Computer Science Principles for Teachers of Deaf Students

Richard E. Ladner
Paul G. Allen School
University of Washington
Seattle, WA, USA
ladner@cs.washington.edu

Andreas Stefik
Dept. of Computer Science
University of Nevada, Las Vegas
Las Vegas, NV, USA
stefik@unlv.edu

Jill Naumann
Model Secondary School for the Deaf
Washington DC, USA
jill.naumann@gallaudet.edu

Erin Peach
Code.org
Seattle, WA, USA
erin@code.org

***Abstract*—A major goal of AP Computer Science Principles (CSP) is equity, that is, that all students should have the opportunity to learn computer science at a basic level. In this experience report, we explore how well the Code.org version of AP CSP meets the needs of Deaf students. We report on a professional development workshop for 14 teachers that teach at schools for the Deaf or in Deaf programs in mainstream schools. These schools and programs use the bilingual approach to teaching with instruction in American Sign Language (ASL) and other resources (e.g., textbooks, workbooks, videos, websites, computer apps, exams) in English. Synthesizing the experiences and advice of the teachers and workshop staff, we offer lessons learned for CS teachers in schools for the Deaf and Deaf programs in mainstream schools, mainstream CS teachers who may have one or a few Deaf students in their classes, and AP CSP content providers.**

***Index Terms*—Computer Science Principles, Deaf, English Language Learners, Bilingual, Professional Development**

I. Introduction

Students who are Deaf [1] in the US commonly have hearing parents and are the only Deaf person in their family. They often learn American Sign Language (ASL) from peers or at school. Consequently, ASL becomes their principal language and English their secondary. In some sense, these students are English Language Learners (ELLs) and some of the approaches to teaching these students is similar to approaches to teaching ELLs [1]. According to the National Center for Education Statistics (NCES Table 204.27) [2], almost 5 million (about 10%) of students enrolled in public schools in the United States (US) in 2016 are ELLs.

The approach to teaching Deaf students at state residential schools for the Deaf and many Deaf programs in mainstream schools is bilingual, with instruction in ASL and with all other resources (textbooks, workbooks, videos, websites, computer apps, exams) in English. Unlike the situation for typical hearing ELLs, who are on a path to learning spoken and written English and being fully bilingual, Deaf students may never master spoken English and may be weak in written English. Nonetheless, the vast majority of these students are intellectually capable of mastering computer science and should be given the opportunity to do so.

In this experience report, we describe a professional development workshop for 14 teachers of Deaf students at state residential schools for the Deaf and Deaf programs at mainstream schools. All these teachers, including 8 who are Deaf themselves, use a bilingual approach to Deaf education. The workshop was conducted in ASL with the help of ASL interpreters for hearing attendees who did not know ASL. The purpose of the workshop was to prepare these teachers to teach the Code.org curriculum for AP Computer Science Principles (CSP) in their respective schools. A major outcome of the workshop is a set of lessons learned, from both the workshop staff and teachers, for the various stakeholders in Deaf education and PreK-12 computer science education.

The following sections include related work on Deaf students and bilingual education. From there, we discuss workshop staff and teacher participants, the workshop program, highlights, and lessons learned for a variety of stakeholders. These stakeholders include teachers in Deaf classrooms, teachers with one Deaf student in their classes, and AP CSP content providers.

II. Related Work

There is very little literature on the preparation of teachers to teach computer science to students with disabilities at the PreK-12 level. At SIGCSE 2019, Stefik *et al.* reported on a professional development workshop for teachers of blind and visually impaired students [3]. Following that workshop, the first two authors created a professional development workshop for teachers at schools for the Deaf and Deaf programs at mainstream schools. In their prior work, there was much concern about replacement of visual content of computer science curricula with non-visual content. In this report, the concerns are quite different and are more about integrating computer science into the bilingual approach to Deaf education.

A. Deaf Students

Deaf students typically fall under the Individuals with Disabilities Education Act (IDEA) (First authorized in 1975 as the Education of All Handicapped Children Act) or Section 504

This work by AccessCSForAll was supported by the National Science Foundation under Grants #CNS-1738252 and #CNS-1738259.

[1]The capital "D" in Deaf is used to indicate that the students referred to in this paper use American Sign Language as their primary language and are considered part of the Deaf Community.

978-1-7281-7172-2/20$31.00 ©2020 IEEE

of the Rehabilitation Act of 1973. According to the National Center for Education Statistics (NCES Table 204.30) [4], in 2017-18, there were about 7 million PreK-12 students with disabilities served under IDEA. This is about 13.7% of all PreK-12 students. As many as 1.5 million more students may be covered under Section 504. Within the IDEA group, about 75,000 students were identified as having a "hearing impairment" which can range from total deafness to moderate hearing loss.

According to NCES (Table 204.60) [5], in 2017, only about 10% of these "hearing impaired" students attended schools for the Deaf of any kind, and 88.3% attended mainstream schools, with some in Deaf programs. The remaining students were in other educational settings, such as home-schooled. In the US, because education is run at the state level and below, there is high variability in the educational opportunities PreK-12 students receive. There is even more variation among Deaf students because of the varying educational philosophies (oral versus bilingual) and the willingness of parents to send their children away from home to residential schools. The oral philosophy of Deaf education focuses on hearing enhancement technology, speech-reading, and speech, and discourages the use of sign language.

B. Bilingual Education

Bilingual Deaf education has a long history going back to the founding of schools for the Deaf in the early 1800s. Strassman *et al.* stated: "for those interventions that have been studied, d/Deaf students' achievement is positively affected by educational practices similar to those recommended for ELLs [1]." Nine sound principles for ELL pedagogy are delineated by Li [6]. Although not all these principles apply to Deaf students, several of them appear to be what our teachers report using with their students.

It may be tempting to think of using a student's first language as a "crutch" in an ELL pedagogy. The work of Sparks *et al.* suggest strength in an ELL's first language leads to eventual strength in the ELL's acquired language [7]. Rather than a crutch, the first language is a launching point. A similar result was found in Deaf students by Scott and Hoffmeister that ASL is a launching point to learn concepts in English [8].

There is a series of two papers authored by A.G.S. Raj *et al.* related to computer science taught in India using Tamil-English bilingual college students [9], [10]. The papers reported on two studies where students were taught computer science lessons in two conditions, English alone and in a combination of Tamil and English. In both studies, the amount of learning in the two conditions were about the same. However, in the case of Tamil + English approach, students were more engaged in the classroom and felt better about their learning. In our workshop, teachers did not report on an all-English approach to teaching computer science. All our teachers reported using a bilingual approach in their classes.

III. Workshop Staff and Teachers

The workshop staff consisted of the four authors of this report. The first two are experienced researchers and practitioners in K-12 computer science education, the third, who is Deaf, is an experienced AP CSP teacher at a school for the Deaf, and the fourth is an employee of Code.org, an endorsed content provider for AP CSP with expertise in accessibility. The first and third authors are bilingual in ASL and English.

We recruited teachers who use the bilingual approach from residential schools for the Deaf, day schools for the Deaf, and Deaf programs in mainstream schools. There were 14 teacher participants in the workshop from 11 different states. All 14 were ASL fluent including 8 who were Deaf. All but three were active teachers in state schools for the Deaf. One was a statewide coordinator for Deaf students in their state; one was a teacher in a Deaf program in a mainstream high school; and one was a middle school teacher in a private K-8 school for the Deaf. Only one was experienced in teaching computer science and another five had experience teaching technology subjects such as coding, robotics, web design, and computer aided design (CAD). The remaining 8 teachers did not have CS teaching experience. Table 1 describes participants in more detail.

IV. Workshop Program

The workshop followed the general approach of the Code.org professional development workshops for teachers. The College Board recognizes Code.org as an endorsed curriculum provider for AP Computer Science Principles. We focused on the Code.org AP Computer Science Principles (2019-2020) curriculum.

Code.org professional development makes heavy use of teacher modeling in a process they refer to as Teacher-Learner-Observer (TLO), in which a pair of teachers prepare a lesson from the Code.org AP CSP curriculum and teach it to the other teachers in the workshop, who act as students [11]. Workshop facilitators and staff act as observers who engage workshop participants in reflection after each lesson. In this case, the lesson was given in ASL with any supplementary materials in English, just like what would happen in a bilingual school for the Deaf.

In total, eight lessons were taught, a model lesson by the staff member who taught at a residential school for the Deaf, and seven more by the teachers. All the teachers created highly visual slides as part of their lessons. All teachers prepared by using Code.org lesson plans, resources, and tools to implement an experiential activity. These activities led to discussions of the computer science concepts used in the activity and introductions to the technical language used in practice. Teachers were asked to be creative in making their lessons as accessible as possible for their students. They were allowed to modify a Code.org lesson to suit the students' needs. Several hours were allotted during the workshop to prepare their lessons. Each lesson was 40 minutes followed by 20 minutes of reflective discussion between the workshop participants who taught the lesson and those acting as students

TABLE I
SUMMARY OF ATTENDEES OF THE WORKSHOP

Job Description	School Setting	CS Teaching Experience
Technology Teacher	State School for the Deaf	Coding, CAD, Robotics
Computer Science Teacher	State School for the Deaf	Computer Science
Math Teacher	State School for the Deaf	none
Math Teacher	State School for the Deaf	none
Math Teacher	State School for the Deaf	none
English Teacher	State School for the Deaf	none
Statewide Coordinator for Deaf students	Multiple Mainstream Schools	none
Technology Teacher	State School for the Deaf	Coding, Web design
English and History Teacher	Mainstream School with Deaf program	Only as a student teacher
Language Development Teacher	State School for the Deaf	none
Math Teacher	State School for the Deaf	none
Math and English	PreK-8 Private School for the Deaf	none
Substitute Teacher	State School for the Deaf	Coding
Biology and Earth Science Teacher	State School for the Deaf	Robotics

separately first, then all together. The 8 lessons covered in the workshop using the TLO model covered algorithms, number systems, network protocols, data compression, and encryption.

In addition to the TLO sessions there were additional sessions most of which involved the teachers in activities: unplugged, using tools, programming, exam preparation, a panel of deaf students, and a discussion of teachers' future plans.

V. WORKSHOP HIGHLIGHTS

The workshop was highly accessible to all teachers and staff. Two certified interpreters were available throughout the workshop to translate from ASL to English and *vice versa*. All the TLO sessions were enthusiastically presented by the teacher pairs, while the remaining teachers kept to their student roles. Almost all the teachers were unfamiliar with the content of their lessons, but were enthusiastic to learn new material and tools. Some teachers worked into the night to prepare their lessons. In several cases, the teachers consulted with the workshop staff to make sure they understood a new concept or tool.

Because one of the staff was an experienced AP CSP teacher of Deaf students, they were able to answer many questions that came up in the sessions about the Explore and Create tasks. They also had experience modifying the Code.org AP CSP curriculum to make it more accessible, which helped ground the discussion about how to make the material more accessible to Deaf students. Another staff member was an expert on programming languages.

The panel of three Deaf students had very interesting backgrounds. All three, two male and one female, were the only Deaf person in their families. Two of the three had very little exposure to Deaf people until adulthood. Now, both are learning ASL and integrating more into the Deaf community. The third had a Deaf friend in the neighborhood growing up so had more exposure to Deaf people and was fluent in ASL. The students were from three different universities, but all participating in the same summer research program studying various problems in human-computer interaction (HCI) at the site of the workshop. One, a PhD student, was a mentor in the program and the other two, both seniors, were participants. The teachers asked many questions of the students trying to understand their motivations for entering the computing field. All three were engaged in technology from a relatively early age, indicating that early exposure and success in technology may be an important factor in their choice of field of study.

On the final day of the workshop, the teachers were asked what their plans were and this varied. Several said they will now take on the role of the "computer science advocate" at their school. All but three of the teachers were from state schools for the Deaf that go from PreK-12. Several of them mentioned starting with the Code.org CS Discoveries curriculum for students in 6th-10th grade, then moving to the AP CSP curriculum. Several mentioned establishing a summer computer camp to increase interest in a future AP CSP offering. One teacher summed up the experience of many of the teachers in saying:

> I was reluctant. But really happy after all. I feel part of a family. I can now see the big picture. All this can be done without a lot of money. I will talk to the principal about starting a CS and robotics program at my school.

VI. LESSONS LEARNED

There were several lessons learned that impact teachers in Deaf classrooms, teachers with one Deaf student in their class, and AP CSP content providers.

A. Lessons for Teachers in Deaf Classrooms

The lessons learned for teachers in Deaf classrooms is based on the experiences of the one staff member who has actually taught AP Computer Science Principles for three years and the experiences of all 14 teachers who taught various subjects in all-Deaf bilingual classrooms. As mentioned earlier, in this setting, the language of instruction is ASL, while most of the academic resources are in English. Some basic advice is that students should sit in a circle so they can all see each other to keep lines of visual communication open when using ASL. Hands-on activities were reported as more effective than lecturing or using captioned video. This is quite compatible with a variety of preK-12 curricula from various providers.

Teachers should not feel compelled to follow exactly the Code.org AP CSP curriculum, especially when it applies to the amount of time suggested for each activity. Because reading speed and comprehension are often different with Deaf students, activities may take longer than the time allotted. Furthermore, preparation time can take longer in order to make modifications to the curriculum for Deaf students. Examples of modifications include creating a more visual slide deck for the lesson and creating meaningful warm-up activities for the Deaf students in preparing for the lesson (e.g., describing a virtual game of battleship to introduce all students to the rules before a lesson).

Although captions on videos are important for access, they are often hard to follow because students need to split attention between captions and content. It is easy to miss some captions because the content has captured a student's attention, or *vice versa*. Videos in ASL may be beneficial.

ASL does not have a standards committee, nor do most human languages. Thus, there are no standard signs used for many concepts, including computing concepts. Indeed, there are even regional signs for the word "computer" used around the United States. This is not necessarily a problem because all the regional signs for "computer" have one translation in English. There was some debate among our teachers as to whether a computer concept should even have sign or, alternatively, just be finger spelled.

B. Lessons for Teachers with one Deaf Student

Most Deaf students are in mainstream schools. Even if they are in a mainstream school with a Deaf program, it is highly likely that they would be the only Deaf student in the computer science class they take. This student will typically have an Individualized Education Program (IEP) or Section 504 Plan that calls for either a sign language interpreter or real-time captionist. The sign language interpreter translates what is being spoken in English to ASL or something called Pidgin Signed English (PSE). PSE is close to ASL except signs are done in English order. A real-time captionist is a highly trained person who translates spoken English to text, word for word. It is important to note that sign language interpretation and real-time captioning are not perfect, so there will be misunderstandings.

There are some Deaf students who use hearing technology (hearing aid or cochlear implant) to hear what is being said. These "assistive listening devices" may be aided by speech reading (often called lip reading). In any case, sight-lines are important to these students. Deaf students should be close to a teacher (to speech read or hear the teacher), interpreter (to see signs), or caption screen (to see text of what is spoken). The teacher should be close to the slides, interpreter, and captionist, so as to minimize the time to switch attention.

C. Lessons for Content Providers

Previous advice to content providers is first do an accessibility audit of your AP CSP curriculum [3]. Teachers in the workshop emphasized that all videos in the curriculum should be captioned and that English explanations should be straightforward and easy to follow. Visual and non-textual ways of explaining content can be very useful. When possible, it is recommended to provide visual and illustrative content in formats which can be incorporated into teachers' classrooms. They also mentioned that it is helpful to think of Deaf students as English Language Learners (ELLs) who number in the millions.

Providing videos in ASL directly could meet the visual communication needs for Deaf students and assist teachers in introducing concepts to students. However, we see this as more an ideal, as there are significant challenges. First, considerable work would be needed to address the concerns raised by the 14 teachers in this workshop regarding regional consistency of signs and establishing a consensus around finger spelling or creating signs for technical concepts. Second, given the cost in hiring interpreters, teams that do not know ASL could require significantly more resources to create the videos, as opposed to captions, which can be added easily and cheaply.

REFERENCES

[1] B. K. Strassman, K. Marashian, and Z. Memon, "Teaching academic language to d/deaf students: Does research offer evidence for practice?" *Gallaudet University Press*, vol. 163, no. 5, pp. 501–533, 2019.

[2] "NCES Table 204.27," Retrieved from: https://nces.ed.gov/programs/digest/d18/tables/dt18_204.27.asp.

[3] A. Stefik, R. E. Ladner, W. Allee, and S. Mealin, "Computer Science Principles for Teachers of Blind and Visually Impaired Students," in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '19. New York, NY, USA: ACM, 2019, pp. 766–772. [Online]. Available: http://doi.acm.org/10.1145/3287324.3287453

[4] "NCES Table 204.30," Retrieved from: https://nces.ed.gov/programs/digest/d18/tables/dt18_204.30.asp.

[5] "NCES Table 204.60," Retrieved from: https://nces.ed.gov/programs/digest/d18/tables/dt18_204.60.asp.

[6] J. Li, "Principles of effective english language learner pedagogy," New York, NY: College Board. Retrieved from Educational Resources Information Center website: https://files.eric.ed.gov/fulltext/ED562799.pdf, 2012.

[7] R. L. Sparks, J. Patton, L. Ganschow, and N. Humbach, "Do L1 Reading Achievement and L1 Print Exposure Contribute to the Prediction of L2 Proficiency?" *Language Learning*, vol. 62, no. 2, pp. 473–505, 2012. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9922.2012.00694.x

[8] J. A. Scott and R. J. Hoffmeister, "American Sign Language and Academic English: Factors Influencing the Reading of Bilingual Secondary School Deaf and Hard of Hearing Students," *The Journal of Deaf Studies and Deaf Education*, vol. 22, no. 1, pp. 59–71, 12 2016. [Online]. Available: https://doi.org/10.1093/deafed/enw065

[9] A. G. Soosai Raj, K. Ketsuriyonk, J. M. Patel, and R. Halverson, "Does Native Language Play a Role in Learning a Programming Language?" in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '18. New York, NY, USA: ACM, 2018, pp. 417–422. [Online]. Available: http://doi.acm.org/10.1145/3159450.3159531

[10] A. G. Soosai Raj, E. Zhang, S. Mukherjee, J. Williams, R. Halverson, and J. M. Patel, "Effect of Native Language on Student Learning and Classroom Interaction in an Operating Systems Course," in *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE '19. New York, NY, USA: ACM, 2019, pp. 499–505. [Online]. Available: http://doi.acm.org/10.1145/3304221.3319787

[11] "Code.org AP CSP Curriculum," Retrieved from: https://studio.code.org/courses/csp-2019.