# ALL: Accessibility Learning Labs for Computing Accessibility Education

Saad Khan
Rochester Institute of Technology
Rochester, NY, USA
sk6786@rit.edu

Heather Moses
Rochester Institute of Technology
Rochester, NY, USA
hlm8500@rit.edu

Samuel Malachowsky
Rochester Institute of Technology
Rochester, NY, USA
samvse@rit.edu

Daniel Krutz
Rochester Institute of Technology
Rochester, NY, USA
dxkvse@rit.edu

## ABSTRACT

Our *Accessibility Learning Labs* not only inform participants about the need for accessible software, but also how to properly create and implement accessible software. These experiential browser-based labs enable participants, instructors and practitioners to engage in our material using only their browser. In the following document, we will provide a brief overview of our labs, how they may be adopted, and some of their preliminary results. Complete project material is publicly available on our project website: http://all.rit.edu

## CCS CONCEPTS

• **Human-centered computing** → *Accessibility*; • **Social and professional topics** → *CS1*; *Software engineering education.*

## KEYWORDS

Accessibility Education, Computing Education, Computing Accessibility

**ACM Reference Format:**

Saad Khan, Heather Moses, Samuel Malachowsky, and Daniel Krutz. 2021. ALL: Accessibility Learning Labs for Computing Accessibility Education. In *26th ACM Conference on Innovation and Technology in Computer Science Education V. 2 (ITiCSE 2021), June 26–July 1, 2021, Virtual Event, Germany.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3456565.3460028

## 1 INTRODUCTION

To fill the existing void in accessibility education, we have created *a comprehensive collection of laboratory activities that are essential to accessibility education.* Together, these labs are referred to as the *Accessibility Learning Labs* (ALL). The primary goals of these educational accessibility labs are to increase student awareness of the need to create accessible software and to inform students about fundamental accessibility concepts. Each lab contains comprehensive and straightforward material on a specific accessibility topic. Thus, the labs are able to be easily integrated into a variety of existing introductory computing courses (*e.g.,* Computer Science I & II) due to their easy-to-adopt nature. No special software is required to use the labs since they are web-based and run on any computer with a reasonably modern web browser.

Each lab addresses one accessibility issue and contains: I) Relevant background information on the examined issue, II) An activity to emulate this accessibility problem as closely as possible, III) Details about how to repair the problem from a technical perspective, and IV) Information from actual people about how this accessibility issue has impacted their life.

The 'Tips, Techniques, and Courseware Session' will provide participants with a mechanism to I) Learn about our provided educational accessibility material, II) Gain an understanding of how they may adopt and benefit from the material, and III) Understand some of our preliminary results regarding the effectiveness of the material. No prior experience from tutorial participants will be required.

### 1.1 Lab Availability

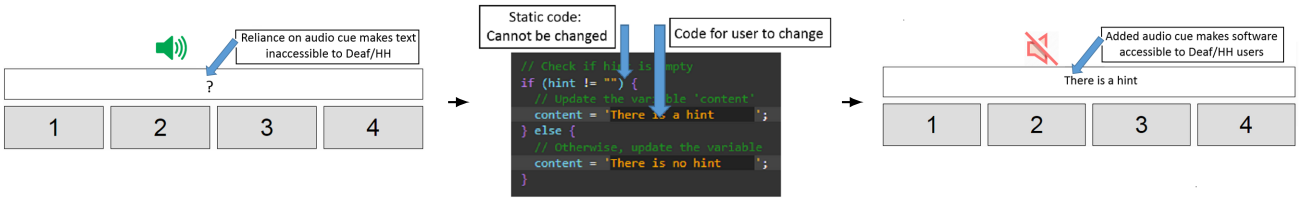
Users require only an internet connection and a web browser (Safari, Chrome, Edge, Firefox, Opera, etc.) for adoption. Complete lab material, including lecture slides, videos, quiz, and activities, is publicly available on our project website: http://all.rit.edu

### 1.2 Results of Pedagogical Analysis

We evaluated the labs in ten sections of a CS2 course at our university, with 321 students participating [1]. Our primary findings include: I) The labs are an effective way to inform participants about essential topics in creating accessible software II) The labs demonstrate the potential benefits of our proposed experiential learning format in motivating participants about the importance of creating accessible software III) The labs demonstrate that empathy


*ITiCSE 2021, June 26–July 1, 2021, Virtual Event, Germany*

ACM ISBN 978-1-4503-8397-4/21/06.
https://doi.org/10.1145/3456565.3460028




**(a) Pictured above is a game where the user must choose the correct box (numbered 1-4). Hints are available via the box with the question mark, signified by an audio cue. This software is inaccessible since the user cannot hear the cue and the visual message is not relevant.**

**(b) Students make code adjustments via a mock IDE used through a browser. The hint box label is updated to indicate whether a hint is available or not so that the user does not rely solely on an audio cue.**

**(c) The software is made more accessible to deaf and hard-of-hearing users by the student adding an informative visual message.**

**Figure 1: Example of student repairing accessibility problem using simulated environment**

material increases learning retention. For example, as demonstrated in Figure 2, the created material (Group B and Group C) resulted in higher post-activity quiz scores in relation to the baseline.

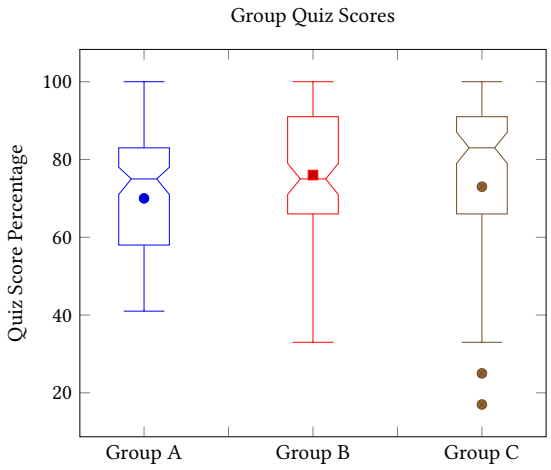



**Figure 2: Student quiz scores for the three evaluation groups.**

## 2 LAB STRUCTURE

The instructional activities utilize an interactive structure, enabling participants to not only learn about essential concepts in accessible software, but also to experience the implications of inaccessible software and repair the inaccessible software. An example lab activity is represented in Figure 1, where in Figure 1a the participant experiences the software as a disabled user would. Figure 1b demonstrates the repair process for the participant and Figure 1c demonstrates how the participant made the software more accessible. These phases are described in further detail in the steps below.

(1) **Participant learns about foundational accessibility issue:** The participant is first provided appropriate background information about the specific accessibility issue. The objective for this 3-5 minute phase is to provide a fundamental understanding of the accessibility issue and the challenges encountered by people with this disability when using inaccessible software. For example, our first lab focuses on accessibility for deaf/hard-of-hearing users. The first section of that lab provides information on the deaf/hard-of-hearing population and the challenges they face.

(2) **Participant experiences accessibility issue:** Participants are tasked with performing a simple yet unique activity in each lab. After completing the activity, the participant repeats the activity, but with an *accessibility emulation* feature. For example, our 'Deaf/Hard of Hearing' lab activity instructs the participant to select a specific icon when provided with an audio notification, which, for a hearing user, would be a trivial task. To emulate what a deaf/hard-of-hearing user may experience, the audio notification is not played, making the activity much more difficult (Figure 1a). While it is impossible to entirely emulate the experiences of a user with disabilities, the objective of this phase is to emulate their experiences as closely as possible.

(3) **Participant repairs accessibility issue:** Using a simulated, web-based development environment, the participant repairs the inaccessible aspect of the software. For example, in the 'Deaf/Hard of Hearing' lab, the repair is made by adding a visual cue to accompany the audio cue (Figure 1b).

(4) **Participant experiences the benefits/impact of their repair:** Lastly, the participant repeats the activity with their repairs incorporated into the software (Figure 1c). Optimally, with the enacted repairs, the software should be more accessible, even with the accessibility emulation feature active. If they are not satisfied with their repairs, the participant may conduct the previous step of repairing the software again.

## Acknowledgements

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

## REFERENCES

[1] Weishi Shi, Saad Khan, Yasmine El-Glaly, Samuel Malachowsky, Qi Yu, and Daniel E. Krutz. 2020. Experiential Learning in Computing Accessibility Education. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Companion Proceedings (ICSE '20)*. Association for Computing Machinery, New York, NY, USA, 250–251. https://doi.org/10.1145/3377812.3390901