furthermore, the total length of a course impacts completion rates. Figure 2(e) contains a full comparison of these courses with Code in Place. Of course, all these courses are contributing to the same goal of increasing exposure to programming.

3.3 Qualitative Impact

In our experience teaching the class, we encountered many qualitative instances of impact and engagement in the course. Students self-reported that after Code in Place they were able to find employment as software developers, become teachers, and find meaning during a difficult time.

On the discussion forum, we found students engaged in an active and uplifting community centered around learning. Examples include the formation of meme threads to share computer science jokes, baked cookies in the shape of Karel the robot, and the emergence of self-organised volunteer student groups that contributed by translating and transcripting lectures for other students. The forum also had an active student answering community that responded to other peers' conceptual questions.

Students expressed an interest in creating t-shirts as a way to celebrate participation in the class. We decided to turn this into another venue for expressing our shared humanity. To that end, we ran a crowd-sourced project for students to design and select a class t-shirt, whose purchase would also result in a charitable contribution to an organization the students would select. Through a vote, students chose to donate the t-shirt proceeds to https://www.buildon.org/. Over $15,000 was raised for the organization through the sale of over 1,000 t-shirts. Perhaps even more heartwarming was a drive organized by some of the students in the class to purchase t-shirts for other class participants who would have liked a shirt but were unable to afford one (especially those living in countries with low per capita income). This effort led to hundreds of t-shirts being purchased for others as a way for more fortunate students in the class to give thanks and support others.

As the course drew to a close, students expressed interest in follow-up resources or a next Code in Place. We created a master thread of different continuation resources and found this post to be extremely popular amongst students. Students still post daily to the discussion forum, even months after completion of the course.

3.4 A Model for Sustainable Section Leading

In this endeavour we set out to see if there was an opportunity to jointly offer a teaching and learning experience. This initial iteration was limited by the number of section leaders who applied (50,000 students who applied vs. 1,300 section leaders), but dropout mostly affected only students. Is there an opportunity to find a balance between section leaders and students? First a few observations: (1) Given that so few section leaders dropped out, there is less need to have "emergency on call" section leaders which could allow for section leaders to teach multiple sections. (2) A large number of students both completed the course and said that they would like to teach if we were to offer it again. In the post course survey, 34% of the respondents either agreed (16%) or strongly agreed (18%) that they would like to section lead in future iterations. This suggests the possibility of perpetually offering a class like this by recruiting section leaders from previous iterations of the course.

4 DISCUSSION

Every type of good teacher. In Code in Place we confirmed that a truly broad set of people make for excellent instructors. With participation from young university students to retirees, representing numerous countries, all were able to bring inspiration and provide education. We believe there is opportunity to broaden participation not only among learners, but teachers as well.

Section leading as a learning experience. Hands-on teaching exposure is an extremely valuable experience, both for improving teaching ability and to solidify understanding of the course concepts. As such, Code in Place was as much a course for teachers as it was a course for students. By nurturing both students and teachers simultaneously, we were able to create a powerful symbiotic relationship. A generally hard problem in scaling education is how to train teachers – especially, how do we give more people their first teaching experience. For many, their first teaching experience occurred as one of the 900 section leaders of Code in Place.

Sense of responsibility drives completion rates. Why was section leader completion so high? We believe that teaching instills a notable sense of responsibility that holds people accountable. Interestingly, we note that section leaders, who had high completion rates, did not recommend the class to the exceptional level that students did, although the latter had more dropout. To us this suggests that completing the commitment for section leaders was less about “enjoyment" than it was about some function of productive “social pressure". For section leaders, dropping out would mean letting down 10 people for whom they were responsible. In the summer 2020 iteration of “CS Bridge" [20] (a CS1 course offered internationally), many of the ideas and curricula from Code in Place were repeated (and many of the instructors were the same), but students were sourced from high schools and their parents were included in their admission. Section leaders offered office hours in addition to section, and sections were held 12 times instead of 5. In this course, completion increased to 86% for a comparable set of material.

4.1 Limitations

Maintaining section size. For section leaders, any dropout in students is demoralizing. As such, a 56% completion rate leaves many sections feeling diminished by the end of the course. One solution would be to start with more than 10 students per section, anticipating that some students will drop out. However, we believe the biggest opportunity to improve is to raise student retention.

Scaling class size. The choice to host a human-centric class was a difficult one to make as it necessitated that we limit enrollment (in this case to roughly 10,000 students). Of the 50,000 applicants, the second set of 10,000 looked almost identical to the 10,000 we chose to teach. Making this cuttoff was difficult. In the future we would like to offer an open-source companion set of materials for students who were not admitted to the class. In the first offering, there was not enough time to provide such companion materials. Increasing the number of section leaders by extending the application window to more than 4 days could also help address this issue.

4.2 Conclusion

Code in Place was an uplifting community service project. Through the experience we showed a proof of concept that many people who are traditionally not given the chance to teach are capable of doing so. This demonstrates a model for scalable human-centric CS1 education, part of our mission of CS4All.

4.3 Acknowledgements

Code in place was special because of the incredible team that put on the project. It was a volunteer project, run by a large group of kind individuals working towards a common cause. In addition to the section leaders, dozens of people volunteered their time to make the course work. We deeply appreciate all of their work, and are thankful for the chance to get to work with such as special team. We would like to thank: Lisa Einstein, Ana Saavedra, Yosefa Gilon, Laura Mapstone, Brahm Capoor, Zach Birnholz, Jennifer Widom, John Mitchell, Kate Rydberg, Scott Maxwell and the EdStem team, Richard Freling and the CodePost team, Isaac Pohl-Zaretsky, Jason Ford, and Nathan Dalal. This is a short list of the fantastic team: for more details see the code in place website.

REFERENCES

[1] Swetha Amaresan. [n.d.]. What Is a Good Net Promoter Score? https://blog.hubspot.com/service/what-is-a-good-net-promoter-score

[2] Isaac Chuang and Andrew Ho. 2016. HarvardX and MITx: Four Years of Open Online Courses – Fall 2012-Summer 2016. *SSRN* (2016). http://dx.doi.org/10.2139/ssrn.2889436

[3] Dan Davis, Ioana Jivet, René F Kizilcec, Guanliang Chen, Claudia Hauff, and Geert-Jan Houben. 2017. Follow the successful crowd: raising MOOC completion rates through social comparison at scale. In *Proceedings of the seventh international learning analytics & knowledge conference*. 454–463.

[4] Tawanna Dillahunt, Zengguang Wang, and Stephanie D Teasley. 2014. Democratizing higher education: Exploring MOOC use among those who cannot afford a formal education. *International Review of Research in Open and Distributed Learning* 15, 5 (2014), 177–196.

[5] Jeffrey Forbes, David J. Malan, Heather Pon-Barry, Stuart Reges, and Mehran Sahami. 2017. Scaling Introductory Courses Using Undergraduate Teaching Assistants. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '17)*. 657–658. https://doi.org/10.1145/3017680.3017694

[6] Michael E. Goldberg. 2015. MOOCs and meetups together make for better learning. https://theconversation.com/moocs-and-meetups-together-make-for-better-learning-35891

[7] Philip J Guo. 2013. Online python tutor: embeddable web-based program visualization for cs education. In *Proceeding of the 44th ACM technical symposium on Computer science education*. 579–584.

[8] Philip J Guo. 2015. Codeopticon: Real-time, one-to-many human tutoring for computer programming. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 599–608.

[9] John Hennessy. 2012. Online Education: The Coming Tsunami? CRA Snowbird Conference presentation. http://archive2.cra.org/uploads/documents/resources/snowbird2012_slides/hennesy.pdf

[10] Andrew Ho, Justin Reich, Sergiy Nesterko, Daniel Seaton, Tommy Mullaney, Jim Waldo, and Isaac Chuang. 2014. HarvardX and MITx: The first year of open online courses, fall 2012-summer 2013. *Ho, AD, Reich, J., Nesterko, S., Seaton, DT, Mullaney, T., Waldo, J., & Chuang, I.(2014). HarvardX and MITx: The first year of open online courses (HarvardX and MITx Working Paper No. 1)* (2014).

[11] Mike Joy, Nathan Griffiths, and Russell Boyatt. 2005. The Boss Online Submission and Assessment System. *J. Educ. Resour. Comput.* 5, 3 (sep 2005), 2–es. https://doi.org/10.1145/1163405.1163407

[12] René F Kizilcec, Chris Piech, and Emily Schneider. 2013. Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. In *Proceedings of the third international conference on learning analytics and knowledge*. 170–179.

[13] Chinmay Kulkarni, Koh Pang Wei, Huy Le, Daniel Chia, Kathryn Papadopoulos, Justin Cheng, Daphne Koller, and Scott R. Klemmer. 2013. Peer and Self Assessment in Massive Online Classes. *ACM Trans. Comput.-Hum. Interact.* 20, 6, Article 33 (Dec. 2013), 31 pages. https://doi.org/10.1145/2505057

[14] Jean Lave. 1996. Teaching, as learning, in practice. *Mind, culture, and activity* 3, 3 (1996), 149–164.

[15] Einstein Lisa. [n.d.]. Help Profs Teach Stanford's Popular, and Now Free, Online Intro to Coding Course. ([n. d.]).

[16] David J. Malan. 2016. This was CS50x. https://medium.com/@cs50/this-was-cs50x-82be0995862b

[17] U.S. Bureau of Labor Statistics. 2017. Employment Projections. https://www.bls.gov/emp/tables.htm

[18] Eren Orbey. 2020. How Harvard's Star Computer-Science Professor Built a Distance-Learning Empire. *The New Yorker* (July 21 2020).

[19] Chris Piech and Sami Abu-El-Haija. 2020. Human Languages in Source Code: Auto-Translation for Localized Instruction. In *Proceedings of the Seventh ACM Conference on Learning@ Scale*. 167–174.

[20] Chris Piech, Lisa Yan, Lisa Einstein, Ana Saavedra, Baris Bozkurt, Eliska Sestakova, Ondrej Guth, and Nick McKeown. 2020. Co-Teaching Computer Science Across Borders: Human-Centric Learning at Scale. In *Proceedings of the Seventh ACM Conference on Learning@ Scale*. 103–113.

[21] Stuart Reges, John Mcgrory, and Jeff Smith. 1988. The effective use of undergraduates to staff large introductory CS courses. *ACM Sigcse Bulletin* 20, 22–25. https://doi.org/10.1145/52965.52971

[22] Eric Roberts, John Lilly, and Bryan Rollins. 1995. Using undergraduates as teaching assistants in introductory programming courses: An update on the Stanford experience. *SIGCSE '95*, 48–52. https://doi.org/10.1145/199688.199716

[23] Laura S Tenenbaum, Margery K Anderson, Marti Jett, and Debra L Yourick. 2014. An innovative near-peer mentoring model for undergraduate and secondary students: STEM focus. *Innovative Higher Education* 39, 5 (2014), 375–385.

[24] Jenny Van Doorn, Katherine N Lemon, Vikas Mittal, Stephan Nass, Doreén Pick, Peter Pirner, and Peter C Verhoef. 2010. Customer engagement behavior: Theoretical foundations and research directions. *Journal of service research* 13, 3 (2010), 253–266.

[25] Erin L Woodhead, Preston Brown, Susan Snycerski, Sean Laraway, Nicholas Bathurst, Greg Feist, and Ronald F Rogers. 2017. An Examination of the Outcomes of a Brief and Innovative Partnership: SJSU and Udacity. *Innovative Higher Education* 42, 5-6 (2017), 463–476.