



Increasing the Participation of Students with Disabilities in Exploring Computer Science and Computer Science Principles courses

Proceedings of an *AccessCS10K*
Capacity Building Institute

January 23 – 24, 2015
Baltimore, Maryland



Supported by
National Science Foundation



Increasing the Participation of Students with Disabilities in Exploring Computer Science and Computer Science Principles Courses

Proceedings of the January 2015 *AccessCS10K* Capacity Building Institute (CBI)

This publication shares the proceedings of a CBI entitled Increasing the Participation of Students with Disabilities in Exploring Computer Science and Computer Science Principles Courses. The content may be useful for people who

- participated in the CBI
- teach Exploring Computer Science (ECS) or Computer Science Principles (CSP)
- train teachers of ECS or CSP courses
- seek to increase their understanding of issues surrounding the participation of students in computing studies and careers
- would like to access resources to help make their courses, services, and activities more welcoming and accessible to students with disabilities
- have promising practices to share with others

About **AccessCS10K**

AccessCS10K: Including Students with Disabilities in Computing Education for the Twenty-First Century (*AccessCS10K*) works to increase the successful participation of students with disabilities in high school ECS and CSP courses. Central to this work is partnerships with other projects funded by the Computing Education for the 21st Century program of the National Science Foundation (NSF) Directorate for Computer and Information Science and Engineering (CISE) that train computer science teachers and develop curricula for ECS and CSP. *AccessCS10K* is led by the Department of Computer Science and Engineering and the DO-IT (Disabilities, Opportunities, Internetworking, and Technology) Center at the University of Washington (UW) and the Department of Computer Science at the University of Nevada, Las Vegas. It is funded by CISE (Grant #CNS-1440843).

Both ECS and CSP courses are designed to introduce the big ideas in computer science and to inspire students from diverse backgrounds to consider computational competence as part of their career plans. *AccessCS10K* focuses on the inclusion of students with disabilities in these courses through two objectives:

- building the capacity of ECS and CSP teachers to serve those students through professional development for trainers who provide professional development to teachers, curricular units, online tutorials, virtual communities of practice for teachers, and real-time, individual teacher support; and
- creating accessible materials, both tools (including iterative refinement and deployment of the Quorum language) and curricular units, that ECS and CSP teachers and students can use in their classrooms.

The ultimate goal of *AccessCS10K* is to include the participation of people with disabilities in computing academic studies and careers and to enhance those fields with their unique perspective and expertise.



Contents

About the CBI	7
CBI Agenda.....	9
Presentation Summaries.....	11
Working Group Discussion Summaries	15
CBI Participants.....	19
Communities of Practice	21
Resources	23
Acknowledgments	25



About the CBI

Increasing the Participation of Students with Disabilities in Exploring Computer Science and Computer Science Principles Courses CBI, sponsored by *AccessCS10K*, was held in Baltimore, MD on January 23 – 24, 2015. Its purpose was to encourage efforts to make ECS and CSP courses more accessible and welcoming to students with disabilities. Taking place immediately following the Computing Education for the 21st Century principle investigator meeting, attendees included postsecondary faculty, individuals who provide professional development for ECS and CSP teachers, disability services professionals, and individuals with disabilities.

As is typical of a CBI

- All participants contributed to its success.
- Speakers participated in group discussions.
- Experts in all topic areas were in the audience.
- Participants gave presentations and participated in large and small group discussions.
- Some predetermined professional development was presented, as well as new content was delivered as the meeting unfolded.
- Participant interests were expressed, and expertise was made known.

The CBI provided a forum for discussing recruitment and access challenges, sharing successful practices, developing collaborations, and identifying systemic change initiatives for increasing the participation of students with disabilities in ECS and CSP courses.

Topics discussed included

- universal design of instruction (UDI) and academic accommodations;
- profiles of computing professionals with disabilities;
- accessibility of programming environments;
- assistive technology and accessible IT design; and
- best practices for making courses welcoming and accessible to students with disabilities.

The agenda for the CBI, summaries of the presentations, and working group discussions are provided on the following pages.



CBI Agenda

Friday, January 23

1:30 – 2:45 pm

Welcome

Introductions, housekeeping, *AccessCS10K* goals, objectives, activities, and resources

Richard Ladner, Andreas Stefik, Sheryl Burgstahler

2:45 – 3:15 pm

Careers in Computer Science and Disability

Richard Ladner

3:30 – 4:30 pm

Accommodations vs. Universal Design in K-12 Education

Sheryl Burgstahler

4:30 – 5:15 pm

Accessibility and Technology

Richard Ladner and Andreas Stefik

5:30 – 6:15 pm

Intro to Quorum/Hour of Code activity

Andreas Stefik

6:30 – 7:45 pm

Dinner and Working Group Discussion

What aspects of the Exploring Computer Science (ECS) or Computer Science Principles (CSP) curriculum might present accessibility challenges?

Sheryl Burgstahler

Evaluation of Day 1

Saturday, January 24

8 – 9 am

Breakfast Discussion

What resources or tools would be helpful to prepare for students with disabilities in their courses?

9 – 10 am

Report Back on Dinner and Breakfast Discussions and Further Conversation

Richard Ladner, Andreas Stefik, and Sheryl Burgstahler

10:15 – 11 am

An Accessible Programming Lesson

Andreas Stefik

11 – 11:45 am

Large Group Discussion

How can you prepare teachers teaching ECS or CSP to be welcoming to students with disabilities?

11:45 am

***AccessCS10K* Resources**

12 – 1 pm

Lunch and Discussion

What resources or tools would be helpful to you or to teachers in order to be prepared for students with disabilities in their courses?

CBI Evaluation



Presentation Summaries

Careers in Computer Science and Disability

Presenter: Richard Ladner

People with a wide variety of disabilities can do almost anything in any computing field. Their expertise and perspective drive innovation, and many are interested in accessibility research and development. According to the United States Department of Labor, there is a large gap between the number of computing jobs available and the number of students in computer science who can fill those jobs. People with disabilities, currently underrepresented in computing fields, can help meet this need.

The United Nations Convention on the Rights of Persons with Disabilities Article 27 states, “Parties recognize the right of persons with disabilities to work, on an equal basis of others; this includes the right to the opportunity to gain a living by work freely chosen or accepted in a labour market and work environment that is open, inclusive and accessible to persons with disabilities.”

William A. Wulf stated in 2000, “I believe that engineering is a highly creative profession. Research tells us that creativity does not spring from nothing; it is grounded in our life experiences, and hence limited by those experiences. Lacking diversity on an engineering team, we limit the set of solutions that will be considered, and we may not find the best, the elegant solution.” Individuals with disabilities in computing fields contribute to such diversity. Their expertise and perspectives help drive innovation.

There are successful role models with disabilities in science and engineering fields, including inventor Thomas Edison who was deaf, physicist Stephen Hawking who has amyotrophic lateral sclerosis (ALS), “father of the internet” Vince Cerf who is hard of hearing, and acoustic modem inventor Robert Weitbrecht who was deaf. There are computing professionals in industry and academia, as well as successful students in computing fields, who have disabilities. These role models allow children to think, “I can be like that person;” for parents to recognize, “My child can have a future;” and for teachers to believe, “My student can be more than I imagined.”

Accommodations vs. Universal Design in K-12 Education

Presenter: Sheryl Burgstahler

Ability exists on a continuum, where all individuals are more or less able to see, hear, walk, read print, communicate verbally, tune out distractions, learn, or manage their health. In K-12 education in the United States, every child is ensured a free, appropriate education in as integrated of a setting as possible. However, in postsecondary education, students must meet whatever course or program requirements apply and are offered reasonable accommodations as needed.

Accommodations and universal design (UD) are two approaches to access for people with disabilities. Both approaches contribute to the success of students with disabilities in computing classes. Accommodations are a reactive process, providing access for a specific student and arise from a medical model of disability. Students might be provided with extra time on tests, books in alternate formats, or sign language interpreters.

In contrast, UD is a proactive process rooted in a social justice approach to disability and is beneficial to all students. UD is designing products and environments to be usable by all people, to the greatest extent possible, without the need for adaptation or specialized design. A UD approach can benefit people who face challenges related to socioeconomic status, race, culture, gender, age, language, or ability. Applying UD to information technology would include building in accessibility features and also ensuring compatibility with assistive technology. In other words, a universally designed website would have text alternatives for graphics, present context via text and visuals, include captions and transcripts for all video and audio content, ensure that all content and navigation can be reached with the keyboard alone, and spell out acronyms.

UD of instruction is an attitude that values diversity, equity, and inclusion. It can be implemented incrementally, focuses on benefits to all students, promotes good teaching practice, does not lower academic standards, and minimizes the need for accommodations. UD can be applied to all aspects of instruction, including class climate, interactions, physical environments and products, delivery methods, information resources and technology, feedback, and assessment. Examples include

- Arranging seating so that everyone has a clear line of sight.
- Avoiding stigmatizing a student by drawing undue attention to a difference.
- Using large, bold fonts with high contrast on uncluttered overhead displays and speak aloud all content.
- Providing multiple ways to gain and demonstrate knowledge.
- Avoiding unnecessary jargon; defining terms.
- Providing scaffolding tools (e.g., outlines).
- Providing materials in accessible formats.
- Providing corrective opportunities.
- Testing in the same manner in which you teach.
- Minimizing time constraints as appropriate.

Educators who effectively apply UD and accommodations level the playing field for students with disabilities and make instruction welcoming to, accessible to, and usable by all students. They minimize, but do not eliminate, the need for accommodations.

Accessibility and Technology

Presenter: Richard Ladner

Access can be provided by humans, either through something like sign language interpreters, or by technology through screen readers, dictation software, scanning keyboards, or other assistive technology. Accessibility is about equal access, usability, and being welcoming. The proactive approach to accessibility is making a product as usable as possible for as many people with varying abilities, while accommodations for an individual are reactive.

Blind or low vision students might use a range of technology to assist their learning. This includes

- built-in magnification such as Windows Magnifier or Apple Zoom
- external magnifiers such as closed circuit television
- braille translation, braille printers, braille displays, and note takers
- tactile graphics
- screen readers that provide speech output, such as Jaws, Window-Eyes, NVDA, and VoiceOver
- audio description of video presentations

Deaf or hard of hearing students might communicate in sign language, including American Sign Language, Pidgin Signed English, and Signed Exact English. They may use

- smart hearing aids
- cochlear implants and FM systems to amplify sound
- interpreters, real-time captioning, or note takers
- captions for videos, done by a professional company or using tools such as those in YouTube

Students with mobility-related disabilities might use

- speech input such as Dragon Naturally Speaking
- ergonomic keyboards, large keyboards, mini keyboards, virtual keyboards, or keyboard guides
- trackballs or mousekeys
- switches and eye gaze trackers

Students with learning disabilities might use

- Livescribe pen to take notes and transfer their notes on to a computer
- speech input
- text to speech software or screen readers
- visual organization or analysis tools
- equation editors

Introduction to Quorum / Hour of Code Activity

Presenter: Andreas Stefik

Quorum is a programming language originally designed for individuals who are blind or have low vision. It uses human factors data as an evidence-foundation and was inspired by two observations: 1) much of the computer science education literature relies on visual representations and 2) text-based programming in languages with traditional syntax (e.g., C++ or Java or Python's whitespace rules) are difficult to understand through audio.

Language designers rarely use scientific evidence involving humans in making decisions for their coding language, and even venues with a stated interest in the human impact of programming languages (e.g., the Psychology of Programming Interest Group) rarely collect replicable scientific evidence. In designing Quorum, we conducted randomized controlled trials on competing language designs. Via these trials, we've found that static typing improves productivity over dynamic typing and that some languages are just as difficult to use initially for novices as languages that are randomly generated (so-called placebo languages). Quorum's approach is different in that we use the data collected from such trials to make ongoing changes to the language, thereby making it easier to use.

Currently, about a thousand students are taught Quorum each year, and nearly half of schools for the blind in the US teach programming using it. Quorum runs on the Java Virtual Machine and can be used for a wide variety of applications (e.g., the web, music, speech, screen readers). We are currently adding support for Lego Robotics and a computer gaming engine, which will be released in June of 2015. Teaching materials for Quorum include an 18-week curriculum, online/offline tools support, a draft textbook, and real-time support via email or Facebook. Visit www.quorumlanguage.com to learn more and code.org/qrm for an online tutorial.



Working Group Discussion Summaries

Notes related to specific questions discussed by small groups are presented below.

What aspects of the Exploring Computer Science (ECS) or Computer Science Principles (CSP) curriculum might present accessibility challenges?

- Programming environments and languages, including Python, Alice, and Scratch, may not be accessible to some students (e.g., those who are blind or who have mobility-related disabilities).
- Teachers without a strong computer science background are often resistant to learning another programming language or teaching multiple languages in one class.
- Some assignments, such as creating animations, are fundamentally inaccessible to some students with disabilities.
- Pair programming during the Advanced Placement (AP) test can create logistical problems between someone with a disability and someone who doesn't.
- There is very little existing knowledge of how people with dyslexia, ADD, autism, or other invisible disabilities work in each programming language or utilize text-based versus drag-and-drop programming environments.
- Requesting accommodations for the AP test is cumbersome.
- Some teachers do not have the right resources—there may not be money in a school's budget for accessible software or hardware.
- Teachers and students aren't given the right tools or knowledge to surmount accessibility obstacles.
- There are very few computer science professionals with disabilities serving as role models.

What resources or tools would be helpful to prepare for students with disabilities in their courses?

- Create scaffolding (e.g., an outline of the program) to help students, especially those with learning disabilities.
- In pair programming or group activities, make sure there are multiple roles and ways to contribute, so students of all abilities can work together.
- Incorporate universal design when designing curricula and lessons.
- Work with special education teachers and other allies to determine what changes are needed in the classroom to make instruction more accessible in general or for a specific student.
- Discover what activities are both accessible and interesting, and look into alternatives, like a vocal interface for Scratch.
- Consider using unplugged activities (*csunplugged.org*), which can be very accessible and a great way to draw people in.
- Be aware that block languages can be great introductions to programming, but students who continue their computing education will move on to using text languages eventually.
- Look to professionals with visual or other disabilities to help students with disabilities learn to program.
- Identify inaccessible tools and ways to modify the tools to make them more accessible.
- Buy equipment and software that is more accessible, such as a talking debugger or a tool that beeps before all errors in a program.
- Contact the College Board to discuss accommodations and ask about the visual artifact requirement for the CSP assessment.

What resources or tools would be helpful to you or to teachers in order to be prepared for students with disabilities in their courses?

- Determine how to use a screen reader with text-based programming languages.
- Find out what environments are more accessible.
- Create materials for school administrators and counselors that show the potential for students with disabilities to take computer science classes.
- Demonstrate accommodations and how making a more accessible environment can allow students to work without modifications.
- Create resources so teachers know what tools/languages are the most accessible for people with different types of disabilities.
- Ensure that CSP students can create audio rather than visual artifacts to fulfill the visual artifact requirement of the AP exam.
- Provide a checklist for accessibility specifics in a number of activities, including unplugged activities and programming activities.
- Provide videos that model accessible lessons and accessibility issues in computer science.
- Create a website that compiles resources for high school teachers who work with students with disabilities, particularly a website that has resources for various types of disabilities.
- Provide grants for teachers to buy accessible or assistive technology.
- Provide workshops on accessibility through the Computer Science Teacher's Association.



CBI Participants

Stakeholder groups represented in the CBI included

- postsecondary faculty,
- individuals who provide professional development for ECS and CSP teachers,
- disability services professionals, and
- individuals with disabilities.

The following individuals participated in the CBI.

Owen Astrachan
Professor of the Practice of Computer Science
Duke University

Bradley Beth
Senior Program Coordinator
The University of Texas at Austin

Brianna Blaser
Program Coordinator/Counselor, DO-IT
University of Washington

Sheryl Burgstahler
Director, Accessible Technology Services
University of Washington

Gail Chapman
Director of Outreach, Exploring Computer
Science
University of California, Los Angeles

Lissa Clayborn
Acting Executive Director
Computer Science Teachers Association

Lyla Crawford
Program Coordinator, DO-IT
University of Washington

Zachary Dodds
Professor of Computer Science
Harvey Mudd College

Brian Harvey
Senior Lecturer, Emeritus
University of California, Berkeley

Cecily Heiner
Assistant Professor
Southern Utah University

Lynne Ipina
Associate Professor, Mathematics
University of Wyoming

Tom Keller
Senior Project Director
University of Washington

Richard Ladner
Professor
University of Washington

Irene Lee
Director, Learning Lab
Santa Fe Insititute

Joyce Malyn-Smith
Principle Investigator
Education Development Center

Daniela Marghitu
Faculty Coordinator
Auburn University

Ralph Morelli
Professor
Trinity College

Brook Osborne
High School Program Manager
University of Washington

Lori Pollock
Professor
University of Delaware

Maureen Psaila-Dombrowski
NM-CSforAll Program Manager
Santa Fe Institute

Jennifer Rosato
Computer Science Assistant Professor
College of St. Scholarstica

Derrick Smith
Associate Professor
University of Alabama in Huntsville

Nigamanth Sridhar
Associate Professor
Cleveland State University

Andreas Stefik
Assistant Professor
University of Nevada, Las Vegas

Chery Takkunen
Chair, Graduate Education Programs
The College of St. Scholastica

Chinma Uche
Math and Computer Science Teacher
Capitol Region Education Council

Sarah Wille
Researcher
Outlier Research and Evaluation/ University of Chicago

Aman Yadav
Associate Professor
Michigan State University



Communities of Practice

AccessCS10K engages stakeholders that include national leaders within Communities of Practice (CoPs). CoPs share perspectives and expertise and identify practices that promote the participation of people with disabilities in STEM fields. Those most related to *AccessCS10K* are described below.

Broadening Participation CoP

In this CoP, individuals who administer projects that serve to broaden participation in STEM fields

- discuss how to recruit participants with disabilities and accommodate them in their programs and activities and how to make their offerings more accessible overall
- recruit their participants with disabilities into disability-related e-mentoring, internships, academies, and workshops to complement their activities
- co-sponsor events and discuss potential new projects and share funding possibilities

Computing Faculty, Administrators, and Employers CoP

Computing professionals, faculty, and administrators as well as representatives from industry and professional organizations use this CoP to increase their knowledge about disabilities and make changes in computing departments that lead to more inclusive practices. Participants

- gain and share knowledge and help identify issues related to the underrepresentation of people with disabilities in computing fields
- help identify and field test Computing Department Accessibility Indicators to make computing departments more welcoming and accessible to students with disabilities
- help organizations make their websites accessible to visitors with disabilities, their conferences accessible to attendees with disabilities, and their conference programs inclusive of disability-related topics
- identify campus computing events to which students with disabilities might be invited
- discuss how to include accessibility topics in postsecondary computing curriculum

You and your colleagues can join CoPs by sending the following information to *doit@uw.edu*:

- CoP(s) you would like to join
- name
- position / title
- institution
- postal address
- email address

For information about other CoPs hosted by *AccessComputing*, consult the *AccessComputing* website at *www.uw.edu/accesscomputing/get-involved/educators-employers/communities-practice*.



Resources

The *AccessCS10K* website at www.uw.edu/accesscomputing/accesscs10k contains

- information about project goals, objectives, activities, and project partners
- evidence-based practices that support project goals and objectives
- resources for students with disabilities
- educational materials for teachers and administration

AccessCS10K maintains a searchable database of frequently asked questions, case studies, and promising practices related to how educators can fully include students with disabilities in computing activities. The Knowledge Base can be accessed by following the “Search Knowledge Base” link on the *AccessCS10K* website.

The Knowledge Base is an excellent resource for ideas that can be implemented in computing programs in order to better serve students with disabilities. In particular, the promising practices articles serve to spread the word about practices that show evidence of improving the participation of people with disabilities in computing.

Examples of Knowledge Base questions include the following:

- How can I make my computing department more accessible to students with disabilities?
- What adaptive technology is typically provided to students with disabilities on postsecondary campuses?
- What are specific computer applications that can assist students with learning disabilities?
- Are there any web-based tutorials on accessibility?
- How can principles of universal design be used to construct a computer lab?

Individuals and organizations are encouraged to propose questions and answers, case studies, and promising practices. Contributions and suggestions can be sent to accesscs10k@uw.edu.

For more information on *AccessCS10K*, universal design, and accessible computing education, review the following websites and brochures.

- To find more information on universal design in education, visit the Center for Universal Design website at www.uw.edu/doiit/programs/center-universal-design-education/overview.
- To learn more about and get involved with *AccessCS10K*, visit www.uw.edu/accesscomputing/accesscs10k.
- To learn more about and to get involved with a similar project, *AccessComputing*, visit www.uw.edu/accesscomputing.
- To read profiles of computing professionals and students with disabilities, visit the Choose Computing profiles at www.uw.edu/accesscomputing/get-informed/choosecomputing/profiles.



Acknowledgments

AccessCS10K capacity building activities are funded by the National Science Foundation (Grant #CNS1440843). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the CBI presenters and publication authors and do not necessarily reflect the views of the National Science Foundation.

DO-IT
University of Washington
Box 354842
Seattle, WA 98195-4842
doit@uw.edu
www.uw.edu/doit
206-685-DOIT (3648) (voice/TTY)
888-972-DOIT (3648) (toll free voice/TTY)
206-221-4171 (FAX)
509-328-9331 (voice/TTY) Spokane

AccessCS10K Principal Investigators: Richard E. Ladner, Ph.D.
and Andreas Stefik, Ph.D.
Co-principal investigator: Sheryl Burgstahler, Ph.D.



© 2015 University of Washington. Permission is granted to copy this publication for educational, noncommercial purposes, provided the source is acknowledged.