Introducing Accessibility to High School Students

Brian Kelly
bk7031.rit.edu
Rochester Institute of Technology
Rochester, New York

Yasmine El-Glaly
elglaly@wwu.edu
Western Washington University
Bellingham, Washington

ABSTRACT

We designed and implemented an online module that explains accessibility to non-major high school students. The module demonstrates one of the important Web Content Accessibility Guidelines, known as Reflow, using hands-on experience. Reflow is a particularly essential feature for the accessibility of webpages for people with cognitive impairments. The module consists of an interactive lecture and engaging quizzes. We held an online one-hour workshop where the activity was presented to 52 high school students. We collected objective and subjective data about students' knowledge and learning experience using 3 questionnaires. Our findings indicate that the module is effective in raising students' awareness of disabilities, leveraging their knowledge on accessibility, and maintaining high interest in pursuing a computing degree.

CCS CONCEPTS

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

KEYWORDS

accessibility, cognitive impairment, high school, education, reflow

ACM Reference Format:
Brian Kelly and Yasmine El-Glaly. 2021. Introducing Accessibility to High School Students. In *SIGCSE '21: 52nd ACM Technical Symposium on Computer Science Education, March 17–20, 2021, Toronto, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3408877.3432466

1 INTRODUCTION

There is a high demand for computing professionals and it is expected to grow by 16% over the next 8 years [5]. There exist several outreach programs and courses that aim to inform and teach high school students about fundamental computer science principles [4] and increase their interest in computer science [15]. Most of these programs focused on introducing computational thinking to K-12 education [19, 25]. However, introducing accessible computing to K-12 education has not been addressed to the best of our knowledge. Accessible computing is a fundamental computing concept and a solid aspect of computing ethics [13, 16]. Teaching accessibility is of high importance at the college level and there are several courses and programs that cover the topic of accessibility [3, 18, 22]. Kawas et al. found that more educators can teach accessibility if enough teaching resources are made available [12, 17].

We designed an interactive module that introduces accessibility to early computing learners, e.g. high school students, or first-year college students. We present in this paper the third version of the module, and the lessons we learned from previous offerings. The module includes a lecture in the form of slides and a hands-on activity. Both the slides and the activity are hosted on GitHub[1], and are accompanied by quizzes that were created using google forms[2]. The activity was built using HTML and CSS. The pedagogical design of this accessibility module is based on active learning. Active Learning is built out of short course-related activities that all the students of the class are allowed to engage in [10]. We chose the topic of reflow. Reflow, as outlined by the Web Content Accessibility Guidelines (WCAG) 2.0, is how the content fits on a page [24]. The intent behind the reflow guideline is to make sure those with vision impairments can enlarge or shrink text while still being able to read and make out the content [24]. Beyond site impairments, however, the success criterion is also intended to avoid horizontal reading or cutting off content from the viewport which would significantly increase the effort required to read [24]. With a higher level of effort required to make out the content, people with cognitive impairments will have a harder time making out the content. We balanced hands-on activities keeping in mind that students do not necessarily know how to program. We refrained from kinesthetic activities as it is less engaging for high school students [9].

The accessibility module was used in a synchronous online workshop, where 52 students participated. Using objective and subjective assessment, we found that students learned about accessibility, and maintained a high interest in pursuing a computing degree. The contributions of this work are threefold:

(1) We present an accessibility module that is designed to meet the educational needs of high school students.
(2) This module was used in a workshop attended by 52 high school students. By analyzing the collected data using pre and post-questionnaires, we found that the results indicate the effectiveness of this module.
(3) All teaching materials discussed in the paper, e.g. lecture, activity, quizzes, etc. are made publicly available online. These resources can help other instructors to infuse accessibility into their courses.

2 RELATED WORK

While there is a fair amount of research in the past relating to pedagogy in the field of computing science [6, 7, 14, 20]. The idea behind

[1] https://github.com/Teaching-Accessibility/Reflow
[2] https://www.google.com/forms/about/

SIGCSE '21, March 17–20, 2021, Toronto, Canada

ACM ISBN 978-1-4503-8062-1/21/03…$15.00
https://doi.org/10.1145/3408877.3432466

these papers is how to improve pedagogy with computing students, and we intended to look at better ways to teach accessibility to computing students. From these papers, we derived that students appear to react better to active learning approaches. Active learning, a focus of our pedagogical approach, has appeared to be a reliable way to distribute information to students and to keep them engaged [11, 21]. As mentioned previously, active learning is built out of short course-related activities that all the students of the class are allowed to engage in [10]. The best process that we could equate to this line of thinking is something akin to teaching modules. To the best of our knowledge, there has been some research in the field of accessibility modules like the one implemented in the following papers [8, 12, 13]. These papers made use of accessibility teaching modules with statistically significant success. However, the aforementioned papers are not aimed towards high school students, but instead towards college-aged students. This paper intends on demonstrating how to implement a module and the results of such with high school students.

3 ACCESSIBILITY MODULE

The accessibility module consists of two main components: 1) The lecture; meant to break down the complex concepts into bite-sized pieces and 2) The hands-on activity; meant to give the students the opportunity to experience the accessibility problem in effect, reverse engineer the problem, and finally fix it.

3.1 Lecture

What is engineering? –> What is software engineering? –> Apps for All (accessibility) –> Reflow as example

The module lecture started with a breakdown of what it means to engineer software and what even is engineering. As our target demographic was high-school students who may or may not have a full understanding of software engineering a wide net was cast. From there, we gave the students a lightning breakdown of engineering disciplines (IE Civil, Mechanical, and Industrial Engineering) and a brief look into the engineering life-cycle. For the last part of the wide net component of our lecture, we moved on to the more specific topic of what software engineering is all about through alliteration; Define, Develop, Design, and Deliver. With the basics covered, we moved onto the highly specified topic of reflow. The intent behind moving straight into the topic of reflow was broken into two components. The first, to get a basic awareness of building accessible software, and the second was to illustrate how software can be used to solve a problem and that there are still plenty of problems that need inventive young developers to fix. We chose the topic of reflow due to its relevance to impairments such as Attention-Deficit/Hyperactivity Disorder(ADHD) given the presumption that the students may know someone with this impairment and be able to empathize [1]. ADHD is a disorder marked by continuous trends of inattention or hyper activity[2]. Reflow of a page may lead users with ADHD and similar disorders to lose track of elements or move too quickly through elements [1]. At this part of the lecture, the students took part in the first part of the hands-on activity. They were asked to visit an inaccessible website and attempt to navigate it. More details on the activity are described in the next section. After the first part of the activity, we then lectured the students on what exactly they were seeing. We demonstrated that the problems they were running into were based on reflow errors and proceeded to define reflow and give the WCAG 2.0 standards [1] that accompany the reflow definition. Post defining reflow, we used real-life websites as examples of good and bad reflow design. Now that the students have a definition for reflow, and examples in their heads, we gave them the tools to find and fix reflow errors. The tools in question are the inherent tools of most modern web browsers otherwise known as the inspection tools. The students were then informed of who exactly this was helping; specifically those with cognitive impairments. The lecture ended with a breakdown of what a cognitive impairment is, and a slide of further resources.

3.2 Hands-On Activity

1- Demonstration

For the first part of the activity, students were given a dummy website, called Zapado, that appeared to have an abundance of white space and an odd flow. Figure 1a exemplifies the site with improper reflow. We intentionally designed and built this website to demonstrate the negative impact of improper reflow. The website is hosted on GitHub and is publicly available [3]. The students were given a "Can you tell?" quiz to see if they could identify miscellaneous elements of the page. The questions are summarized in Table 1. From our perspective, we knew that half the page was pushed out of, if not entirely, the viewport making some of the answers nearly impossible to know. The students were instructed to, by any means they see fit, to navigate the website.This provides a solid demonstration of the accessibility error in progress.

Students were then given the accessible version of the website. Figure 1b exemplifies the site corrected with proper reflow. Students were asked to answer the same "Can you tell?" quiz as before to demonstrate that, after fixing the site, the information is now conveyed differently and they have a better understanding of the site they were on. For the sake of humor and student engagement, the site itself is revealed to be somewhat nefarious with its end goal to buy all your personal information as well as your identity.

2- Repairing

After the students have progressed through the lecture, they are now appropriately equipped with the browser inspection tool to find where the problems are that they can fix. We used Chrome as an example for a web browser where inspection tools can be easily activated, e.g. using a shortcut. After the inspection tool window appears, we will click the overlapped windows icon at the top left corner (surrounded by the red square in Figure 2a). Then, we will see the various viewport options shown in as shown in Figure 2b. By changing the viewport size, we can see how a webpage will look like on devices of different screen sizes. The students are provided the website's script and are allowed to solve the problem for themselves however they see fit. There is a corrected version of the website's script that the students could access to reverse engineer the fixes.

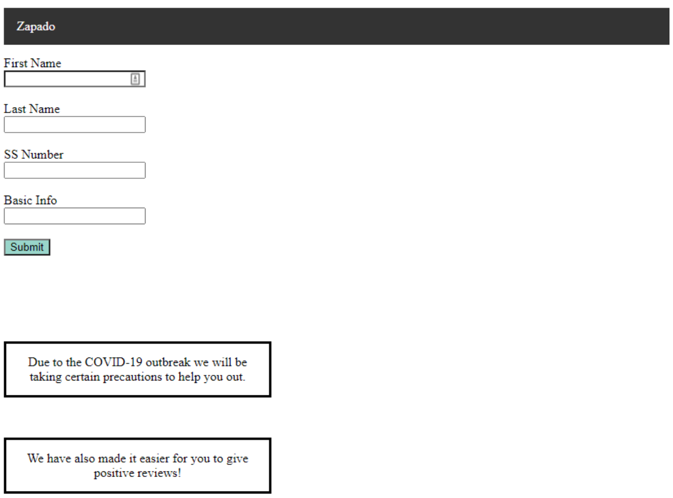
4 ONLINE WORKSHOP

The Rochester Institute of Technology (RIT) offers annual workshops for high school students to learn about the various majors.

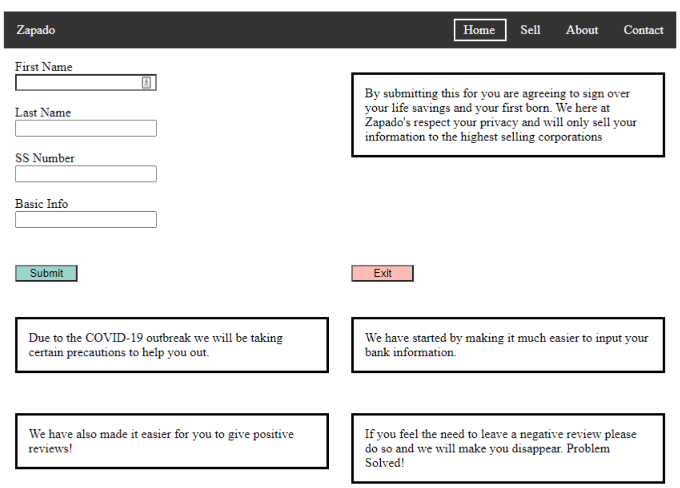
[3]https://github.com/Teaching-Accessibility/Reflow

"Can You Tell?" Quiz Questions	Can be found in the inaccessible website?	Can be found in the accessible website?
1) What is the website's name?	Yes	Yes
2) Which page are you on in the website?	No	Yes
3) What are you doing, e.g. shopping, selling, etc.?	No	Yes
4) Are they helping you out due to COVID-19?	Yes	Yes
5) If yes to the previous, in what ways are they helping out?	No	Yes
6) Are they making all reviews easier to post?	No	Yes

Table 1: The Can You Tell quiz given to students after visiting the Zapado website.

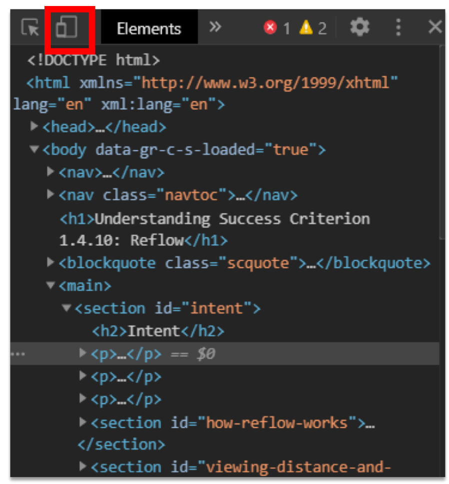


(a) A dummy website called Zapado with no-reflow.

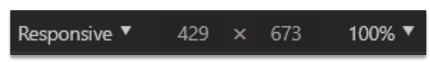


(b) The Zapado website with reflow.

Figure 1: The hands-on activity that shows the same website without reflow (top) and with reflow implemented (bottom).



(a) Chrome's inspection tool.



(b) Chrome's viewport options.

Figure 2: A screenshot of the web browser inspection tool that was used to demonstrate how reflow can be tested for.

This year, 2021, all workshops were online and free. We used the *Accessibility Module* in the Software Engineering workshop. High school students who are interested in learning about the Software Engineering major self-registered in the workshop. There were no elimination or selection criteria, and prior knowledge of programming was not expected. Students were expected to have access to a computer or a phone and internet service to join the workshop, which was hosted through Zoom [4]. The workshop was one hour long and followed the sequence of events shown in Figure 3. The instructors used the chat window to distribute links to the students (e.g., quizzes links). Students were encouraged to interact by texting in the chat window as well. The number of high school students who attended the workshop is 52 (Female: n=12, Non-Binary: n=2).

4.1 Data Collection

We hypothesized that the accessibility module will raise students' awareness on the topic of accessibility (H1). We also hypothesized

[4]https://zoom.us/

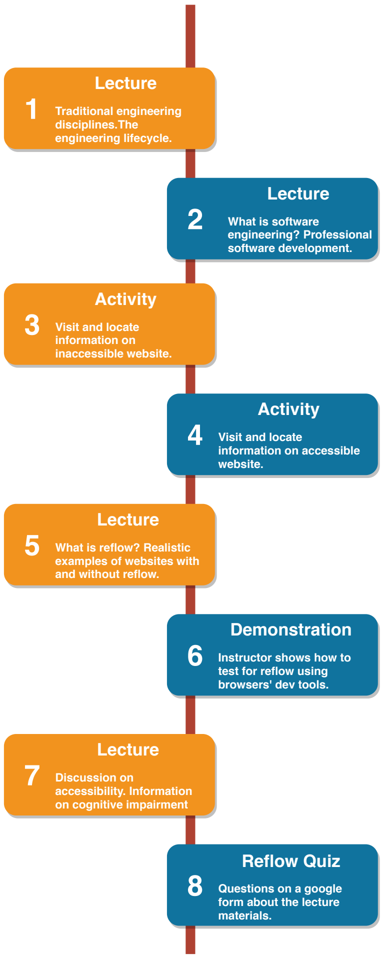


Figure 3: The order by which the accessibility module was taught in the workshop.

that the accessibility module will leverage students' interest in pursuing a computing-related degree (H2). We collected data to assess both hypotheses. We asked the students to fill a pre-questionnaire at the beginning of the workshop, and a post-questionnaire at the end of the workshop. Additionally, we asked the students to answer a quiz on the topic of Reflow right after the lecture material was delivered. The pre and post questionnaires included the following questions: 1) Rate the statement "I am interested in a computing degree." on a scale from 1 to 5; 2) How do you define accessibility?; and 3) List two or three different types of disabilities. We instructed the students to use their wordings in answering the open-ended questions and refrain from looking up the definition of accessibility online. The post questionnaire included two open-ended questions seeking students' subjective feedback on the workshop and their learning experience.

The Reflow quiz contained 8 multiple-choice questions about the information explained during the lecture portion of the workshop. Questions covered information on reflow as an example of web content accessibility guidelines, and cognitive impairment as an example of the main beneficiary of web accessibility. Each question had 4 options, where only one option is the correct answer. Students were asked to rely on their memory while answering the questions, and not to search for answers online. The multiple-choice questions with two of the provided choices are shown in Table 2.

5 FINDINGS

We grouped our findings into three categories; knowledge transfer, interest in computing, and overall learning experience.

1- Learning about accessibility (increased)

To assess the first hypothesis, we analyzed the data we collected from the pre and post questionnaires and the reflow quiz. We compared students' answers about their perception of accessibility and disabilities. We also conducted a descriptive statistical analysis of students' performance in the reflow quiz.

Types of disabilities: In the pre-questionnaire, when students asked to give examples of disabilities, most of the answers covered physical disabilities, e.g., blindness and deafness. Forty-two percent of students did ***not*** mention any form of invisible disabilities, i.e. learning disabilities or cognitive impairment. This percentage has dropped to 13% in the post questionnaire as more students mentioned invisible disabilities such as dyslexia. This implies that students became more aware of forms of disabilities other than physical disabilities.

Definition of accessibility: Accessibility can be defined as the design and development of technologies that allow people with disabilities to interact with the web [23]. We did not expect students to know the accessibility definition before the lecture, but we wanted to learn about their awareness and how it changed after the workshop. In the pre-questionnaire, most of the answers did not come close to the accessibility definition and we marked 78.8% of the answers as wrong. Examples of answers that we considered wrong:

"*When it is very easy to reach someone.*"

"*The rate and speed at which things are available.*"

Examples of answers that we considered correct:

"*Accessibility is when something is accessible to those with disabilities.*"

"*Something that allows people with disabilities to use something to the same extent as anyone else.*"

We acknowledge that these definitions are either recursive (i.e., accessibility is when something is accessible), or vague (i.e., something allows .. to use something ..). However, as the answers referred to the support of people with disabilities, we counted them as correct. In the post-questionnaire, we only marked answers as correct if they include a reference to a digital solution. Examples of correct answers in the post-questionnaire:

"*Features of software that allows people with disabilities to experience the software the same as anyone else.*"

"*Accessibility is providing equitable solutions (ex. Reflow websites) so that those with disabilities can perform just the same as [people*

Reflow Quiz Questions	Choices
1) What is a cognitive impairment ?	(a) Something that inhibits a user's ability to perform one or more cerebral tasks. (correct) (b) A physical, mental, or behavioral enhancement of cerebral tasks. (wrong)
2) What forms can cognitive impairments take?	(a) Physical, mental, or behavioral (correct) (b) Mental or emotional (wrong)
3) What is reflow?	(a) The way browsers and document readers organize content to fit the width of the viewport (correct) (b) The ability to arrange pictures on a page to align with various viewports (wrong)
4) What is another name for reflow?	(a) Responsive Web Design (correct) (b) Developmental Design (wrong)
5) According to WCAG 2.0, what is the highest zoom that interfaces should be designed for?	(a) 400% (correct) (b) 4000% (wrong)
6) According to WCAG 2.0, should you hide content when switching from desktop to mobile views?	(a) No, the same content should be accessible on all viewports, some elements may need resizing (correct) (b) Yes, on smaller viewports there is no need to see some information (wrong)
7) How can the inspection tool help us as developers, in the case of reflow?	(a) It allows us to see our interfaces in different viewports (correct) (b) It allows us to generate a new interface for each viewport (wrong)
8) Who does this help?	(a) Everyone (correct) (b) Only those with cognitive impairments (wrong)

Table 2: The quiz given after the Reflow lecture is made of multiple choice questions. The table shows the correct answer and a sample of the incorrect options.

without disabilities.]"
We found that the percentage of correct answers in the post-questionnaire has flipped, compared to the pre-questionnaire results, and 82% of the answers were correct. Students' answers included wording that is closer to the W3C definition such as navigation, usability, participation, ability, and ease of access.

Reflow: The maximum possible score in the Reflow quiz is 8. There was not a particular question that was answered more incorrectly than the others. Most questions were answered incorrectly by 3 to 4 students. Even the last question asking "Who does this help?", which we assumed very easy, was answered incorrectly by two students. The highest score is 8 (achieved by 42 students), and the lowest score is 4. Most of the students scored 87.5% or higher, the mean score is 95% with a standard deviation equals to 0.86.

To sum, the collected data indicates that students had a better grasp of the definition of accessibility and different types of disabilities. They also learned some technical details about reflow and cognitive impairment.

2- Interest in a computing major (unchanged)

We compared students' ratings of their interest in pursuing a computing degree in the pre and post questionnaires. We found that the answers in the pre-questionnaire ranged from 2 to 5 ($\mu = 4.3, SD = 0.7$). In the post questionnaire, the ratings ranged from 2 to 5 ($\mu = 4.31, SD = 0.7$). The numbers almost did not change, showing a high interest in studying computing. In the follow-up question asking about the reasoning behind the given rating, we found that the student who rated their interest in computing on the lower side in the post-questionnaire explained that "Microelectronics seems more interesting." We note that students who attended the workshop are already curious about computing and want to learn more about the field. We conclude that the accessibility module helped students to remain interested in the computing field, and having a focus on accessibility in the workshop did not scare the students away or turned off their passion for computing.

3- Learning Experience

Students were asked to describe their learning experience in the post questionnaire and state what they liked and/or disliked about the workshop. Students' answers were mostly positive with 94.7% of the answers were about what they liked. Two students mentioned that they did not like the workshop because of its emphasis on accessibility. Their comments came as follows.
"*Not focused enough on software, too focused on reflow and disabilities.*"
"*I didn't learn too much about the details of Software Engineering; I feel like the workshop would have been more engaging if it was more focused on that.*"

Other students favored the accessibility topic. For example, two students commented on the following.
"*I liked learning about how you can use reflow to improve user experience, especially for people with disabilities.*"
"*It was good to focus on the accessibility aspects of software development.*"
Some students discussed their comfort with the technical details explained in the accessibility module. For example, one student said: "*It was generalized so you didn't need any in-detail knowledge about a topic beforehand for it to be understandable, but if you did have prior knowledge, it would still benefit you.*"

Seventy-one percent (71%) of the students liked the active learning aspect of the accessibility module. Some of their feedback goes as follows.

"*I liked the interactivity and the quizzes. It made it easier to visualize the subjects that were talked about.*"
"*Being able to do a small activity during the meeting was fun and engaging.*"

In sum, students found that the accessibility module is comprehensible and informative. The active learning design of the module in particular has fostered students' engagement and helped them stay focused during the one hour of the workshop.

6 DISCUSSION

During the past three years, we offered workshops for high school and first-year college students to introduce them to accessibility. Students attended these workshops voluntarily and we had collected data from them to gauge their interest in computing in general and in accessibility in particular. These previous workshops were less engaging than the one we reported in this paper and caused frustrations at times to the students. This is despite the fact that previous workshops were in-person, and the current one is online. In this section, we will reflect on the lessons we learned from our previous versions of the accessibility module. We will also highlight the changes we made through these three years and why these changes yielded a better module design.

Mistake#1: Too many technical details.
In the previous workshops, we designed the activity part of the accessibility module so that students will run a mobile app on a computer lab. The code was provided to the students, and they were asked to run the app on Android Studio (already installed in the lab), make few edits to the code, and re-run the code. This sequence appeared to be too advanced for some of the students. When few students get stuck, or machines did not work as expected, it negatively impacts the learning process of the whole group. Moreover, some students felt the workshop activity evolves around programming, and students without prior knowledge feel left out, and mistakenly think that this major is not for them. In the current design, we mitigated this mistake by transitioning from a mobile app to a website. The code is hosted on GitHub, and students can access the website using GitHub pages [5]. By providing the students with the appropriate link, they do not need to compile or run code to visit and navigate the activity website. At the same time, students with programming experience can dive in, access the website code (HTML and CSS), and learn more about the programming behind reflow.

Mistake #2: Complex knowledge.
We had tried to cover accessibility, disabilities, and web content accessibility guidelines (WCAG) in addition to programming in one setting. The result was that students were confused and disengaged. Working with high school or first-year college students requires us to break down complex knowledge. In the current design of the accessibility module, we simplified several concepts. For example, instead of having several slides on WCAG, we explained reflow as a technique that is required for web accessibility. We added that this is an industry-standard among other standards that you can learn more about. We also provided a list of further readings and online resources for those who are interested in learning more.

Mistake #3: Too few examples.
Another issue with previous workshops is that we did not provide examples for accessible and inaccessible software. The only inaccessible software (to them) that they interacted with was the app we built and provided. To mitigate this problem, we added examples for real websites that are publicly available. We demonstrated how reflow was correctly implemented in one website and improperly implemented in the other. In the current module design, we embodied abstract concepts by offering expressive examples and explaining how accessibility can be practically applied and impacts the final product, user experience, and people's life.

Mistake #4: Lack of context.
Previously, we started the lecture by talking about disabilities and the percentage of people with disabilities in the world. We had explained accessibility, without referencing software engineering. Students had left the workshop without fully understanding the connections between accessibility and the computing major. Currently, we start the accessibility module by explaining the big picture, i.e. what is software engineering and its development cycle. We then explain that software engineers should design and build their products so that all people can use them. Hence, students can realize the connection between software engineering and accessibility.

7 CONCLUSION AND FUTURE WORK

Teaching accessibility to high school students is feasible when proper pedagogy methods are employed. We detailed in this paper an accessibility module that incorporated: 1) lecture for direct and fast information dissemination; and 2) hands-on activity for students' engagement and the benefits of active learning. The design of the accessibility module was built based on the lessons learned from previous iterations. The balance in the breadth and depth of the accessibility knowledge and its implementation (reflow as an example) has set this module to success. The current module was shown to be effective in teaching students about accessibility and reflow. As we made the module with all its components available online, instructors can easily use them as is or edit them to better fit their course and teaching style. The benefits of introducing accessibility to young students include 1) More students may grow interested in computing, especially those who assume computing has no human factor; 2) More computing students will be aware of their professional responsibility early in their education, which can have a positive lasting impact on their work ethics. In the future, we plan on creating more accessibility modules that avoid the pitfalls we discussed in this paper. We also plan on disseminating these teaching materials among high school teachers and first-year computing course professors.

ACKNOWLEDGMENTS

This material is based upon work supported by the United States National Science Foundation under grant #1825023.

[5]https://teaching-accessibility.github.io/ReflowCC/Inaccessible/

REFERENCES

[1] 2017. Diverse Abilities and Barriers. https://www.w3.org/WAI/people-use-web/abilities-barriers/#cognitive.

[2] 2020. NIMH » Attention-Deficit/Hyperactivity Disorder. https://www.nimh.nih.gov/health/topics/attention-deficit-hyperactivity-disorder-adhd/index.shtml

[3] Catherine M Baker, Yasmine N El-Glaly, and Kristen Shinohara. 2020. A systematic analysis of accessibility in computing education research. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. 107–113.

[4] Lenore Blum and Thomas J Cortina. 2007. CS4HS: an outreach program for high school CS teachers. *ACM SIGCSE Bulletin* 39, 1 (2007), 19–23.

[5] U.S. Department of Labor. Bureau of Labor Statistics. 2018. Occupational Outlook Handbook. https://www.bls.gov/ooh/computer-and-information-technology/computer-and-information-research-scientists.htm.

[6] Tom Crick and Sue Sentance. 2011. Computing at school: stimulating computing education in the UK. In *Proceedings of the 11th Koli Calling International Conference on Computing Education Research (Koli Calling '11)*. Association for Computing Machinery, New York, NY, USA, 122–123. https://doi.org/10.1145/2094131.2094158

[7] Valentina Dagienė. 2011. Informatics Education for New Millennium Learners. In *Informatics in Schools. Contributing to 21st Century Education (Lecture Notes in Computer Science)*, Ivan Kalaš and Roland T. Mittermeir (Eds.). Springer, Berlin, Heidelberg, 9–20. https://doi.org/10.1007/978-3-642-24722-4_2

[8] 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* (Seoul, South Korea) *(ICSE-SEET '20)*. Association for Computing Machinery, New York, NY, USA, 49–60. https://doi.org/10.1145/3377814.3381710

[9] Yvon Feaster, Luke Segars, Sally K. Wahba, and Jason O. Hallstrom. 2011. Teaching CS unplugged in the high school (with limited success). In *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education - ITiCSE '11*. ACM Press, Darmstadt, Germany, 248. https://doi.org/10.1145/1999747.1999817

[10] Richard M Felder and Rebecca Brent. 2009. Active Learning: An Introduction. (2009), 7.

[11] Qiang Hao, Bradley Barnes, and Mengguo Jing. 2020. Quantifying the effects of active learning environments: separating physical learning classrooms from pedagogical approaches. *Learning Environments Research* (June 2020). https://doi.org/10.1007/s10984-020-09320-3

[12] Saba Kawas, Laura Vonessen, and Andrew 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*. ACM, Minneapolis MN USA, 983–989. https://doi.org/10.1145/3287324.3287399

[13] Stephanie Ludi, Matt Huenerfauth, Vicki Hanson, Nidhi Rajendra Palan, and Paula Garcia. 2018. Teaching Inclusive Thinking to Undergraduate Students in Computing Programs. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. ACM, Baltimore Maryland USA, 717–722. https://doi.org/10.1145/3159450.3159512

[14] Linda Mannila, Valentina Dagiene, Barbara Demo, Natasa Grgurina, Claudio Mirolo, Lennart Rolandsson, and Amber Settle. 2014. Computational Thinking in K-9 Education. In *Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference (ITiCSE-WGR '14)*. Association for Computing Machinery, New York, NY, USA, 1–29. https://doi.org/10.1145/2713609.2713610

[15] Chad Mano, Vicki Allan, and Donald Cooley. 2010. Effective in-class activities for middle school outreach programs. In *2010 IEEE Frontiers in Education Conference (FIE)*. IEEE, F2E–1.

[16] 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*. ACM, Baltimore Maryland USA, 399–400. https://doi.org/10.1145/3132525.3134808

[17] 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*. 1–9.

[18] 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 (TACCESS)* 8, 4 (2016), 1–26.

[19] Jake A Qualls and Linda B Sherrell. 2010. Why computational thinking should be integrated into the curriculum. *Journal of Computing Sciences in Colleges* 25, 5 (2010), 66–71.

[20] L. Rolandsson. 2013. Changing Computer Programming Education: The Dinosaur That Survived in School: An Explorative Study about Educational Issues Based on Teachers' Beliefs and Curriculum Development in Secondary School. In *2013 Learning and Teaching in Computing and Engineering*. 220–223. https://doi.org/10.1109/LaTiCE.2013.47

[21] Kate Sanders, Jonas Boustedt, Anna Eckerdal, Robert McCartney, and Carol Zander. 2017. Folk Pedagogy: Nobody Doesn't Like Active Learning. In *Proceedings of the 2017 ACM Conference on International Computing Education Research* (Tacoma, Washington, USA) *(ICER '17)*. Association for Computing Machinery, New York, NY, USA, 145–154. https://doi.org/10.1145/3105726.3106192

[22] Kristen Shinohara, Saba Kawas, Andrew J Ko, and Richard E Ladner. 2018. Who teaches accessibility? A survey of US computing faculty. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. 197–202.

[23] W3C. 2019. Introduction to Web Accessibility. https://www.w3.org/WAI/fundamentals/accessibility-intro/.

[24] W3C. 2020. Web Content Accessibility Guidelines (WCAG). https://www.w3.org/WAI/standards-guidelines/wcag/.

[25] Aman Yadav, Sarah Gretter, Jon Good, and Tamika McLean. 2017. Computational thinking in teacher education. In *Emerging research, practice, and policy on computational thinking*. Springer, 205–220.