

# *AccessComputing* Promotes Teaching Accessibility

By Amy J. Ko and Richard E. Ladner, *University of Washington*

**A**ccessComputing is a National Science Foundation (NSF) funded alliance, begun in 2006, whose goal is broadening participation in computing. Since then, it has been focused on this single goal and its approach to achieving this has focused on three high level strategies.

- Direct intervention: work directly with students with disabilities to encourage them to go into computing fields and to help them build their capacity to do so.
- Institutional change: work with computing departments and organizations to help make them more welcoming and accessible to people with disabilities.
- Dissemination: inform the computing community about the importance of including people with disabilities and providing them resources to do so.

After ten years, *AccessComputing* is extending its institutional change strategy by adding an important new strategy—curricular change. Rather than just working with individual

departments, the alliance will now help computer science (CS) instructors at all levels across the nation to incorporate accessibility and universal design directly in their coursework. In this article, we discuss some of our latest efforts to affect this change and how teachers can help achieve it.

## WHO ARE PEOPLE WITH DISABILITIES

According to the World Health Organization there are about one billion people in the world who have a significant disability [21]. This is about one in seven people who were either born with a disability or acquired one during their lifetime through disease, accidents, wars, or as a result of the natural aging process. An impairment that causes a disability may be stable throughout life such as one caused by cerebral palsy, progressively become more severe such as with retinitis pigmentosa, or vary from day to day or even hour to hour such as in bipolar conditions. The WHO considers disability to be far more than just a health issue, but also a social issue:

## AccessComputing Promotes Teaching Accessibility

*[Disability] is a complex phenomenon, reflecting the interaction between features of a person's body and features of the society in which he or she lives. Overcoming the difficulties faced by people with disabilities requires interventions to remove environmental and social barriers [21].*

The scale of disability is tremendous. The National Center for Education Statistics (NCES) collects data on students with disabilities at all levels in the United States, finding that about 13% of all K-12 students have Individualized Education Plans (IEPs) under the Individuals with Disabilities Education Act (IDEA) [13]. By our estimates, based on the percentages in Washington State, another 2% of all students have Section 504 Plans [18], which provides for accommodations for a student with a disability without changing learning objectives. About 11% of U.S. college students have a disability and about 5% of graduate students have a disability [12]. All totaled there are about 10.2 million students in the United States with disabilities at all levels of the education pipeline. Naturally, disability cuts across gender, race, ethnicity, and socioeconomic status, making it a barrier to any effort to broaden participation in computing.

Because of the broad spectrum of disabilities, there is no simple solution to providing everyone access to a computing education. Indeed, some students with developmental disabilities may never learn to read or write, much less learn computer science. One proactive approach to including students with disabilities is to use the principles of universal design of instruction [6]. Universal design does not cover all students, but specific accommodations can make computer science accessible. For example, students who are blind or low vision can use screen readers to have access to the computer screen. If applications and webpages are designed to be accessible then these student can participate fully. There are many other accommodations that can make computer science accessible to students with other disabilities [16]. As we shall discuss, however, it takes more than accommodations and technology to make computing education truly accessible to all.

### ACCESSCOMPUTING EFFORTS

*AccessComputing* has 45 formal partners including academic institutions and computing organizations. Partners share the goal of *AccessComputing* and to varying degrees act to achieve the goal. In addition to partners, *AccessComputing* has worked with over 175 collaborators who have led at least one effort to help achieve the goal. Partners and collaborators have hosted student interns, hosted events focusing on disability, or led other activities that promote the inclusion of people with disabilities in computing fields.

The *AccessComputing* team facilitates direct interventions for students, where over 700 students with disabilities have participated over the past ten years. More than 400 students have been placed in industry or academic internships. Many have been funded to attend conferences in their sub-disciplines and professional development conferences like the Tapia Celebration of Diversity

in Computing [14] and the Grace Hopper Celebration of Women in Computing [9]. More than 80 deaf and hard of hearing students participated in the Summer Academy for Advancing Deaf and Hard of Hearing in Computing between 2007 and 2013. This rigorous nine-week program offered students a computer pro-

## As part of *AccessComputing's* institutional change strategy the alliance is promoting the teaching of accessibility and disability related topics.

gramming course for academic credit and an animation course for certificate credit. Students met deaf or hard of hearing mentors in computing fields and visited local high tech companies.

Institutional change is accomplished by working with *AccessComputing* partners and collaborators to help their computing departments and organizations become more welcoming and accessible to people with disabilities. Examples of activities include holding birds of a feather sessions focused on disability issues at the SIGCSE, CHI, and Tapia conferences, helping design and fund capacity building institutes at universities, and working with SIGCHI to make their annual conference more accessible.

The backbone of dissemination efforts is the *AccessComputing* Knowledge Base [1] that contains over 400 questions and answers, case studies, and evidence-based practices that can help inform the broad computing community about disability related issues. In addition, *AccessComputing* staff members often give presentations at conferences and write articles like this one to help keep people aware of issues related to the inclusion of people with disabilities in computing fields [5,6,7,10].

### TEACHING ACCESSIBILITY

As part of *AccessComputing's* institutional change strategy the alliance is promoting the teaching of accessibility and disability related topics. There are a number of reasons for this promotion. First, the more students know about accessibility and disability the more likely that they will consider more diverse users when they get into the computing workforce. There is evidence that more than half of software written today is for a user interface (UI) [11]. Anyone writing UI software should be considerate of the one billion people in the world who have disabilities and might use their UI. And anyone designing user interfaces should feel empowered to advocate for people with disabilities in their designs and product roadmaps.

Second, employers are increasingly expecting computer and information science students to know about accessibility. The Teach Access Initiative [17] has identified a number of major companies that have explicitly added knowledge of accessibility to job descriptions. For example, in a recent Facebook job description for a Software Engineer, Mobile, Android UI is the

Finally, from an intellectual standpoint, there are principled reasons for students to learn about accessibility and disability. Designing for blind users can pose significant challenges for learning and creativity (e.g., is it possible to design a game that is accessible to blind users?). Accessibility also presents compelling and meaningful illustrations of modern techniques of computer science (e.g., using machine vision to detect and read the text to blind people). Accessibility can also be a powerful way to teach students design empathy, taking them out of their comfort zones and empowering them to see the world from the perspectives of those for whom they will soon engineer software.

We advocate three basic strategies for including accessibility and disability in courses: change a lecture, add a lecture, and add a new course.

[illegible]

Adding a whole new lecture to a course is more effort, but a natural fit for fields such as human-computer interaction (HCI), robotics, computer vision (CV), and natural language processing (NLP), which all play a role in creating accessible technologies. For example, Ladner has given a lecture in an undergraduate CV class on the computer vision problems that are needed

# Visual to Tactile Image

The diagram illustrates the process of converting a visual graph into a tactile format. On the left, a visual coordinate plane shows a shaded triangle with vertices at  $(0,0)$ ,  $(5,0)$ , and  $(2,4)$ . The lines are labeled with equations:  $y = -2x + 4$ ,  $y = x$ , and  $x = 5$ . An arrow points to the right, where a tactile version of the same graph is shown. The tactile version uses Braille labels for the equations and coordinates, and the shaded region is represented by a textured area.

Entirely new courses can be a powerful way to engage students in creating accessible technologies. For example, many computer science and engineering programs have capstone design courses that help students bring what they have learned in various courses together in a project. In the “Accessibility Capstone” students identify a specific accessibility issue, and then try to design and build a solution [3]. Each group has a consultant who either has a disability or is someone who works closely people with disabilities, say as an occupational therapist. There are similar accessibility capstone courses at the University of Michigan called Engineering for Grace [8] and at MIT called Principles and Practice of Assistive Technology [15]. From our experience, students in these courses stay highly engaged because the problem they are trying to solve with technology has the potential to help someone in a tangible way.

Our own efforts to teach accessibility at our university can help illustrate how simple some of these strategies can be to implement, even without existing instructor expertise on accessibility. We set out with a simple goal of adding a web accessibility lecture to biggest web development course taught on campus.

acm Inroads ■ inroads.acm.org 67

*AccessComputing* Promotes Teaching Accessibility

interactions with information technologies), recently hired three instructors to teach its required undergraduate web development course. All three instructors had drafted a new curriculum that included a frontend web development course, a backend web development course, a lower-division engineering skills course, and a set of advanced web technology electives. We reached out to these three instructors near the end of this curriculum overhaul and asked, “What are you teaching about accessibility?”

Their answers were enlightening. First, they expressed a sincere desire to teach it, but also a lack of expertise. If they were to teach anything, they wanted some instruction of their own to feel confident with the material. Second, they had little insight into *where* accessibility might fit into the new curriculum. They had plenty of ideas, but not a lot of certainty about them. Finally, their collective, team-based response was critical to their willingness to incorporate accessibility—the more senior instructor suggested that one could pilot something in the summer and then teach the others how it went. Their existing collaborations and well-established teaching team ensured that no one felt like they were taking on the challenge alone.

Since they were open to including accessibility in their courses, we arranged a two-hour deep dive with the three instructors to cover web accessibility basics [4,19] and allow time for brainstorming. One of the *AccessComputing* staff did the heavy lifting, reusing a slide deck that he had shared with many other institutions, and then the first author of this article, Ko, facilitated the brainstorming. Once the instructors understood the scope of concepts they might teach, they generated ideas (e.g., use accessibility to make learning markup more interesting, use screen readers to explain ideas in software testing, use persona of people with disabilities to bring the human perspective to the material). With a little time and a little focus, these three creative instructors had no problem selecting an idea and deciding who would pilot it.

The last piece of our efforts was to ensure the pilot actually happened. Ko set up some reminders for himself: 1) check in with the instructor who was piloting just before course prep time; 2) point them to existing materials and resources in the *AccessComputing* Knowledge Base; and 3) check in after the unit had been piloted. With just a few hours of email, meeting, and brief reminders, we managed to incorporate accessibility as a core subject in a required class that reaches over 270 undergraduates a year, most of whom take this knowledge directly into industry as web developers, software engineers, and designers.

Each institution is going to require its own tactics. The approach we used worked because of good timing, a culture of teaching excellence, and a commitment to hiring a cohort of dedicated instructors. Every context is no doubt different and requires a bit of creativity to navigate; *AccessComputing* is here to help you find a way.

## CONCLUSION

Teaching about accessibility and disability is a powerful vehicle for engaging students in computer science. Furthermore,

knowledge of accessibility related technologies and practices will help make computer and information science students more employable in the modern job market. If you teach or know about computer science curriculum that includes accessibility or disability related topics, please let the authors know. *AccessComputing* is developing a resource devoted to sharing accessibility and disability related curricula [2]. ♦

### Acknowledgement

This material is based in part upon work supported by the National Science Foundation under Grant Number I539179. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

### References

1. *AccessComputing* Knowledge Base; <http://www.washington.edu/accesscomputing/search-accesscomputing-knowledge-base>, Accessed 2016 April 20.
2. *AccessComputing* Resource for Curricula; <http://www.washington.edu/accesscomputing/resources/curricula>, Accessed 2016 May 25.
3. Accessibility Capstone; <http://courses.cs.washington.edu/courses/cse481h/>, Accessed 12 July 2016.
4. ARIA Standards; <https://www.w3.org/WAI/intro/aria>, Accessed 2016 April 25.
5. Bellman, S., Burgstahler, S., & Ladner, R. “Work-based learning experiences help students with disabilities transition to careers: A case study of University of Washington projects.” *Work*, 48,3 (2014), 399-405.
6. Burgstahler, S. “Universal design: Implications for computing education.” *ACM Transactions on Computing Education (TOCE)*, 11,3 (2011), 19.
7. Burgstahler, S., Ladner, R. E., & Bellman, S. “Strategies for increasing the participation in computing of students with disabilities.” *ACM Inroads*, 3,4 (2012), 42-48.
8. Engineering for Grace, University of Michigan; [http://eecs.umich.edu/cse/faculty/chesney/articles\\_media.html](http://eecs.umich.edu/cse/faculty/chesney/articles_media.html). Accessed 2016 April 20.
9. Grace Hopper Celebration of Women in Computing; <http://ghc.anitaborg.org/>, Accessed 2016 May 25.
10. Ladner, R. E., & Burgstahler, S. “Increasing the participation of individuals with disabilities in computing.” *Communications of the ACM*, 58, 12 (2015), 33-36.
11. Myers, B. A., & Rosson, M. B. “Survey on user interface programming.” In *Proceedings of the SIGCHI conference on Human factors in computing systems* (ACM, June 1992), 195-202.
12. NCES Table 311.10; [http://nces.ed.gov/programs/digest/d13/tables/dt13\\_311.10.asp?referrer=report](http://nces.ed.gov/programs/digest/d13/tables/dt13_311.10.asp?referrer=report). Accessed 2016 April 20.
13. NCES Table 204.30; [https://nces.ed.gov/programs/digest/d13/tables/dt13\\_204.30.asp](https://nces.ed.gov/programs/digest/d13/tables/dt13_204.30.asp). Accessed 2016 April 20.
14. Richard Tapia Celebration of Diversity in Computing; <http://tapiacconference.org/>. Accessed 2016 May 25.
15. Principles and Practice of Assistive Technology, MIT; <http://courses.csail.mit.edu/PPAT/fall2014/index.html>. Accessed 2016 April 20.
16. Tagayuna, A., et al. “A two-year comparison of support provision for persons with disabilities in postsecondary education.” *Journal of Vocational Rehabilitation*, 22, 1 (2005), 13-21.
17. Teach Access website; <http://teachaccess.org/>. Accessed 2016 April 20.
18. Washington State Report Card, Office of Superintendent of Public Instruction; <http://reportcard.ospi.k12.wa.us/summary.aspx?groupLevel=District&schoolId=1&reportLevel=State&year=2014-15>. Accessed 2016 April 20.
19. Web Content Accessibility Guidelines 2.0; <https://www.w3.org/TR/WCAG20/>. Accessed 2016 April 25.
20. Wobbrock, J. O. et al. “Ability-based design: Concept, principles and examples.” *ACM Transactions on Accessible Computing (TACCESS)*, 3, 3 (2011), 9.
21. World Health Organization Disabilities webpage; <http://www.who.int/topics/disabilities/en/>. Accessed 2016 April 20.

### Amey J. Ko

The Information School  
University of Washington  
Box 352840, Seattle, WA 98195  
[ajko@uw.edu](mailto:ajko@uw.edu)

### Richard E. Ladner

Computer Science & Engineering  
University of Washington  
Box 352350, Seattle, WA 98195  
[ladner@cs.washington.edu](mailto:ladner@cs.washington.edu)