Co-Teaching Computer Science Across Borders: 
Course-In-A-Box for International Intro Programming

Chris Piech¹, Lisa Yan¹, Lisa Einstein¹, Baris Bozkurt², 
Ana Saavedra¹, Eliska Sestakova³, Ondrej Guth³, Nick McKeown¹  
¹Stanford University, ²Izmir Demokrasi Universitesi, ³Czech Technical University  
piech@cs.stanford.edu, nickm@cs.stanford.edu

ABSTRACT

Programming is fast becoming a required skill set for students in every country. We present CSBridge, a model for cross-border co-teaching of CS1, along with a corresponding open-source course-in-a-box curriculum made for easy localization. In the CSBridge model, instructors and student-teachers from different countries come together to teach a short, stand-alone CS1 course to hundreds of local high-school students. The corresponding open-source curriculum has been specifically designed to be easily adapted to a wide variety of local teaching practices, languages, and culture. In the last six years, the curriculum has been used to teach CS1 material to over 1,000 K-12 students in: Colombia, Czech Republic, Turkey and Guinea. Joint teaching creates a positive high-quality learning experience for students around the world and a powerful, high-impact professional development experience for the teaching team—instructors and student-teachers alike.

KEYWORDS

Course-in-a-box; co-teaching; cs for all

1 INTRODUCTION

CS education has made substantial progress towards the goal of CS for All—in the United States, online, and in some regions of the world. However there is mounting evidence of a growing global digital divide where access to CS education is heavily dependent on which region you were born in [5]. Despite progress, out of the more than 1.3 billion K-12 students in the world [33], a small fraction will have the opportunity to learn to code. Though online learning gives access to CS, the post-mortem on massive online open-access courses (MOOCs) suggests that human teaching remains an essential element for successful education. The CS education community is thus faced with a challenge: how can we scale high-quality in-person and culturally relevant education more evenly across the globe? For students of all genders, socio-economic backgrounds and country of origin.

Scaling quality in-person CS education globally has two major barriers: First, good learning ideas have trouble spreading across borders; one teacher’s great classroom idea will experience substantial friction when adapted to other classrooms [11]. These challenges are exacerbated across teaching contexts in different countries; as a result, cultural, pedagogic, and organizational know-how may be siloed within countries and institutions. Second, many students live in communities without trained CS teachers and as such there is a “chicken and egg” problem—the lack of teachers with the required content knowledge prevents students from becoming future teachers [3, 31]. We note that both of these challenges exist in all countries—including in the United States [34].

With the goal of CS for All Countries, we developed CSBridge, a curriculum designed to support cross-border co-teaching. The course is jointly taught to students in diverse communities around the world in order to reinforce and bootstrap local communities of CS practice and to reduce the friction preventing the spread of curricula and pedagogy across borders. We discuss two key contributions in this paper.

1. CSBridge Course-in-a-Box. The CSBridge curriculum is an open-source Java-based course built around the popular ACM Java libraries and informed by decades of shared ideas in the SIGCSE community [28]. The course assumes no prior knowledge and is built so that all students can—by the end—build the Breakout game from scratch [22] and make their own creative final project. The primary learning goal is to inspire students to choose to study computer science in higher education. It is intentionally designed to be a high-quality course for any country; unlike traditional curricula, CSBridge has been especially tailored to be linguistically simple and easy to translate and localize. The contribution of this paper includes CSBridge as an open-source and free to use CS1 curriculum-in-a-box: assignments, a clonable website, slides, lesson plans, etc. It is meant to be usable by expert and novice teachers alike.

2. Cross border co-teaching. In CSBridge, university instructors and undergraduate teaching assistants (UTAs) from different countries join together to collectively teach a course in each country. We argue that co-teaching is a surprisingly effective way to facilitate open transfer of pedagogy ideas, cultural understanding, and technical know-how. Our model bridges communities that otherwise would have few channels for sharing CS teaching insight.
Since 2014, we have used CSBridge to teach ten courses in five countries to over 1000 students. The curriculum has also successfully been adopted by four external courses representing a diverse set of stakeholders: large research institutions, small community colleges and PeaceCorp volunteers. We evaluate the effectiveness of the curriculum and explain the multi-layered benefits for students, UTAs, and professional instructors. We then discuss details on how to implement your own version of cross-border co-teaching. This paper was written by eight instructors from around the world who were brought together by CSBridge.

1.1 Motivation and Course Philosophy

**Inspiring + Practical Hands on Learning.** Decades of work in the broader SIGCSE community has produced CS1 assignments that enable students to learn-by-doing while inspiring them to want to learn more [16]. Engaging coursework can demystify CS and—if presented at the right time—can help students chose to pursue computer science as a profession. We would like to inspire beginner students to not only use software, but also create it. Our initial intention was to bring these experiences to a broad set of learners from around the world in a culturally appropriate way.

**Fostering Near-Peer Teaching.** The use of near-peer, student teachers has been used in a wide variety of CS1 contexts [9, 24, 25, 29]. This teaching model has two advantages for students—cognitive and social congruence. In CSBridge, we use undergraduate teaching assistants (UTA for short) as near-peer teachers. Because UTAs themselves are in the process of becoming experts, they often can explain difficult problems to students in the same social and cultural language and can also serve as close role models. The support communities that form around near-peer teaching can be of critical importance for learners.

We believe that co-teaching is an exciting opportunity. Through this experience, we hoped to inspire UTAs to experience the joy of and consequently identify as—and aspire to become—professional CS instructors. Moreover, the co-teaching, visiting [redacted] university has 20 years of experience with a computer science UTA program. This program shares best practices for training UTAs across cultures so that they can be adapted to CSBridge and future courses during and the academic year.

**Transfer of organizational know-how.** There is a pressing need for better methods of sharing pedagogy ideas and (the equally important) associated institutional knowledge and technical infrastructure. In education, good ideas surrounding pedagogy, accessibility to CS, and institutional organization are often lost in translation, as much of an idea’s effectiveness is often in the details and situational context. The need is especially pressing for large-audience university courses run by a large staff team. In concert with the SIGCSE community, [redacted] university has taught CS1 to tens of thousands of students. We would like to exchange expertise in CS education with other countries in order to gain a better understanding of how to make CS accessible to many diverse social and cultural contexts.

**Participation in CS for All.** Every country in the world should be working towards fostering a thriving software industry within its economy. While huge strides have been made in online education, large swaths of the world lack the prerequisite skills and technology to access the benefits of online education. CSBridge can serve as a catalyst or spark for people to begin engaging in independent learning. Doing so requires work across borders. We choose to join together to collaboratively teach a course, as opposed to having outsiders dictate decisions to local hosts. To that end, our model is built for sharing knowledge both ways.

1.2 Prior work

CSBridge presents a novel combination of ideas: co-teaching, how to create inclusive and engaging curricula and international CS teaching. As such it builds upon a large body of work from several fields of thought in the computer science education community.

Learning to program for non-English speakers presents substantial challenges [7, 12]. Efforts to address this need have been mainly online: AppInventor and Code.org [15] are block based programming languages that are popular for K12 students around the world and have been translated into many languages [35, 37]. AppInventor’s ability to deploy student code to android phones is a powerful feature. Unfortunately block-based languages didn’t suit the needs of co-teaching CS1 as most international university courses are currently in languages like Java. There have been many projects to teach programming outside the US: Eric Robert’s Bermuda project [2], AddisCoder etc [26, 36].

Coteaching has a rich literature [13] especially with respect to inclusive education [19, 32]. We believe it could be one way to address the identified issues in diffusion of technology [1, 20, 27].

Service learning, where students travel to other countries for community service projects, as a way to enhance social good in computing education is well-practiced [4, 6, 10, 14, 23]. Service learning can be beneficial for students to learn about social responsibility and empathy, but can have negative impacts on the host community [21]. Cross border co-teaching brings many of the same benefits for the hosts and guest community. However, CSBridge from a service project as it is a joint project between colleagues from different countries, mitigating the costs.
1.3 Applied Contexts

We emphasize that CSBridge is designed to be as broadly useful as possible. Having said that we have directly implemented the course and corresponding model in four specific contexts (Figure 1). Three of the four courses were taught at universities; the fourth offering was taught in the town of Koumbia, Guinea, a 12-hour drive from the capital. The village had a newly built computer cluster but with intermittent electricity and internet.

- **Turkey**: Taught (in English and Turkish) four out of six times in at AAA University in Istanbul, Turkey. 200 students per course from high schools across the country.
- **Colombia**: Taught (in Spanish) once at Universidad de BBB in Bogotá, Colombia. 100 students per course from high schools across the country.
- **Czech Republic**: Taught (in English) twice at CCC University in Prague, Czech Republic. 100 students per course from high schools around the country.
- **Guinea**: Taught (in French) once at Koumbia High School in Koumbia, Guinea. 20 students.

**External courses.** The curriculum is free-to-use, and as such it has featured in a series of other courses in Izmir and Istanbul, Turkey; Curitiba, Brazil; and Nairobi, Kenya. The CSBridge curriculum was itself inspired by [redacted] a course taught at [redacted] University in the United States to around 1,800 students a year. The improved course material that arose from cross-border co-teaching has now been integrated back into the original course.

**Pacing.** All four courses were taught over the summer in an intensive 2-3 week, full-day course. This structure best fits the needs of the host and instructors as well as the local students. The pacing of the course can easily be modified, and a majority of the external courses that have used the curriculum have taught the material over a longer period of time.

**Case studies.** This paper describes in detail the three courses hosted at universities, which were largely the same. We also share experiences from the Guinea course, where we adapted CSBridge to a non-university, rural context.

2 INTERNATIONAL CS1-IN-A-BOX

The course assumes no prior CS knowledge and aims to equip students with a sound understanding of basic concepts of programming and the skills to author an interactive game in Java within two weeks, though it could be expanded as needed. Given the limited time, the program is highly efficient. Since the core learning goal is to inspire students to further pursue computer science, it is presented as a series of engaging, educational coding tasks.

The course-in-a-box that we provide, as the name suggests, has all the resources that a novice or expert instructor would need to teach their own instance of the class. It comes with a personalizable website, code, slides, assessment tools, and of course assignments. Moreover we include tools to help run a team of section leaders and lesson plans with fun activities away from the computer. The content can be accessed on [redacted URL].

### 2.1 Built for localization

Language of instruction plays a key role to guarantee equitable access. Students learn CS better when it is taught in their mother tongue [7, 8, 12, 18], and as such CSBridge teaching and learning materials is designed to adapt to the local language of the participating students. The core material is written in English, and comes with a program (powered by Google Translate, and adjustable by human translators) to translate the entire course into non-English languages. The base curriculum from which we translate is low word count: assignments, APIs, documentation and slides are all as visual as possible and any explanation is written in Simplified Technical English [17]. This makes it easy to (1) translate, and (2) read for non-English speakers. We have translated the entire course into Spanish, French and Turkish. For Spanish and French, we also fully translated the student Java APIs (Karel, console, and graphics) from the ACM Java task force libraries [28, 30]. For example, in Spanish the method `println()` became `imprimir()`. We have published this automatic translating tool, from which anyone can then override the APIs or text and thus translate into a new language.

The course activities are designed to be relevant and interesting in different countries. K-12 students are motivated in surprisingly similar ways around the world—for example, Karel the Robot is loved everywhere. For certain projects the assignments are written to be culturally translated. As an example in Prague, students have Karel repair Charles Bridge, whereas in Turkey they repair Efes. The challenge is the same, but the motivation is localized.

### 2.2 Course plan

The course covers Karel, Console, Graphics and Events to accelerate students to the point where they can write exciting graphical programs in Java (see Figure 2 for the units and assignments). We fit the course into a two week intensive program. The first day introduces control structures, loops and method/function writing via Karel the Robot: a kick-start for a beginner, that allows for engaging problem solving from day one. The rest of the first week covers all the required concepts to start designing the Breakout game [22]. The second week is dedicated to implementing Breakout and building a creative project. All assignments are designed with extensible options for students to be creative and go above and beyond the core goals (a path which most students take). This

<table>
<thead>
<tr>
<th>Unit</th>
<th>Assignment</th>
<th>Unit</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Karel</td>
<td>Collect Newspaper</td>
<td>4. Events</td>
<td>Running Tall</td>
</tr>
<tr>
<td>2. Console</td>
<td>Build Charles Bridge</td>
<td>5. Breakout</td>
<td>Target</td>
</tr>
<tr>
<td></td>
<td>Karel Repair</td>
<td></td>
<td>Smart Film</td>
</tr>
<tr>
<td></td>
<td>Random Circles</td>
<td></td>
<td>Making Tracks</td>
</tr>
<tr>
<td></td>
<td>Random Circles</td>
<td></td>
<td>Mouse Location</td>
</tr>
<tr>
<td></td>
<td>Random Circles</td>
<td></td>
<td>Breakout</td>
</tr>
<tr>
<td></td>
<td>Random Circles</td>
<td></td>
<td>Sorted Numbers</td>
</tr>
<tr>
<td></td>
<td>Random Circles</td>
<td></td>
<td>Show</td>
</tr>
<tr>
<td></td>
<td>Random Circles</td>
<td></td>
<td>The Line</td>
</tr>
</tbody>
</table>

Figure 2: 2w curriculum from Karel to open-ended Graphics.
design facilitates even pacing across a heterogeneous group of students, and it encourages creativity in programming early on.

In the 2-week course the daily schedule includes a morning and an afternoon lecture, a morning discussion section—where UTAs lead a meeting with students for conceptual discussion—and morning and afternoon labs. Every day, an average of five programming tasks are implemented by the students during the labs. One of the most critical activities is the section; UTAs meet for an hour with 8-10 students as a group to consider possible strategies for a programming task and to review and discuss course concepts. These sections are essential to build a small communities united towards the same target goals—learning the concepts and solving a problem—under the guidance of a mentor UTA. It is also the activity that trains the UTA as an effective educator, communicator, and future instructor.

In the second week, the program hosts invited talks on different topics to increase student exposure to the broader domain of CS. We include modules on the inner workings of the internet, artificial intelligence, and a gentle introduction to JavaScript for web development. We also host chats where students can get to know the local and visiting instructors, as role models. This is an opportune time to encourage women in computing.

3 CO-TEACHING ACROSS BORDERS

Because human teaching remains an essential element for successful education, it is important to collectively develop our ability to educate. Early-career instructors usually begin by teaching in the same way as they were once taught; improving and innovating upon teaching style tends to be a slow process of trial and error.

On the other hand, co-teaching can be more efficient and faster way to spread high-quality pedagogy, possibly across cultural boundaries. Teachers can compare, discuss and even inspire among themselves. Moreover, less experienced teachers can more than just learn. With their new and young point of view, the young teachers can inspire their experienced counterparts as well.

We use co-teaching for both our instructors and for UTAs (Figure 1b). All teachers come together as a team to help make the course as effective as possible for the students. The instructors select the material and alternate giving each lecture. Visiting and local UTAs run labs together and are jointly responsible for helping students when they get stuck. In Colombia and Guinea, recruitment of the teaching team has as a requirement fluency in both English and the local participant language. In Prague and Istanbul, the class was offered in a mix of simplified English and the local language; as such, students needed basic proficiency in English.

3.1 Financial considerations

The CSBridge model is built to be as economical as possible. The financial requirements are primarily salary for both instructors and UTAs (hired at a student-teacher rate of 1:10), followed by flights for guest teachers and guest UTAs. We note that employing UTAs dramatically reduces the cost, among other pedagogical benefits. The host campus provides access to computer labs and facilities at no cost. To minimize additional costs, we suggest that visiting teachers use home-stays or any accommodation the host university may have available.

The total budget for teaching a two-week, full-time class of 200 students comes out to approximately $12,000, Budgeting $500 for round-trip flight costs per visitor, $1,000 for instructor pay, and $200 for UTA pay. We do not charge students as we wanted to increase access to CS for all. To break even, each student would have to pay $60, which could be a reasonable way to balance the costs. Our budget has been graciously covered by a private benefactor who believes that the in-person co-teaching model is a highly impactful use of charitable funds.

3.2 What it takes to get started

Since the course-material is ready to go, the biggest step to starting your own instance is to develop a working relationship between the host and visiting community (around one year prior to the course). Three of our partnerships were created by students/colleagues currently at [redacted] university who were from the host country. One of our partnerships was created by "cold-emailing" a university. Once the partnership is established, there are many paths to developing a healthy course. Our next step was to select instructors from both universities and to have them remotely plan the course together. Each party individually recruits UTAs; the local partner recruits students. The course-in-a-box includes a detailed list of timelines that we use before the course starts.

4 EDUCATIONAL OUTCOMES

The course benefits more than just the students who are taught. It is productive for the teachers involved and the universities. Student learning is evaluated by analyzing their work and pre-post surveys. Moreover the student final projects tell their own story of the progress that student’s made.

4.1 Student Outcomes

Assignment learning goals. The student projects are designed such that as students program, their work is saved to a local git repository. When students upload their final submission, in addition they submit the entire process by which they solved the problem. This offers us a rich and unique way to measure if students are achieving the learning goals. We analyzed the git repositories for Breakout submitted by HS students in Colombia and Prague (Figure 3a) and compared them to repositories of [redacted] university students. In all cases, the completion rate was high (95% in both CSBridge and the US university). More impressively, the students at CSBridge students from Prague (µ = 5.8) and Bogota (µ = 6.2) were on-average able to complete their assignments in significantly (p < 0.001) less time than university students (µ = 7.4). After the CSBridge program the university used the improved pedagogy that came from co-teaching and time to complete Breakout subsequently dropped (µ = 6.7). This is noteworthy, especially considering that students at the university had substantially more self-reported prior experience.

Interest in CS. To further understand the impact we were having on students and teachers we conducted a series of surveys. In the Istanbul course, from 2015 onwards, we performed a mandatory pre/post survey for the students and one survey for the UTAs. In July 2017, we surveyed high school students from prior years to assess if the program had lasting impact and motivated CS study
While we observe a large fraction of students intending to pursue a degree in computer science, we find anecdotal evidence that students are especially proud of their submissions to this program. Many students build interactive games (Figure 3b), where they utilize ACM graphics and web images to demonstrate a strong understanding over programming logic and design. We find anecdotally that students are especially proud of their submissions to this open-ended project, and instructors share creative, innovative submissions at course conclusion.

### 4.2 Undergraduate Teaching Assistant growth

We surveyed 20 UTAs in July 2017 to understand how the program affected their career choices. Figure 3b illustrates the change in Likert item responses with bootstrapped 95% confidence intervals. Their experience was overwhelmingly positive, and they enjoyed the co-teaching community. The results demonstrate small improvements in identity and perceived collaboration in the field of CS, and a large increase in teaching and leadership confidence.

Anecdotal comments reveal that the experience has been formative to confidence and has inspired many UTAs to continue teaching (see Appendix). The UTAs form friendships which shape their teaching identity. Many UTAs indicated interest in teaching in the future, either as an academic or as community service, and some mentioned that the UTA group was incredibly supportive and fun. In fact, almost half of our UTA staff from year to year and several graduated UTAs working in industry take vacation time to help with CSBridge in following years.

Four of the United States UTAs have gone on to become full-time lecturers in universities, both public and private. Several UTAs are contributing to open-source teaching materials.

### 4.3 Instructor Outcomes

Co-teaching has proven to be a great way of spreading high-quality pedagogy and inspiration: **Clear communication** in CSBridge course, it is crucial to explain each topic in the simplest way as possible: We are teaching HS students from a range of cultural and linguistic backgrounds. Co-teaching has been a great way of finding the simplest explanations efficiently. This was especially apparent in the development of metaphors for teaching CS1 concepts. For example, variables are like boxes, methods are like toasters, pointers are like URLs. By co-teaching, we can also easily discuss challenges and together come up with better way of explanation.

**Enjoyable teaching** Co-teachers inspire each other and develop new and better ideas how to prepare lectures attractive to students. For example, this year Czech lectures used special lecture themes and their slides will now be used around the world. Regardless of what country you are in, students really appreciate any fun during the lecture. We inspire each other to include jokes, personal experiences as part of lecturing. Some ideas have to been seen to be shared, such as how to handle questions from advanced students so as to avoid alienating beginners.
The results of this incredible professional development experience are largely intangible: we are all now connected to a growing community of teachers who care about the craft of CS1 education. The instructors of csbridge are writing this paper together as a testament to what we learned.

4.4 University Outcomes

For all universities, a major outcome is increasing interest in CS among high school students which also implies increasing number of students to enroll in CS1.

At AAA University, following the first two successful CSBridge offerings, the university decided to adapt the UTA program to the CS1 course of 180 engineering students during the academic year. In order to design a system appropriate for their own cultural context, the instructors visited [redacted] University to observe in-person the UTA program. The local instructors and UTAs involved in co-teaching in CSBridge formed the team to run the very first UTA program in Turkey, which has run successfully in the AAA CS1 course for four semesters. Consequently, AAA University is now planning to integrate UTAs in other engineering courses.

In addition, the AAA UTA group has formed a close-knit student group dedicated to designing and running new teaching activities. The student group has carried out two teaching and social responsibility projects: a weekend-based CSBridge external program for local high school students, and an unofficial CSBridge-inspired CS1 course in Python for non-engineering Koç undergraduates. Instructors noted that UTAs exhibited high individual growth—increased self-reliance, improved communication skills, and improved coding style.

5 GUINEA CASE STUDY

Does CSBridge work outside of a university context? In Koumbia we had the opportunity to adapt the program to an especially high-need, low-resource context. One of the authors of this paper had extensive experience teaching science in Koumbia, a farming village in rural Guinea, West Africa, where most families live without electricity or running water on less than a dollar a day. While students rarely touch computers, smart phones are ubiquitous.

Students and teachers alike expressed a desire to learn to use their technology productively both for the sake of learning and in order to have access to job opportunities. As such, in 2018 AJDE installed a solar-powered technology center with 10 laptop computers. CSBridge collaborated with AJDE to expand their technology trainings to coding trainings, executing a three-week CSBridge camp for 20 high school students ages 13-20. The teaching team consisted of one of the authors as instructor and three UTAs (one Guinean recent university graduate, one Guinean advanced high school student, and one current PeaceCorp Volunteer in Koumbia).

Given the unique context, the structure starkly differed from the other three CSBridge programs: The curriculum was ported offline, and there were more adjustments to adapt assignments to local scenarios. For Newspaper-Picking Karel, the first programming task, in Koumbia there are no newspapers; the program was consequently changed to have Karel get water from a well.

In Guinea, we paced the CSBridge program around the existing infrastructure, computer proficiency, and cultural realities including daily power outages, and student’s familial responsibilities. The curriculum was completed through the Events segment, at an average of two new concepts and two new programming tasks daily. While students did not build Breakout, they still enjoyed independent projects—creating their own creative Graphics, Karel, and Console projects. Students were happy to gain employable skills, learn more about computers and build things; they were also excited to teach their friends and have fun. Several expressed interest in teaching for-pay in their own computer learning labs. UTAs similarly appreciated the chance to augment their learning. The program has inspired two new iterations in the coming year, and PeaceCorp Guinea held a training session to train its volunteers to be code-instructors. We plan to contribute to the national-scale effort for CS in Guinea.

From this experience, we emphasize that for future adaptations of CSBridge to sociocultural contexts like Koumbia’s, a slow and flexible pace is key. By pacing the course to the students, even those with basic computer skills (e.g., typing and mouse use) were able to complete programming tasks and benefit from CSBridge.

6 DISCUSSION

Cross-Cultural Insights. In some cultures (e.g., Czech, Turkey, and Guinea), the teacher-student relationship—and, consequently, any communication—tends to be very formal. This can even be connected with language such as Czech, where the border between formal and informal way of speaking is strict and context-dependent. As a result, there can be a communication barrier between teachers and students. Although Czech students do ask questions during lectures and other teaching forms, this cultural-induced barrier may be the reason why there are less discussions and interactivity during lectures compared to a United States-based environment. All four contexts initially expressed doubts that UTAs would be accepted by students, however students and UTAs formed similar, positive near-peer relationships in all countries.

Language. In Colombia and Guinea instruction needed to be in Spanish and French. Not all of the Colombia team were native speakers, however having students learn in their mother-tongue outweighed the costs of a few mistranslations.

Both ways. All universities learned pedagogy from each other and as such improved their CS1 offerings. Moreover, we all learned about country nuances of CS-education: university entrance requirements, regional differences in access, etc.

External use. CSBridge has inspired spinoffs in Kenya (NaiCode) and Brazil (ZZZ.com). One of the authors of this paper, an instructor who taught in two CSBridge-Istanbul events is also initiating a UTA program in his new department: Izmir Democracy University Electrical Engineering Department. Peace Corps Volunteers in Guinea have been adapting the CSBridge course into an after school programs. We invite anyone to use and extend our work.

---

The local Association des Jeunes pour la Défense des Droits des Enfants (AJDE) hypothesized that if parents saw their children learning job-marketable skills such as coding at school they would be motivated to support their children’s education. In the 2019 school year, high school students learned typing, Excel, and Word.
7 CONCLUSION

In this paper, we presented CSBridge, a CS1 Java-based curriculum-in-a-box translatable to different languages and adaptable to multiple cultural contexts. When paired with our model for cross-cultural co-teaching, we take one step closer to the goal of CS for All Countries, by sharing and adapting CS educational experiences in many different contexts. Across four different countries, CSBridge has positively impacted high school-level students and teachers alike. The SIGCSE community can use the published curricula and its associated cultural translation tool to facilitate creative exploration in CS1 in different communities. Through the in-person, human connections enabled by CSBridge, we hope to contribute to an international community of csed-practice.

REFERENCES


