Mapping Accessibility Assignments into Core Computer Science Topics: An Empirical Study with Interviews and Surveys of Instructors and Students

Emily Kuang
Golisano College of Computing and Information Sciences
Rochester Institute of Technology
ek8093@rit.edu

Selah Bellscheidt
Computer Science
Western Washington University
bellscs@wwu.edu

Di Pham
Computer Science
Western Washington University
phamd7@wwu.edu

Kristen Shinohara
School of Information
Rochester Institute of Technology
kristen.shinohara@rit.edu

Catherine M. Baker
Computer Science
Design & Journalism
Creighton University
CatherineBaker@creighton.edu

Yasmine N. Elglaly
Computer Science
Western Washington University
elglaly@wwu.edu

ABSTRACT

Incorporating accessibility education into undergraduate computer science (CS) programs is essential for preparing future technology professionals to create inclusive technology. However, many CS programs lack accessibility coverage, often confining it to human-computer interaction (HCI) courses. To address this gap, we developed accessibility assignments seamlessly integrated into core CS courses. We collaborated closely with ten instructors to select and customize these assignments to suit their needs. To evaluate the impact of these assignments, we conducted interviews with instructors and administered surveys and interviews with their students. Our findings indicate significant improvement in students' familiarity with accessibility concepts and confidence in implementation following completion of the assignments. However, their mindset and future interest in accessibility remained the same. Instructors found it straightforward to incorporate these assignments without compromising core computing concepts. In sum, we validated a foundation for effectively resourcing instructors with accessibility teaching materials and increasing their capacity in accessibility knowledge.

CCS CONCEPTS

• **Human-centered computing** → Accessibility; • **Social and professional topics** → Professional topics; Computing education.

KEYWORDS

accessibility education, computer science instructors, computer science courses

CHI '24, May 11–16, 2024, Honolulu, HI, USA

ACM ISBN 979-8-4007-0330-0/24/05
https://doi.org/10.1145/3613904.3642097

ACM Reference Format:
Emily Kuang, Selah Bellscheidt, Di Pham, Kristen Shinohara, Catherine M. Baker, and Yasmine N. Elglaly. 2024. Mapping Accessibility Assignments into Core Computer Science Topics: An Empirical Study with Interviews and Surveys of Instructors and Students. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11–16, 2024, Honolulu, HI, USA.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3613904.3642097

1 INTRODUCTION

The ubiquity of technology in modern society underscores the importance of ensuring that they are accessible to all individuals. As technology evolves, so too must the education of future technology professionals in the principles and practices of accessibility. Currently in computer science (CS) education, accessibility is confined primarily to specialized courses, such as human-computer interaction (HCI) [25, 34], rather than being directly integrated into core CS curriculum; this approach is dependent on the personal initiative of the course instructor [36]. Furthermore, many instructors do not feel comfortable teaching accessibility due to having limited knowledge in the area [36]. This compartmentalization of courses and lack of instructor expertise raise concerns about if students are adequately equipped to incorporate accessibility into their future work as CS professionals.

To tackle this issue, our research bridges the accessibility education gap in CS programs by pioneering an approach to directly incorporate accessibility assignments into core CS courses. Instead of a dedicated course on accessibility, which risks being discontinued due to low enrollment [5], we exposed students to accessibility concepts within foundational CS courses. We developed three versions of assignments that covered accessibility topics on Braille, Web Content Accessibility Guidelines (WCAG) [18], and accessibility checkers, while ensuring that students still met the technical learning objectives for core topics: dictionaries, ArrayLists, and binary trees. Previous work on integrating accessibility into CS courses was largely implemented by instructors with prior accessibility experiences [31, 35, 48], therefore we focused on fusing topics for instructors without accessibility backgrounds. Our goal was to enable instructors to teach a topic they may not have expertise in.

Through meetings with instructors prior to the academic term, we tailored these assignments to align with their unique pedagogical needs, ensuring a smooth fit into their courses.

To evaluate the impact of these assignments, we conducted experience sampling and interviews with eight instructors and interviews with 15 students to gather qualitative feedback about the assignments from eight different institutions. We collected 249 matched student responses to the pre-assignment and post-assignment surveys, which measured their accessibility knowledge, implementation, mindset, and future interest. Our study illustrates that instructors found our accessibility integration approach easy to implement, affirming that it exposed students to accessibility concepts without compromising their grasp of core computing concepts. Our findings also reveal significant advancements in students' understanding of accessibility concepts and their confidence in developing and evaluating accessibility features.

In sum, our research validated a foundational approach for addressing the accessibility education gap within undergraduate CS programs. By providing instructors with resources and enhancing *their* accessibility knowledge, we contribute a strategy about how to equip future technology professionals with accessibility knowledge toward creating a more inclusive digital landscape.

2 RELATED WORK

2.1 Accessibility Incorporation in The CS Curriculum

In previous research studies, instructors incorporated accessibility into computing courses in two main ways: throughout the entire curriculum and in specific elective classes [3, 6, 13, 15]. Teaching accessibility as part of a broader curriculum has been more sustainable than accessibility-specific programs, which were discontinued due to low enrollment [5]. Accessibility was also taught in specific elective classes such as accessibility in software engineering [10, 30] and universal design [23]. In such courses, student knowledge and awareness of accessibility increased, as well as comfort level in interacting with people with disabilities [23]. In other courses, accessibility was not the main topic, but was integrated as a theme throughout [4, 26, 37, 45]. This method was frequently used in HCI courses. In other work, instructors integrated accessibility through the use of accessibility modules within a non-accessibility specific class [12, 19, 29, 44, 48]. Although some educators and researchers have explored introducing accessibility into computing education, most practices integrate accessibility into elective, not core, CS courses. A few efforts have been made to add accessibility into core courses [3]. We expand on prior work by investigating effective approaches to incorporate accessibility into core courses.

2.2 Accessibility Teaching and Instructors Preparedness

Accessibility is taught through different pedagogical approaches, including lectures, group projects, in-class activities, guest speakers, videos, interactions with people with disabilities and research [3]. Poor et al. investigated the impact of class projects involving building and testing a UI that includes non-mouse- and non-keyboard-based input–on student knowledge of accessibility [35], finding that students developed a greater understanding and awareness of accessibility and usability issues. Palan et al.'s work examining a week of lectures during an HCI course, focused on topics such as accessible website design, common assistive technologies, and legal requirements, similarly revealed an increase of knowledge and awareness [31]. Another study found that accessibility modules increased student learning of accessibility and motivated them to create accessible software [12]. Zhao et al. conducted a longitudinal study comparing four different methods for teaching accessibility–a week of lectures, team design project involving accessibility, interaction with a person with a disability, and collaboration with a team member with a disability. This study found that lectures, team projects, and interaction with people with disabilities gave students greater awareness of accessibility issues and increased the likelihood that they would design with accessibility in mind in the short term. However, when surveyed two years later, students showed no significant improvements in their understanding of accessibility than before the class that incorporated accessibility [48]. Students also viewed accessibility as a specialized skill that was not applicable for future jobs [8]. These results suggest that a single class may not have enough influence on students' knowledge of accessibility long-term. In sum, hands-on activities and experiential learning were found to be effective methods for teaching accessibility in the short term [12]. Our work follows a similar approach as we incorporated accessibility into programming assignments, providing the students with a hands-on learning experience.

Teaching accessibility was shown to be challenging for instructors who are not experts in accessibility [38], and thus is largely reliant on the personal initiative of the instructor. This need for expertise may explain why HCI instructors are more likely to teach accessibility [36]. Due to these instructor characteristics and preferences, barriers to teaching accessibility include lack of knowledge, lack of course materials, limited preparation time, little administrative support and no space in the curriculum to include new topics [21]. To address the accessibility knowledge gap of computer science instructors, Kawas et al. proposed a "micro professional development" model to provide instructors with knowledge and resources on how to integrate accessibility into core curriculum by mapping accessibility topics to learning objectives in CS courses [21]. Building on this recommendation, we worked closely with instructors to map accessibility learning objectives to CS topics and equipped instructors with the necessary materials and resources for effective implementation.

2.3 Accessibility Skills and Learning Outcomes

Computing curriculum that includes accessibility generally tends to focus on a common set of learning objectives, even though no such objectives have yet been officially established: technical knowledge of guidelines and requirements; empathy, *e.g.*, inclusive design; and future career pathways in accessibility [3]. When teaching design, an emphasis is placed on how to prevent or remove barriers that people with physical disabilities encounter [38]. Accessibility experts offered recommendations on accessibility and disability knowledge that can be covered in software engineering and machine learning courses, such as accessibility requirements and machine learning bias [9, 10, 24]. Whereas our previous work identified skills desired

by industry [18], the lack of official learning objectives motivated our concurrent proposal defining five accessibility knowledge units for CS courses [11]. However, this nascent curricular area has yet to be explored fully.

To determine what accessibility skills are sought by software employers, Martin et al. analyzed LinkedIn job posts, which found that employers are not seeking accessibility skill sets in general software jobs [28]. Instead, job postings request generic accessibility knowledge or expertise in specific guidelines and implementation techniques, such as Web Content Accessibility Guidelines (WCAG) or Accessible Rich Internet Applications (ARIA) [17]. Accessibility-specific roles often requested candidates be able to educate coworkers in topics such as WCAG and accessible design and to advocate for accessibility [28]. These skills comprise expertise that tech companies seek in job applicants, and specifically show that employers desire accessibility knowledge but may be overwhelmed by compliance issues and the increasing demand (*i.e.*, seeking applicants to help educate others). As such, they inform our understanding of the growing technical demand for accessibility knowledge and skills. Informed by prior work, we chose accessibility learning objectives that are (1) foundational to accessibility education, and (2) required by the tech industry.

2.4 Evaluation Methods of Accessibility Modules

Researchers have employed a variety of methods to evaluate the effectiveness of accessibility modules. In most prior studies, the evaluation of accessibility modules was predominantly based on collecting data from students. Evaluation methods include the use of pre- and post-course questionnaires and surveys [3, 20, 22, 31]. Researchers also evaluated accessibility modules through distribution of quizzes and other graded assignments to students [3]. Prior work also investigated completion rate and retention to determine the longitudinal impact of accessibility teaching methods [48]. Additionally, researchers interviewed students after course completion to collect qualitative data on their learning experience [3, 19]. To our knowledge, no prior work collected feedback from instructors in order to evaluate accessibility modules. Papers dedicated to instructors focused on their overall experience in teaching accessibility, rather than evaluating specific modules, and these instructors were well-versed in accessibility [21, 36]. Additionally, in most prior work, individual instructors implemented their modules at their home institutions [12, 29, 31, 48]. Thus, there is a need for understanding the instructors' perspective and interinstitutional analysis of the efficacy of accessibility modules. Our study addresses this need by recruiting instructors from different institutions across the United States and gathering feedback from instructors in addition to students.

3 METHOD

In this IRB-approved research, we sought to determine the feasibility and impact of integrating accessibility topics into programming assignments in core computing courses by CS instructors. First, we created and piloted two assignments that combined accessibility topics and core computer science learning objectives. We ran the pilot from March to May 2022. Then, adjusting our approach based on the pilot, we investigated how eight CS instructors from five different institutions used the assignments from September 2022 to April 2023. We also studied how well students in these courses learned CS and accessibility concepts. We collected impressions and feedback from professors throughout the term, surveyed students on their assignment experience, and interviewed professors and students about their experiences. We used experience sampling to gather insights from professors, pinging them with quick questions at regular intervals (about once a month). We used this approach to elicit quick, short, and in-the-moment answers from busy instructors.

3.1 Assignments

Drawing on our expertise in the accessibility and CS domains, we developed assignments that encompassed both accessibility learning objectives and core CS learning objectives. In consultation with the instructors in our study, we refined and adapted these assignments to align more closely with the specific courses they were teaching. Our study involved the following assignments.

3.1.1 Accessibility Tests Analysis. This assignment focuses on searching through an ArrayList. In this assignment, we look at the results of four different automated accessibility checkers on a test suite put together by the UK government. Each test violates a single accessibility guideline on a website. Students are first asked to create a class that encapsulates the data of a single test (the test category, test description, and the results for each of the checkers on that test). Then all the tests are stored in an ArrayList and students are asked to write methods that allow users to query the results for things like the number of tests that a checker passed in a specific category, the number of test descriptions that contain specific keywords, etc. In the introduction to this assignment, students are provided with examples of assistive technology (AT), such as screen readers, as well as the Web Content Accessibility Guidelines (WCAG). Through the assignment, students gain insights into the different tests that are needed to ensure users can use the AT described in the beginning, other barriers that can exist, and the current state of automated accessibility checkers.

3.1.2 Braille Translator with Binary Tree. The goal of this assignment is to create a digital Braille translator, similar to those existing online [42]. Braille translators are a type of assistive technology that can be used for learning Braille, converting English textbooks to Braille, and embossing Braille. This assignment tasks students with implementing a binary tree that translates Braille characters into English letters. The learning objectives of the assignment include understanding and implementation of binary tree data structures and traversing binary trees using recursion while concurrently promoting awareness and appreciation of Braille. The assignment provides an overview of the Braille writing system, its significance and refreshable braille displays for digital interactions. It highlights the importance of understanding braille patterns and introduces the notion of representing Braille characters as strings of 0's and 1's. The assignment introduces the concept of representing Braille encodings as paths within a binary tree. Each level of the tree corresponds to a position within a Braille cell, and the leaf nodes at the seventh level store the English characters associated with

Table 1: Summary of Pilot Data Collection

Instructor	Institution Type	Course	Assignment Used	Number of Pre-surveys	Number of Post-Surveys	Number of Interviews
P1	Liberal arts college/university	Core Concepts in Computer Science	Braille Dictionaries and Text Analysis	9	1	1
P2	Liberal arts college/university	Data Structures & Algorithms	Braille Binary Tree	22	9	3

specific paths. The students are required to construct a binary tree that represents the Braille encoding of the English alphabet and traverse the entire binary tree to produce a file that maps English letters to their respective Braille encodings.

3.1.3 Braille Translator with Dictionaries and Text Analysis. This assignment requires students to implement a Braille translator capable of converting English sentences into Grade 2 Braille representation using dictionaries. Furthermore, students are tasked with creating visual representations of Braille characters using Python's turtle graphics library. The learning objectives of the assignment include handling text files, applying string operations, creating and manipulating lists and dictionaries, and using the turtle graphics library. The objectives also involve understanding Braille encodings, both grade 1 and grade 2. The assignment introduces digital Braille (*i.e.*, digital representations of Braille on a screen) and its binary encoding scheme using 0s and 1s, while also showcasing examples of the contracted (Grade 2) Braille, commonly used by experienced Braille users. Students are instructed to split a string into a list of words. If a word is contained in the dictionary (*e.g.*, the Grade 2 Braille words), it should substitute it with the associated Braille character. Otherwise, the word is parsed into characters. Each character is the key, and the replacement Braille character is the value associated with that key in the dictionary. Additionally, students were tasked with using the turtle graphics library to draw six circles corresponding to the six circles in a Braille character. The raised dots should be represented by filled-in circles and the flat dots should be represented as empty circles.

3.2 Pilot

For the pilot, we recruited two CS instructors at two different institutions to test-run the assignments and collect feedback from their students through pre-assignment and post-assignment surveys and interviews. Table 1 shows a summary of their background information including the institution type, course, and assignment as well as the number of pre- and post- survey responses and interviews collected from each instructor's class.

The pilot pre- and post-surveys had 6 sections: demographics, accessibility knowledge, accessibility implementation (confidence with developing and evaluating assistive technologies), empathy (understanding of people with disabilities), mindset (opinion about the role of accessibility in computer science), and future interest (willingness to consider accessibility in subsequent projects and careers). We observed a steep decline in the number of responses for the post-survey.

We conducted follow-up interviews with four students after the courses concluded. The reactions to the assignments were generally positive, with a student from P1's class mentioning that *"I thought it was a super interesting assignment, especially just the idea of encoding Braille and binary. I really do think it's important to do things like this, putting disabled people and computing together."*

From this pilot, we made changes to the assignment as well as the survey questions. We clarified the assignment instructions and added more background information about assistive technology. We also revised the pre- and post-surveys to focus on the learning objectives of the assignments: we removed the empathy section and transferred relevant questions about different disabilities into the knowledge section because pre-survey responses were overwhelmingly positive, precluding meaningful differences in the post-survey, and we removed questions asking about accessibility knowledge not covered by the assignments (*i.e.*, familiarity with the Americans with Disabilities Act). We changed wording to be clearer and more specific, for example, "When designing or developing future projects, I plan to consider accessibility" to "I plan to consider accessibility in the beginning of future projects." We adjusted our study design to offer increased incentives to reduce the drop-off between pre- and post-assignment surveys and opted to collect student names to allow matching between surveys and confirm changes for individual student learning.

3.3 Instructor Data Collection

3.3.1 Recruitment and Onboarding - Before the Term Begins. Recruitment efforts focused on reaching instructors from different types of institutions with different backgrounds, including instructors who had no experience teaching accessibility topics. We advertised on social media and sent recruitment messages to the TeachAccess [40], AccessComputing [43], and SIGCSE [1] mailing lists. We also compiled a list of minority serving institutions and reached out to the computer science department chair at those institutions. We recruited eight instructors from five institutions, whose teaching experience ranged from 0.5 to 20 years (Table 2). The CS courses where the accessibility assignments were integrated covered data structures, object-oriented programming, and other advanced programming topics.

We met with instructors to share our research goals, explain study logistics and timeline, and facilitate a discussion about assignment details to determine which assignment would best fit their

Table 2: Background of Instructors and Information about the Course and Assignments Used

Instructor	Institution Type	Years of Teaching Experience	Area of Expertise	Prior Experience with Teaching Accessibility	Course	Assignment Used	Assignment Adjustments
M1	Community college	15 years	Computer Science, Java	None	Java Data Structures	Accessibility Tests Analysis	Made an activity extra credit
M2	Liberal arts college	7 years	Human-computer interaction	Yes, during user interface development	Object-Oriented Problem Solving, Data Structure, and Algorithms	Accessibility Tests Analysis and Braille Binary Tree	Added an activity to create a client program
M3	Liberal arts college	5 years	Assistive technology	Yes, runs an accessibility lab and included it into courses	Computational Thinking: Visual Media	Braille Dictionaries and Text Analysis	Removed one activity
M4	R2 university	0.5 years	Human-computer interaction	None	Advanced Programming	Accessibility Tests Analysis	Added JUnit tests
M5	R2 university	6 years	Human-computer interaction	Yes, included it into courses	Advanced Programming	Accessibility Tests Analysis	Added JUnit tests
M6	R2 university	5 years	Internet of Things and Game Theory	None	Advanced Programming	Accessibility Tests Analysis	Added JUnit tests
M7	R2 university	13 years	Computer architecture	None	Advanced Programming	Accessibility Tests Analysis	Added JUnit tests
M8	R1 university	20 years	Computer theory and algorithms	None	Intro to Computer Science	Braille Binary Tree	N/A

course. During the meeting, instructors provided their course syllabi. Researchers collaborated with instructors to determine which existing course assignment could be modified to add accessibility components or swapped with an equivalent accessibility-infused assignment that the researchers designed. Based on their teaching styles and needs of their students, some instructors also requested adjustments to the accessibility-infused assignments (e.g., M4-M7 wanted to add JUnit tests to align with the other assignments in their courses). Instructors were then given a consent form to sign to participate in the study. Instructors received $140 as compensation for their participation.

3.3.2 Experience Sampling - During the Term. Researchers emailed instructors at three points in the term—post-lecture, post-assignment, and post-grading—to solicit feedback on their experience. The post-lecture email asked about changes they made to the assignment, time given to complete the assignment, how they introduced the assignment to the class, and accessibility-themed content that was covered in addition to our assignment. The post-assignment email elicited questions and comments instructors received about the assignment. The post-grading email asked for statistical data on the number of students who had completed the assignment and grade statistics. Instructors were asked to give feedback on the assignment including if it fulfilled learning goals, what went well, and what they would change.

3.3.3 Interview - After the Term. To understand instructors' experience of teaching the accessibility-themed assignments, we conducted a 30–45-minute interview with instructors via Zoom after the term ended. Interviews were video recorded and later transcribed for analysis. The interview contained questions about background in academia and accessibility, assignment experience and implementation, and opinions on the assignment and integrating accessibility into the classroom. For example, we asked, "Have you taught accessibility before? If so, in what context?", "Did any students comment on or ask questions about the accessibility material? If so, what comments or questions did they ask and how did you address the comments or questions?", and "Do you plan to use this resource in the future? Why or why not?"

3.4 Student Data Collection

3.4.1 Survey. On the day that the assignment was introduced, a researcher joined the instructor's class over Zoom to introduce students to the project and survey. Students were given the extra credit option of completing either the pre-survey, or an alternative assignment, which involved watching a 10-minute accessibility-themed video and completing a quiz. To incentivize participation in the study, we offered a $25 raffle. We raffled an additional $25 for classes where we collected over 25 responses.

After completing the assignment, researchers joined the class again via Zoom session to introduce the post-assignment survey. The process and compensation were the same as the pre-survey, with the exception that the post-survey asked students to provide an email address if they wanted to participate in a follow-up interview.

The purpose of the surveys was to determine what students had learned from the assignment and how their thoughts, and opinions on accessibility were impacted by the assignment. We collected students' names in pre- and post- surveys so that we could connect survey responses to make direct comparisons. Survey questions asked students about: demographics, accessibility background knowledge, accessibility implementation, mindset (opinion about the role of accessibility in computer science), and future interest.

3.4.2 Interview - After the Term. We conducted interviews via Zoom with students after the term had ended to learn more about the impact of introducing accessibility-themed assignments on students' understanding of accessibility. Interviews lasted 30-60 minutes and students were compensated $20. Students answered questions on their academic background, course takeaways, experience completing the assignment and prior knowledge on the topic of accessibility. We also asked about what they learned from the assignment and how it influenced their opinions on accessibility, including its relevance to computer science students, and what they would change about the assignment. Interviews were video recorded and transcribed for data analysis.

3.5 Data Analysis

3.5.1 Experience Sampling and Interviews. The interviews with instructors and students were automatically transcribed using the transcription service from Zoom. Then a researcher rewatched the recording to correct any transcription errors. Experience sampling responses from instructors were already provided in written form via emails. Interview transcripts and experience sampling data were combined for analysis for each instructor. Two researchers independently coded the interviews and experience sampling responses using an open coding approach [7]. They then met to discuss and consolidate codes. Afterward, they grouped the codes to identify themes. For example, here is an excerpt of the codebook from the student interviews:

- Feedback/Improvements for the assignment
- Appreciated creativity and real-world application
- Instructions and examples of outputs could be more clear
- Wanted more hands-on accessibility tasks
- Impact of the assignment
- Eye-opening and encouraged future interest
- Willing to apply accessibility knowledge in future assignments
- Less pronounced for those with prior accessibility experience

3.5.2 Surveys. We compiled the pre- and post-assignment survey responses from all sections and replaced all student names with a participant code. To obtain matched data between the pre- and post-surveys, we removed the responses associated with participant codes that existed only in the pre-survey or only in the post-survey. All responses on the Likert, familiarity, and confidence scales were converted to numerical values from 1 to 5. Then we conducted paired two-tailed t-tests to identify if there were any significant differences between the pre- and post-survey responses. For questions that resulted in significant differences, we calculated the effect size using Cohen's d, where 0.2 is considered a small effect, 0.5 is medium, and 0.8 is large [47].

4 RESULTS

In this section, we present detailed insights gleaned from instructor experience sampling and interview data and student survey and interview data. Instructors generally found it easy to integrate accessibility assignments into their courses without compromising core computing concepts and gave suggestions on how to improve the clarity of assignment instructions. Survey responses indicated that students were significantly more familiar with accessibility concepts and confident in accessibility implementation after the assignment, while their mindset and future interest in accessibility remained the same.

4.1 Instructor Experience Sampling and Interview Data

In this section, we describe the feedback from the instructors on their prior accessibility experience, perspectives on incorporating accessibility into CS, student demographics and engagement, and overall experience teaching the assignment.

4.1.1 Experience Teaching Assignment (Preparation, Grades, Willingness to Reuse). To begin, most instructors completed the assignment themselves to become familiar with its mechanics and concepts. Most instructors used the assignments as provided with a few adjustments after we had worked with them to replace an existing assignment. However, some instructors made minor changes as needed. For example, one instructor created lecture slides to introduce the assignment in anticipation of student questions. Another instructor asked their lab TAs to complete the assignment and give suggestions about the clarity of the instructions. Almost all instructors reported that the accessibility assignment grades were comparable to the original assignment that it replaced. For the Braille Binary Tree, M8 observed a lower completion rate than other assignments. Based on statistics about how long students were taking on the assignment, the instructor hypothesized that this difference could be attributed to students not budgeting enough time to work on the assignment, having difficulties working with files in ZyBook, and encountering confusion when skipping over symbols not defined in the Braille dictionary.

> "And I'm also seeing a big spike of when the assignment was started, and it's kind of late. So if you ask

> me what was the problem, I think this is a problem. [Students] really need to spend some time on it…So technically, I don't see it as more difficult than other assignments, but in terms of understanding how to translate the problem to code, I think it needs a little more time." - M8

When asked whether they would be willing to reuse the assignment, most instructors said yes since they found value in using real-world examples to teach technical skills, which are often applied to "toy" examples or datasets. However, several added that to prevent cheating and maintain novelty, they would wait a few terms to reuse the assignment or modify it. Two instructors said that they would consider reusing the assignment if modifications were made. Specifically, in the accessibility checkers assignment, M1 said they would incorporate inheritance into the assignment. M3 indicated that they would be willing to use the Braille Text Analysis assignment if they received a new version since they felt the clarity of the instructions needed to be improved. In general, instructors proposed breaking assignments into smaller, manageable steps and incorporating GUI or visualization components to aid student comprehension of assignment results.

Overall, after giving the assignment, most instructors indicated that there were no major differences between our assignment and the original assignments given in the course. Almost all instructors indicated they would reuse our version, with two instructors who would consider reusing the assignment after modifications.

4.1.2 Prior Accessibility Experience and Perspective on Incorporating Accessibility into CS.. Instructors who had prior experience teaching accessibility incorporated it into their previous classes through several methods, including as a topic in design classes, in class activities, and in a dedicated accessibility module in a software development class, which consisted of a two-week unit where students learned to evaluate the accessibility of products, including their own creations.

Two instructors discussed using simulation as a tool to broaden student awareness of other people's perspectives, but simulations are not without their drawbacks, as M5 expressed. While M5 had asked his students to use a screen reader blindfolded in order to create a scenario where the students had to rely on the screen reader, he did not "want to miscommunicate to students that people who are using screen readers can turn off their disability or something, or that they can wear it like a decoration… but I did want to kind of press them to try a little bit better on their use of the screen reader".

Although more than half of the instructors had no prior experience teaching accessibility, all instructors believed that accessibility belonged in the computer science curriculum. However, some instructors who lacked a background in accessibility did not feel comfortable making recommendations about whether it should be taught in introductory or advanced courses, or both. The other instructors suggested that accessibility should be included as a theme throughout the curriculum, pointing out that including accessibility in one class will not create the mindset shift necessary for most students to actively consider accessibility in their future work. For example, M8 commented:

> "…this major shift will not happen through one class, or even one course. . . . I feel like even if people took a dedicated class, very few mature and reflective students would probably go through transformation in that class, but not the majority of students. And I think for the majority of students, small steps here and there are the way to go." - M8

Similarly, M7 highlighted the importance of consistent curricular emphasis on accessibility to prevent students from believing that accessibility is irrelevant to their future career because certain subjects do not teach anything about accessibility:

> "I think that ideally it would be taught in different aspects throughout the curriculum as just something that we as computer scientists should be aware of and know something about… They could even say, "Oh, I'm going to be a systems person. I don't need to understand this stuff." And so I think that the only way that we really change perceptions in a more serious way is to have it embedded throughout the curriculum and valued by multiple faculty at each level of the curriculum." - M7

Other instructors emphasized that accessibility should be taught in introductory courses, suggesting that these topics could help to recruit students of diverse backgrounds:

> "I think [accessibility] should be taught in introductory courses and advanced courses… if you are including it in introductory courses, it might be able to try and draw more women or other diverse populations, or even maybe disabled people who have some of these disabilities" -M1

Overall, instructors were supportive of incorporating accessibility education into computer science curriculum, regardless of whether they had no prior knowledge, or had included accessibility topics in prior courses.

4.1.3 Class Context and Engagement. Instructors reported that most students were computer science majors or students intending to apply to the computer science major. One instructor's course had no prerequisites, suggesting a different student background than computer science majors. Another institution only offered computer science as a double-major, so these students were intending to have computer science as one of their majors.

Engagement with class activities and effort on assignments varied widely among students. Instructors commented on this phenomenon, indicating that it may have been residual fallout from student experiences during the height of the COVID-19 pandemic rather than any characteristics of the accessibility assignments.

> "These past few semesters after COVID-19, there is this accumulated fatigue. I think other instructors will talk about this too. So I cannot say that people are super enthusiastic, to be honest. Regardless of content, your assignment or my assignments, I have to pull students' teeth to move forward and I had to maybe have more oversight, more supervision. Like usually students are more independent. I can assign things, and they work on them on their own. Right now they seem too easy to drop something. If they can't make it the first time, they just drop it." -M8

Table 3: Number of Student Survey Responses and Matched Responses

Instructor	Number of Pre-surveys	Number of Post-Surveys	Number of Matches
M1	16	16	13
M2	36	32	25
M3	43	44	38
M4	77	58	55
M5	24	23	20
M6	33	28	21
M7	47	40	33
M8	71	47	44
	347	**288**	**249**

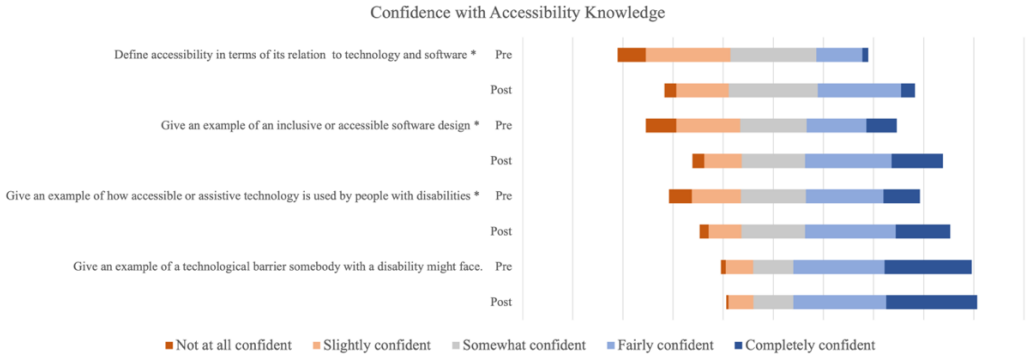

Figure 1: A diverging stacked bar chart showing the distribution of pre and post survey responses to statements about students' confidence with accessibility knowledge. Statements that resulted in significant differences between pre and post survey responses are annotated with an asterisk (*).

Instructors shared that students were more focused on the technical objectives of the assignment, rather than the accessibility concepts. To counteract this tendency, three instructors who had a background in accessibility tried different teaching strategies to drive home the accessibility concepts by employing multiple modalities (*e.g.*, videos and demos). M2 posted a link to a talk on the history of accessibility on their course website, M3 added additional links to resources about Braille and videos about blind and low vision people, and M4 gave an introduction to the assignment by demonstrating website accessibility for screen readers, which drew students into the topic.

> "I showed them this is what a screen reader is, this is how it works, and I showed them examples of a website that's accessible to screen readers and one that isn't. I think they were pretty engaged because they were asking you like oh, try clicking on that, what does that say? Oh, that's awful" - M4

Although M5 also had an accessibility background, he mentioned that he did not have time to find additional accessibility resources for this class.

4.2 Student Survey Data

The survey was divided into four main sections: knowledge, implementation, mindset, and future interest. In this section, we overview the combined survey results from all classes, showing that significant differences were observed for knowledge and implementation, but not for mindset and future interest. Survey responses across the different courses ranged from 16 to 77, averaging 31 matched responses (Table 3). To make a direct comparison between student responses before and after the assignment, the following graphs only contain the matched survey responses.

4.2.1 Accessibility Knowledge. Three out of four statements about accessibility knowledge led to statistically significantly higher responses: (1) 38.8% of students felt fairly or completely confident that they could define accessibility in terms of its relation to technology and software compared to 20.7% in the pre-survey ($p < .0001$, $d = 0.5$), (2) 55.1% were fairly or completely confident that they could give an example of an inclusive or accessible software design compared to 36.0% in the pre-survey ($p < .0001$, $d = 0.5$), and (3) 57.9% were fairly or completely confident that they could give an example of how accessible or assistive technology is used by people with disabilities compared to 45.5% in the pre-survey ($p < .001$, $d = 0.3$) (Figure 1). These differences suggest that the assignments were successful in increasing students' confidence in their knowledge of accessibility and assistive technology.

Regarding students' familiarity with different populations and accessibility concepts, students indicated they were significantly more familiar with blind or low vision people ($p < .001$, $d = 0.4$),

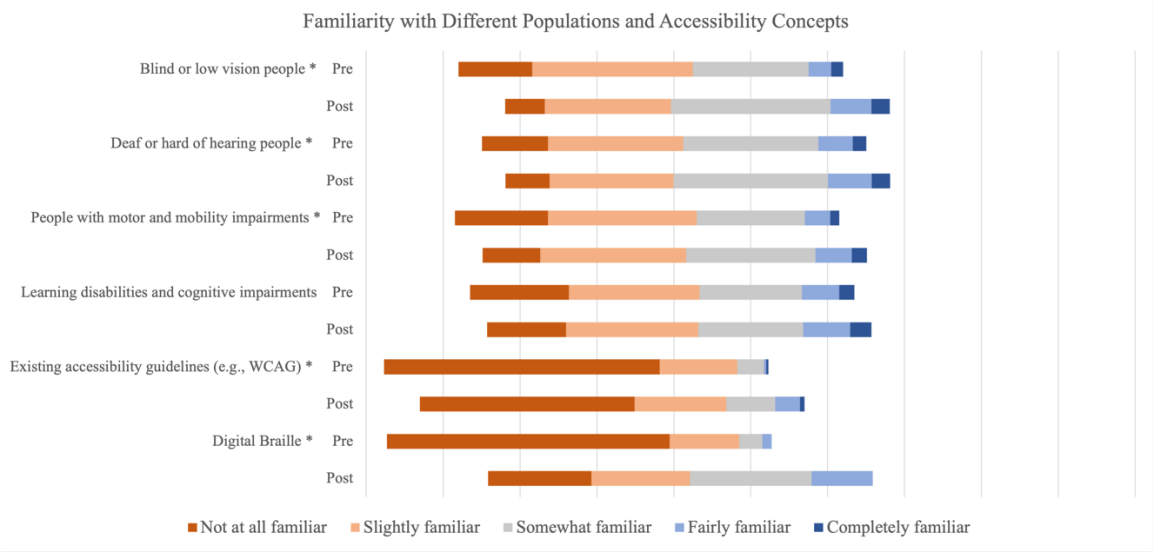

Figure 2: A diverging stacked bar chart showing the distribution of pre and post survey responses to statements about students' familiarity with different populations and accessibility concepts. Statements that resulted in significant differences between pre and post survey responses are annotated with an asterisk (*).

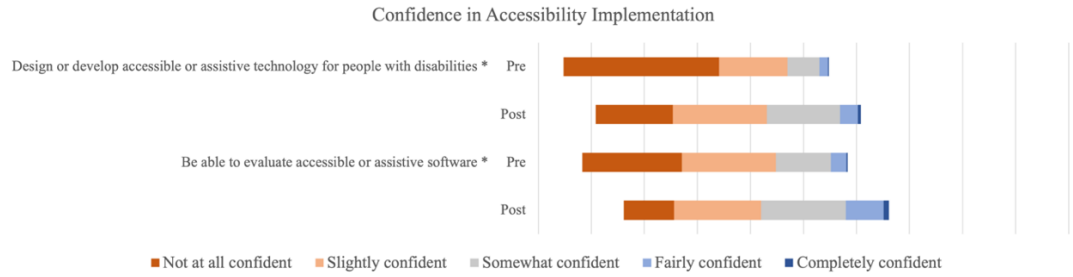

Figure 3: A diverging stacked bar chart showing the distribution of pre and post survey responses to statements about students' confidence in accessibility implementation. Statements that resulted in significant differences between pre and post survey responses are annotated with an asterisk (*).

deaf or hard of hearing people ($p < 0.05$, $d = 0.2$), and people with motor impairments ($p < .01$, $d = 0.3$) after completing the assignment (Figure 2). Prior to the Accessibility Text Analysis Assignment, only 8.1% of students were "somewhat familiar" or more with existing accessibility guidelines. After the assignment, that percentage increased significantly to 20.4% ($p < .0001$, $d = 0.4$). Similarly, 47.6% of students were "somewhat familiar" or more with digital Braille after completing the Braille Dictionary and Text Analysis or the Braille Binary Tree assignment compared to 8.4% before ($p < .0001$, $d = 1.1$).

4.2.2 Accessibility Implementation. Students' survey responses to statements about their confidence with accessibility implementation resulted in significantly different responses between the pre- and post-survey (Figure 3). We observed that prior to the assignment, 15.6% of students were somewhat or more confident about designing or developing accessible or assistive technology. That percentage more than doubled to 35.4% after the assignment ($p < .0001$, $d = 0.6$). Similarly, 48.2% of students were "somewhat confident" or more about being able to evaluate accessible or assistive technology after the assignment compared to 27.0% before ($p < .0001$, $d = 0.5$).

4.2.3 Mindset on Accessibility. In the pre-survey, the median ratings across statements related to mindset on accessibility were either "Somewhat agree" or "Strongly agree," suggesting that the students already had a generally positive mindset prior to completing the assignment. For example, 91.8% of students somewhat or strongly agreed that every programmer should have some accessibility knowledge and 85.4% somewhat or strongly agreed that accessibility concepts should be taught in computing education. The median ratings for these statements remained the same in the post-survey, and we did not observe any significant differences.

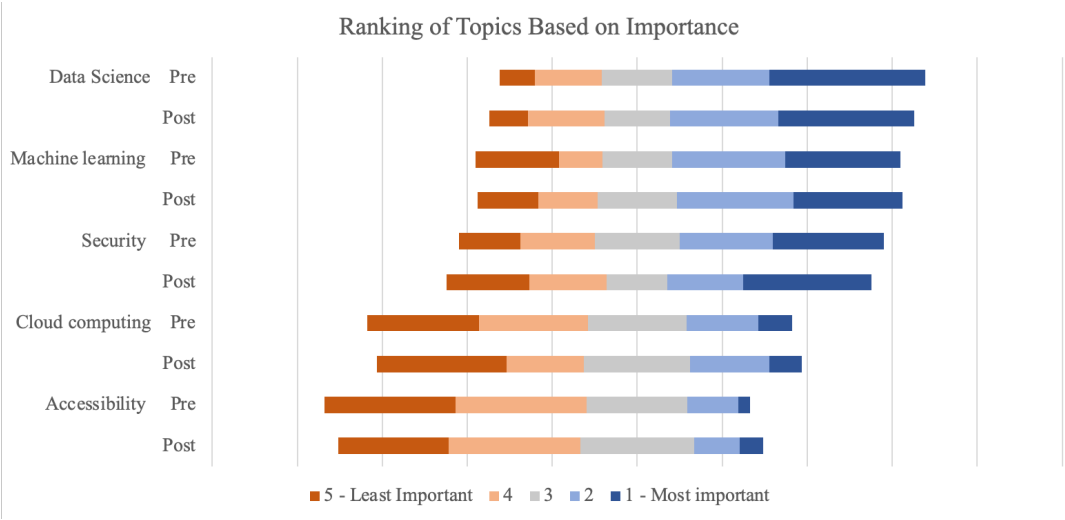

Figure 4: A diverging stacked bar chart showing the distribution of pre and post survey responses to students' ranking of topics based on importance.

We asked students to rank certain topics based on their perceived importance to their computer science degree, where 1 is ranked as most important and 5 is least important (Figure 4). In the pre-survey, the descending order of importance was data science (M = 2.4, SD = 1.5), machine learning (M = 2.7, SD = 1.5), security (M = 2.7, SD = 1.4), cloud computing (M = 3.5, SD = 1.3), and accessibility (M = 3.7, SD = 1.1). In the post-survey, the order remained the same and there were no significant differences compared to the pre-survey, which suggests that the assignment did not have a large impact on students' perceived importance of accessibility relative to the other given topics.

4.2.4 Future Interest in Accessibility. Students' responses to statements about their future interest in accessibility followed a similar pattern to those in the Mindset section since the responses were already generally positive prior to the assignment (Figure 5). For instance, 72.1% of students somewhat or strongly agreed that they were interested to learn more about designing or developing technologies for and with people with disabilities, 64.3% somewhat or strongly agreed that they plan to consider accessibility in the beginning of future projects, and 65.0% somewhat or strongly agreed that they will design and develop future projects with people with disabilities in mind. Although the differences were not significant, we observed a slight increase in the percentage of students who somewhat or strongly agreed that they are interested in pursuing a job or career in accessible technology or a related field after the assignment (27.8% in post-survey compared to 21.7% in pre-survey).

4.3 Student Interview Data

We conducted interviews with 15 students in total across five classes, unfortunately students from three classes did not volunteer for interviews. We asked students about: (1) their major and prior experience with accessibility; (2) any specific feedback they had for the assignment; and (3) their engagement, learning, and impacts from completing the accessibility assignment.

4.3.1 Students' Major and Prior Experience with Accessibility. Most students were computer science majors or double majors. We saw a diverse range in students' prior experience with accessibility, from having no knowledge at all to having worked on research projects centered on assistive technology. Most students had some exposure to accessibility concepts, such as seeing Braille on elevator buttons or noticing that alternative text was provided on some websites. Students with extensive accessibility experience had either had a close relative with a disability, worked with Disability Services at their institution, or participated in summer research projects related to accessibility (*e.g.*, tactile graphics in Braille).

4.3.2 Feedback about Assignments. Students appreciated the creativity and real-world relevance of the assignments, which enhanced their understanding of accessibility concepts. Students felt the assignments challenged them at an appropriate level, promoting critical thinking and problem-solving skills. M2-S5 mentioned that "I guess if it were too complicated, or if it was harder than anything else I had done, it would have discouraged me in a way. I would think that this accessibility stuff, I'd rather just ignore it, because it was hard. . ."

On the other hand, some students encountered challenges in understanding assignment instructions. To address this, they suggested clearer instructions with visual aids to improve assignment navigability. Students struggled with visualizing Braille output, and some recommended adding an audio component to enhance comprehension. Finally, students suggested having more introduction to the accessibility concepts, preferably through videos so that they have a stronger background when completing the assignment (*e.g.*, better understanding of what each accessibility checker means in the text file).

4.3.3 Impact of the Assignment: Accessibility gave Meaning to Abstract Data Structures. Many students mentioned that the assignments were eye-opening and gave them confidence that CS could

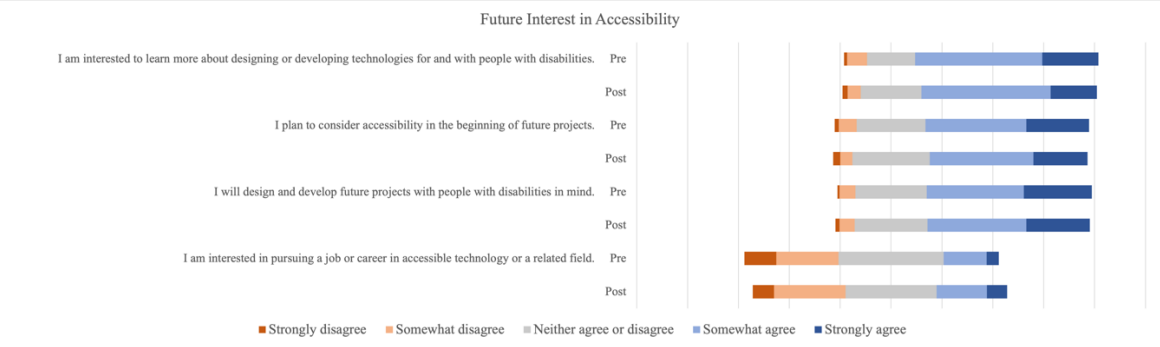

Figure 5: A diverging stacked bar chart showing the distribution of pre and post survey responses to statements about students' future interest on accessibility.

be used to help people. M2-S6 summed it up in the following quote: "I didn't think that CS could be applied to real life in that way. . . So I think it gave me the idea that CS can actually be used to help people... it gave us confidence that we can use the things we learned throughout the course in a practical way." Others felt that the application to real-world examples gave them a new perspective on CS and allowed them to broaden their conception of CS. For example, M2-S5 said "I didn't think that anything that I was doing in my class could even remotely be applied to these things. I think that really makes me appreciate what I've been doing more."

The impact of the assignments varied depending on students' prior exposure to accessibility concepts. Those with limited prior knowledge found the assignments introduced them to accessibility and motivated them to consider it in future projects. For instance, M7-S2 mentioned:

> "I kind of knew there were loose guidelines and stuff like that, but now that I see there's a specific and a laid-out method and everything that people could abide by to make their websites accessible. It's definitely taught me more about it. Now I understand that it's kind of attainable. Honestly, before this, I probably wouldn't have considered accessibility if I was making a website, but now I think that I have experience with it, and you know I would at least look into it." - M7-S2

In contrast, some students with extensive prior knowledge felt the assignments had a lesser impact on them (*e.g.*, "These particular assignments didn't change a ton from me, but only because I already wanted to do accessibility" -M2-S4).

4.3.4 Students Asked for More Accessibility Content. Students reported variable engagement with the assignments, some focused on technical aspects and dedicated time to coding and debugging, while others expressed interest in accessibility concepts. For instance, one student (M4-S2) mentioned concentrating on coding but also navigating accessibility tests, while another (M7-S3) delved into additional research to better understand the input representations.

Several students noted a disconnect between their expectations and the assignment's actual goals concerning accessibility. They expected a more practical approach to making programs accessible but found the assignments leaned towards teaching accessibility concepts. This discrepancy was partly attributed to the introductory nature of the courses, which limited the depth of hands-on accessibility tasks, which students recognized. M2-S2 mentioned:

> "If it was about how we can actually implement it in the real world, maybe that would have been a bit too much for us to handle. We're still in the introductory course learning the basic structures of Java, so I think the assignment made sense." - M2-S2

Despite these challenges, students acknowledged the importance of considering accessibility from the outset of the development process, rather than as an afterthought. While they believed the assignments enhanced their knowledge of accessibility, they also realized the need for more in-depth knowledge and skills to implement accessible solutions, both for their coursework and future careers. Overall, students believed that the assignments provided a valuable experience, increasing their awareness of accessibility's relevance within the field of computer science. They expressed openness to further learning in this domain and suggested hands-on accessibility projects for inclusion in future CS courses.

5 DISCUSSION

Our study yields insights into students' accessibility knowledge, implementation, confidence, mindset, and future interest in accessibility, further enriched by knowledge about instructors' experience with using these assignments. We explore implications for these findings as it pertains to how to equip faculty with teaching materials and resources, how students might learn accessibility, and what changes could help in the future.

5.1 How to Support Instructors to Teach Accessibility in CS

A major barrier in teaching accessibility in computing is the lack of resources available to faculty and their lack of knowledge about accessibility [3]. In particular, instructors feel ill-prepared and uncomfortable delivering material in which they do not have expertise [3]. In this work, we addressed this issue by: (1) supplying instructors with materials, including programming assignments and information on accessibility, and (2) collaborating with instructors on determining which topic in their course is a good candidate

for accessibility infusion, so that they could integrate accessibility topics comfortably into their core computing courses. We provided empirical data that showed that this strategy directly tackles the barrier of lack of resources and knowledge: instructors were able to teach accessibility alongside their main course learning goals. Our instructor support strategy consistently yielded positive outcomes across the various institutions where the assignments were taught. This success was observed despite the variability in the institution type, instructor's years of teaching, and their area of expertise, indicating the generalizability of our instructor support approach.

It is important to note that our approach assumes some initial subject matter expertise to create inaugural materials that can be easily taken up by faculty who are limited in resources and accessibility knowledge [16]. Several researchers on this project had extensive experience in accessibility and were also instructors of CS courses. We went through an iterative process to design and develop these assignments. We started with examining an existing list of assignments in CS courses and selecting possible candidates to infuse accessibility concepts. Then we brainstormed possible ideas and selected accessibility concepts that had tools available for students to check their work, such as screenreaders that are conveniently built into all operating systems and online tools for Braille-English translations. We tested assignments in our own classes (prior to piloting with other instructors) and made improvements through multiple iterations, which included clarifying assignment instructions, adding illustrations and output examples, and making assignment complexity and length adjustable (*e.g.*, designed additional tasks for advanced courses, and developed different versions to require one or two weeks of work). More work is required to develop a set of guidelines for other instructional designers to create similar assignments. We made the assignments in this study available[1] for broader dissemination.

Our approach included consultations with instructors to ensure assignments aligned with their course goals and pedagogical preferences. In undertaking the introductory assignment assembly and scaffolding its integration, we removed common barriers that faculty face (*i.e.*, knowledge and time) [21, 38]. We endeavored to create assignments that could be easily used, regardless of accessibility knowledge, and validated that, with initial parameters in place, teaching accessibility in core computing courses is feasible and sound. Yet, this effort required commitment on our part to prepare the materials in such a way that instructors could easily see the integration into CS concepts they were already teaching. It also required commitment and some effort on the part of the instructors to be willing to include our assignments and adjust them as needed for their courses. Absent this kind of commitment, we would be unable to address the barriers to teaching accessibility. Finally, we were flexible in our approach in support of instructors' autonomy, and although we provided instructors with the basic materials needed to use the assignments, we lowered the adoption effort by modifying the assignments to suit their teaching style and their courses.

The instructors who participated in our study had a mix of prior experience in teaching accessibility, ranging from none to running a lab on assistive technology. The variability in instructor preparedness to teach accessibility aligns with the broader need to employ an even approach to support instructors to incorporate these topics into their courses [36]. Despite these differences, instructors taught accessibility effectively, demonstrating that the design of the assignments and instructor support fulfilled a gap on how to include accessibility in CS [36].

[1] https://accessibilityeducation.github.io/

5.2 Cultivating a Mindset of Accessibility in Students

This study extends prior work that attempted to include accessibility in computing and related courses, in an effort to substantiate accessibility as a core computing concept that ought to be taught comprehensively in computer science [3]. Whereas, prior work focused mainly on HCI, web design, and software engineering courses [10], we targeted core CS courses. And, like prior work, we were able to show that integrating accessibility was effective for enhancing students' knowledge in core courses [12, 31]. Student grades were reported as comparable to typical grades for classes held without our assignments which shows that our materials and methods were effective in infusing accessibility without disrupting knowledge gained about core principles. Interview data showed that the assignments had a positive overall impact on students, but that improvements could be made to foster a deeper understanding and appreciation of accessibility.

We found that students' appreciation of accessibility fell short of a meaningful recognition of the potential of accessibility to be a core aspect of computing (we recall that students continued to rank accessibility below data science, machine learning, and security in post-survey data). Similarly, previous work had shown that there are limitations to how much one course or lecture could change student minds in the long term [8, 31]. Moreover, the technology industry offers a relatively small number of accessibility-focused positions, and general software development positions rarely require accessibility skills [28]. Given the singular exposure to accessibility in education and the lack of visibility of industry's demand for accessibility, it is not surprising that students perceived the importance of accessibility to be lower than other topics, *e.g.*, data science. These findings show that more needs to be done to affect mindset change overall. Indeed, this work shows the effectiveness of including accessibility in core CS courses, mainly that it did not negatively affect student learning about key concepts, and that it raised awareness and increased their knowledge about accessibility. But, future researchers need to build on these practices to increase the number of courses that infuse accessibility, and to broaden the scope of topics that may be included. One approach is to blend accessibility into every part of the curriculum and as part of other career paths, such as in machine learning and data science, similar to Pandey and Dong's work blending accessibility into UI framework documentation [32]. With this approach, we begin to seed introductory computing courses with accessibility, with a long view toward including accessibility in upper division courses and electives as well (aside from HCI, which has been well covered [3, 25, 34]). Furthermore, the finding that the ranking of accessibility among other topics remained unchanged in the post-survey may be a limitation of the question itself. Rankings may

not truly reflect whether mindset on accessibility has improved because it can be seen as a cross-cutting knowledge area instead of a standalone discipline. Future work is required to investigate other ways of measuring students' mindset on accessibility.

Beyond infusing accessibility into courses, academics and educators should be working with industry partners to put accessibility into the requirements of job ads. Recent initiatives such as the Accessibility Skills Hiring Toolkit by Teach Access points towards a future of a knowledgeable and skilled workforce [39]. An increased need for accessibility skills in the tech industry translates into demand for scaling up accessibility in computing education [33]. If students see more accessibility skills included in job ads, they may recognize that accessibility is indeed applicable in their future careers, potentially creating a positive shift in their mindset towards accessibility.

5.3 Recommendations for Integrating Accessibility Throughout the CS Curriculum

Accessibility is—for the first time in 2023—included as part of the updated CS2023 Curriculum guidance [2]. As the CS2023 Curriculum update takes shape, it becomes increasingly crucial to understand how to incorporate accessibility into computing. Researchers have long thought that accessibility is a key component to computing, not just HCI [3]. However, barriers that faculty face, including time constraints and lack of resources and accessibility knowledge, will stymie curricular changes regardless of any official guidance. Our research demonstrated that (1) instructors can successfully adopt new CS materials infused with accessibility content and (2) students can effectively learn accessibility as a fundamental computing concept. We identify two main aspects that contributed to the success of our approach. First, accessibility was added to core courses as a small chunk that does not require prior knowledge and is not too deep nor too complex. This chunking aspect was crucial for the ease of adoption by instructors and ease of learning by students. Second, accessibility was added to the course by swapping an existing assignment rather than adding an extra assignment. This means that the course load for both the instructors and the students remained the same with respect to grading and studying. The accessibility integration approach we have devised is replicable and can be applied to map various accessibility learning objectives into different CS topics [11], thus making accessibility more prominent within the CS curriculum. The implications of introducing accessibility content at an early stage of computing education are profound, as it establishes the foundation of basic accessibility knowledge. Instructors of advanced courses can subsequently build upon this foundation, enhancing the depth of students' understanding. Future research endeavors should focus on providing guidance regarding which accessibility concepts and teaching approaches are most appropriate for students at various stages in their computer science education.

Finally, a recent study has showcased the feasibility and positive impact of incorporating inclusive design principles across a CS curriculum, leading to increased retention of students from underrepresented groups [14]. Similarly, we believe that the integration of accessibility topics throughout the CS curriculum has the potential to enhance students' learning outcomes and foster a more inclusive educational environment.

5.4 Limitations and Future Work

Despite the valuable insights gained from our study, several limitations warrant consideration. First, the study primarily relied on self-reported data from participants, which may introduce response bias. For instance, students who volunteered to participate in the interview tended to have existing interest in accessibility, and we observed strong consensus about the importance of including accessibility in computing among faculty participants.

The study's focus on a single term did not fully capture the long-term impact of integrating accessibility assignments into core CS courses. The study's scope was limited to a specific set of CS courses and instructors, which may not fully represent the broader landscape of CS education. Future research should consider these limitations and aim to provide a more comprehensive understanding of the integration of accessibility assignments in CS education.

Two of our assignments centered around accessibility concepts for blind and low vision users (*e.g.*, screen readers and Braille), while the third encompassed accessibility concepts relevant to blind and low vision users amongst other groups. Our choice to integrate these concepts was influenced by the widespread availability of relevant tools such as screen readers, which are included in all operating systems, and the prevalence of assistive technology for blind and low vision people in literature [27, 41]. Future assignments should be developed to cover a wider range of disabilities.

Lastly, one key element of our approach was applying our own subject matter expertise to creating assignments and working with instructors up front to ensure alignment and feasibility. We acknowledge that this particular effort is both fundamental and a limited resource. Establishing a community of practice may address instructors' training needs to adapt existing accessibility teaching materials, and potentially create their own accessibility materials in the future [46]. More research is needed to investigate best practices with training instructors to integrate accessibility in computing education broadly.

6 CONCLUSION

In conclusion, our study underscores the importance of integrating accessibility content into undergraduate CS courses. We developed and implemented accessibility assignments within core CS courses without compromising the learning of computing concepts. Our empirical evaluation, which included ten interviews with CS instructors, fifteen interviews with their students, and 249 matched survey responses, revealed significant improvements in students' familiarity with accessibility concepts and their confidence in implementing them. However, our data also highlighted ongoing challenges in fostering sustained interest and mindset change among students. To fully embrace inclusivity in technology, we must continue refining our strategies for incorporating accessibility assignments into core CS courses and collaborating with instructors to ensure that accessibility becomes a fundamental component of CS education. Ultimately, our efforts aim to empower future technology professionals to design and develop technology that is accessible to all, thereby advancing a more inclusive world.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Award No. 2121606, 2121428, 2121549, and the Henry Luce Foundation - Clare Boothe Luce Fund. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] ACM SIGCSE. 2023. ACM Special Interest Group on Computer Science Education. Retrieved September 14, 2023 from https://www.sigcse.org/

[2] ACM/IEEE-CS/AAAI. 2023. CS2023 – ACM/IEEE-CS/AAAI Computer Science Curricula. Retrieved September 14, 2023 from https://csed.acm.org/

[3] Catherine Baker, Yasmine Elglaly, and Kristen Shinohara. 2020. *A Systematic Analysis of Accessibility in Computing Education Research*. https://doi.org/10.1145/3328778.3366843

[4] Kimberly Edginton Bigelow. Designing for Success: Developing Engineers Who Consider Universal Design Principles.

[5] Paul Ryan Bohman. Teaching Accessibility and Design-For-All in the Information and Communication Technology Curriculum: Three Case Studies of Universities in the United States, England, and Austria.

[6] Jim A. Carter and David W. Fourney. 2007. Techniques to assist in developing accessibility engineers. In *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility* (Assets '07), 123–130. https://doi.org/10.1145/1296843.1296865

[7] Kathy Charmaz. 2006. *Constructing grounded theory*. Sage Publications, London; Thousand Oaks, Calif.

[8] Paula Conn, Taylor Gotfrid, Qiwen Zhao, Rachel Celestine, Vaishnavi Mande, Kristen Shinohara, Stephanie Ludi, and Matt Huenerfauth. 2020. Understanding the Motivations of Final-year Computing Undergraduates for Considering Accessibility. *ACM Transactions on Computing Education* 20, 2: 1–22. https://doi.org/10.1145/3381911

[9] Samantha Jane Dobesh, Tyler Miller, Pax Newman, Yudong Liu, and Yasmine N. Elglaly. 2023. Towards Machine Learning Fairness Education in a Natural Language Processing Course. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1* (SIGCSE 2023), 312–318. https://doi.org/10.1145/3545945.3569802

[10] Yasmine N. El-Glaly. 2020. Teaching Accessibility to Software Engineering Students. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 121–127. https://doi.org/10.1145/3328778.3366914

[11] Yasmine N. Elglaly, Catherine M. Baker, Anne Spencer Ross, and Kristen Shinohara. 2024. Beyond HCI: The Need for Accessibility Across the CS Curriculum. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2024)*. https://doi.org/10.1145/3626252.3630788

[12] Yasmine El-Glaly, Weishi Shi, Samuel Malachowsky, Qi Yu, and Daniel E. Krutz. 2020. Presenting and evaluating the impact of experiential learning in computing accessibility education. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering Education and Training*, 49–60. https://doi.org/10.1145/3377814.3381710

[13] Mexhid Ferati and Bahtijar Vogel. 2020. Accessibility in Web Development Courses: A Case Study. *Informatics* 7, 1: 8. https://doi.org/10.3390/informatics7010008

[14] Rosalinda Garcia, Patricia Morreale, Lara Letaw, Amreeta Chatterjee, Pankati Patel, Sarah Yang, Isaac Tijerina Escobar, Geraldine Jimena Noa, and Margaret Burnett. 2023. "Regular" CS × Inclusive Design = Smarter Students and Greater Diversity. *ACM Transactions on Computing Education* 23, 3: 34:1-34:35. https://doi.org/10.1145/3603535

[15] Ed Gellenbeck. 2005. Integrating accessibility into the computer science curriculum. *Journal of Computing Sciences in Colleges* 21, 1: 267–273.

[16] Christopher Lynnly Hovey, Lecia Barker, and Vaughan Nagy. 2019. Survey Results on Why CS Faculty Adopt New Teaching Practices. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (SIGCSE '19), 483–489. https://doi.org/10.1145/3287324.3287420

[17] W3C Web Accessibility Initiative (WAI). 2022. WAI-ARIA Overview. *Web Accessibility Initiative (WAI)*. Retrieved September 14, 2023 from https://www.w3.org/WAI/standards-guidelines/aria/

[18] W3C Web Accessibility Initiative (WAI). 2023. WCAG 2 Overview. *Web Accessibility Initiative (WAI)*. Retrieved September 14, 2023 from https://www.w3.org/WAI/standards-guidelines/wcag/

[19] Lin Jia, Yasmine N. Elglaly, Catherine M. Baker, and Kristen Shinohara. 2021. Infusing Accessibility into Programming Courses. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–6. https://doi.org/10.1145/3411763.3451625

[20] Christos Katsanos, Nikolaos Tselios, Athanasios Tsakoumis, and Nikolaos Avouris. 2012. Learning about web accessibility: A project based tool-mediated approach. *Education and Information Technologies* 17, 1: 79–94. https://doi.org/10.1007/s10639-010-9145-5

[21] Saba Kawas, Laura Vonessen, and Amy J. Ko. 2019. Teaching Accessibility: A Design Exploration of Faculty Professional Development at Scale. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 983–989. https://doi.org/10.1145/3287324.3287399

[22] Claire Kearney-Volpe, Devorah Kletenik, Kate Sonka, Deborah Sturm, and Amy Hurst. 2019. Evaluating Instructor Strategy and Student Learning Through Digital Accessibility Course Enhancements. In *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility* (ASSETS '19), 377–388. https://doi.org/10.1145/3308561.3353795

[23] Sri H. Kurniawan, Sonia Arteaga, and Roberto Manduchi. 2010. A general education course on universal access, disability, technology and society. In *Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility*, 11–18. https://doi.org/10.1145/1878803.1878808

[24] Richard E Ladner, Stephanie Ludi, and Robert J Domanski. Teaching about Accessibility in Computer Science Education (DRAFT 1). Retrieved from https://csed.acm.org/wp-content/uploads/2023/03/Accessibility-First-Draft.pdf

[25] Claudia Loitsch, Gerhard Weber, and Jens Voegler. 2016. Teaching Accessibility with Personas. In *Computers Helping People with Special Needs* (Lecture Notes in Computer Science), 453–460. https://doi.org/10.1007/978-3-319-41264-1_62

[26] Stephanie Ludi. 2007. Introducing Accessibility Requirements through External Stakeholder Utilization in an Undergraduate Requirements Engineering Course. In *29th International Conference on Software Engineering (ICSE'07)*, 736–743. https://doi.org/10.1109/ICSE.2007.46

[27] Kelly Mack, Emma McDonnell, Dhruv Jain, Lucy Lu Wang, Jon E. Froehlich, and Leah Findlater. 2021. What Do We Mean by "Accessibility Research"? A Literature Survey of Accessibility Papers in CHI and ASSETS from 1994 to 2019. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (CHI '21), 1–18. https://doi.org/10.1145/3411764.3445412

[28] Lilu Martin, Catherine Baker, Kristen Shinohara, and Yasmine Elglaly. 2022. *The Landscape of Accessibility Skill Set in the Software Industry Positions*. https://doi.org/10.1145/3517428.3550389

[29] Israel Martin-Escalona, Francisco Barcelo-Arroyo, and Enrica Zola. 2013. The introduction of a topic on accessibility in several engineering degrees. In *2013 IEEE Global Engineering Education Conference (EDUCON)*, 656–663. https://doi.org/10.1109/EduCon.2013.6530177

[30] Rafael Molina-Carmona, Rosana Satorre-Cuerda, Carlos Villagrá-Arnedo, and Patricia Compañ-Rosique. 2017. Training Socially Responsible Engineers by Developing Accessible Video Games. In *Learning and Collaboration Technologies. Technology in Education* (Lecture Notes in Computer Science), 182–201. https://doi.org/10.1007/978-3-319-58515-4_15

[31] Nidhi Rajendra Palan, Vicki L. Hanson, Matt Huenerfauth, and Stephanie Ludi. 2017. Teaching Inclusive Thinking in Undergraduate Computing. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*, 399–400. https://doi.org/10.1145/3132525.3134808

[32] Maulishree Pandey and Tao Dong. 2023. Blending Accessibility in UI Framework Documentation to Build Awareness. In *Proceedings of the 25th International ACM SIGACCESS Conference on Computers and Accessibility* (ASSETS '23), 1–12. https://doi.org/10.1145/3597638.3608380

[33] Rohan Patel, Pedro Breton, Catherine M. Baker, Yasmine N. El-Glaly, and Kristen Shinohara. 2020. Why Software is Not Accessible: Technology Professionals' Perspectives and Challenges. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems* (CHI EA '20), 1–9. https://doi.org/10.1145/3334480.3383103

[34] Helen Petrie and Alistair Edwards. 2006. Inclusive design and assistive technology as part of the HCI curriculum.

[35] G. Michael Poor, Laura M. Leventhal, Julie Barnes, Duke R. Hutchings, Paul Albee, and Laura Campbell. 2012. No User Left Behind: Including Accessibility in Student Projects and the Impact on CS Students' Attitudes. *ACM Transactions on Computing Education* 12, 2: 1–22. https://doi.org/10.1145/2160547.2160548

[36] Cynthia Putnam, Maria Dahman, Emma Rose, Jinghui Cheng, and Glenn Bradford. 2016. Best Practices for Teaching Accessibility in University Classrooms: Cultivating Awareness, Understanding, and Appreciation for Diverse Users. *ACM Transactions on Accessible Computing* 8, 4: 1–26. https://doi.org/10.1145/2831424

[37] Kristen Shinohara, Cynthia L. Bennett, Wanda Pratt, and Jacob O. Wobbrock. 2018. Tenets for Social Accessibility: Towards Humanizing Disabled People in Design. *ACM Transactions on Accessible Computing* 11, 1: 1–31. https://doi.org/10.1145/3178855

[38] Kristen Shinohara, Saba Kawas, Amy J. Ko, and Richard E. Ladner. 2018. Who Teaches Accessibility?: A Survey of U.S. Computing Faculty. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, 197–202. https://doi.org/10.1145/3159450.3159484

[39] Teach Access. 2023. Accessibility Skills Hiring Toolkit. Retrieved December 12, 2023 from https://teachaccess.org/resources/accessibility-skills-hiring-toolkit/

[40] Teach Access. 2023. Teach Access. Retrieved September 14, 2023 from https://teachaccess.org/

[41] Yong-Joon Thoo, Maximiliano Jeanneret Medina, Jon E. Froehlich, Nicolas Ruffieux, and Denis Lalanne. 2023. A Large-Scale Mixed-Methods Analysis of Blind and Low-vision Research in ACM and IEEE. In *Proceedings of the 25th International ACM SIGACCESS Conference on Computers and Accessibility* (ASSETS '23), 1–20. https://doi.org/10.1145/3597638.3608412
[42] Online Braille Translator. 2023. Online English To Braille Translator. *Braille Translator*. Retrieved December 12, 2023 from https://brailletranslator.net
[43] University of Washington. 2023. Home | AccessComputing. Retrieved September 14, 2023 from https://www.washington.edu/accesscomputing/
[44] Annalu Waller, Vicki L. Hanson, and David Sloan. 2009. Including accessibility within and beyond undergraduate computing courses. In *Proceedings of the 11th international ACM SIGACCESS conference on Computers and accessibility*, 155–162. https://doi.org/10.1145/1639642.1639670
[45] Ye Diana Wang. 2012. A holistic and pragmatic approach to teaching web accessibility in an undergraduate web design course. In *Proceedings of the 13th annual conference on Information technology education*, 55–60. https://doi.org/10.1145/2380552.2380568
[46] Etienne Wenger. 1998. Communities of practice: learning as a social system. *Systems Thinker* 9, 5: 2–3.
[47] Zach. 2020. How to Calculate Cohen's d in Excel. *Statology*. Retrieved September 14, 2023 from https://www.statology.org/cohens-d-excel/
[48] Qiwen Zhao, Vaishnavi Mande, Paula Conn, Sedeeq Al-khazraji, Kristen Shinohara, Stephanie Ludi, and Matt Huenerfauth. 2020. Comparison of Methods for Teaching Accessibility in University Computing Courses. In *The 22nd International ACM SIGACCESS Conference on Computers and Accessibility*, 1–12. https://doi.org/10.1145/3373625.3417013

A APPENDICES

We provide two versions of the student survey questions.

A.1 Student Survey Questions (pilot version)

A.1.1 Mindset. Please rate how much you agree with the following statements: (1=strongly disagree, 2=somewhat disagree, 3=neither agree or disagree, 4=somewhat agree, 5=strongly agree)

- Every programmer should have some accessibility knowledge.
- Accessibility is an advanced software feature.

A.1.2 Accessibility Background Knowledge. Please rate how confident you are that you could do each of the following at this time: (1=not at all confident, 2=slightly confident, 3=somewhat confident, 4=fairly confident, 5=completely confident)

1. Define accessibility in terms of its relation to technology and software
2. Give an example of an inclusive or accessible software design
3. Give an example of how accessible or assistive technology is used by people with disabilities
4. Give an example of how assistive technology is used by people with disabilities
5. Give an example of a technological barrier somebody with a disability might face
6. Define the purpose of the Americans with Disabilities Act

A.1.3 Empathy (Understanding People). Please rate how confident you are that you understand the barriers that each of the following groups could face when using technology: (1=not at all confident, 2=slightly confident, 3=somewhat confident, 4=fairly confident, 5=completely confident)

1. People who are blind or low vision
2. People who are deaf or hard of hearing
3. People with autism
4. People with learning disabilities
5. People with intellectual disabilities
6. People with motor or movement disabilities

A.1.4 Accessibility Implementation. Please rate how confident you are that you could do each of the following at this time: (1=not at all confident, 2=slightly confident, 3=somewhat confident, 4=fairly confident, 5=completely confident)

1. Design or develop accessible or assistive technology for people with disabilities.
2. Be able to evaluate accessible or assistive software.
3. Understand legal accessibility regulations (e.g., Section 508, Americans with Disabilities Act, etc.).

A.1.5 Future Interest. Please rate how much you agree with the following statements: (1=strongly disagree, 2=somewhat disagree, 3=neither agree or disagree, 4=somewhat agree, 5=strongly agree)

- I am interested to learn more about designing or developing technologies for and with people with disabilities.
- I plan to consider accessibility in the beginning of future projects.
- I will design and develop future projects with people with disabilities in mind.
- I am interested in pursuing a job or career in accessible technology or a related field.
- [in post-survey only] What was your experience completing the assignment with accessibility concepts in this course? Do you have any feedback about improvements that could be made to this assignment?
- [in post-survey only] How do you intend to apply what you learned about accessibility in your future education, career or personal life?

A.2 Student Survey Questions (main version)

A.2.1 Mindset. Please rate how much you agree with the following statements: (1=strongly disagree, 2=somewhat disagree, 3=neither agree or disagree, 4=somewhat agree, 5=strongly agree)

- Every programmer should have some accessibility knowledge.
- Accessibility is an advanced software feature.
- Accessibility belongs in computer science.
- Accessibility concepts should be taught in computing education.
- Accessibility concepts should be taught in required computer science courses.
- Accessibility concepts should be taught in electives courses.
- Computer science can be applied to help people with disabilities.
- Having accessibility knowledge and skills will help me be more successful in my future job.
- Accessibility content (e.g., assistive technologies such as screen readers or accessibility guidelines) is appropriate for this course.
- Accessibility concepts should not be taught in required computer science courses.
- Please rank the following topics based on their importance to your computer science degree: security, cloud computing, machine learning, data science, accessibility

A.2.2 Accessibility Background Knowledge. Please rate how confident you are that you could do each of the following at this time: (1=not at all confident, 2=slightly confident, 3=somewhat confident, 4=fairly confident, 5=completely confident)

1. Define accessibility in terms of its relation to technology and software.
2. Give an example of an inclusive or accessible software design.
3. Give an example of how accessible or assistive technology is used by people with disabilities.
4. Give an example of a technological barrier somebody with a disability might face.

Please rate how familiar you are with the following at this time: (1=not at all familiar, 2=slightly familiar, 3=somewhat familiar, 4=fairly familiar, 5=completely familiar)

1. How familiar are you with accessibility for blind or low vision people?
2. How familiar are you with accessibility for deaf or hard of hearing people?
3. How familiar are you with accessibility for people with motor and mobility impairments?
4. How familiar are you with accessibility for people with learning disabilities and cognitive impairments?
5. How familiar are you with [existing accessibility guidelines (e.g., WCAG) / digital Braille / augmentative and alternative communication devices]?

A.2.3 Accessibility Implementation. Please rate how confident you are that you could do each of the following at this time: (1=not at all confident, 2=slightly confident, 3=somewhat confident, 4=fairly confident, 5=completely confident)

1. Design or develop accessible or assistive technology for people with disabilities.
2. Be able to evaluate accessible or assistive software.

A.2.4 Future Interest. Please rate how much you agree with the following statements: (1=strongly disagree, 2=somewhat disagree, 3=neither agree or disagree, 4=somewhat agree, 5=strongly agree)

1. I am interested to learn more about designing or developing technologies for and with people with disabilities.
2. I plan to consider accessibility in the beginning of future projects.
3. I will design and develop future projects with people with disabilities in mind.
4. I am interested in pursuing a job or career in accessible technology or a related field.
5. 5. [in post-survey only] What was your experience completing the assignment with accessibility concepts in this course? Do you have any feedback about improvements that could be made to this assignment?