Debugging for Inclusivity in Online CS Courseware: Does it Work?

Amreeta Chatterjee
Oregon State University
Corvallis, Oregon, USA
chattera@oregonstate.edu

Rudrajit Choudhuri
Oregon State University
Corvallis, Oregon, USA
choudhru@oregonstate.edu

Mrinmoy Sarkar
Techno International New Town
Kolkata, West Bengal, India
mrin2ksarkar@gmail.com

Soumiki Chattopadhyay
Oregon State University
Corvallis, Oregon, USA
liudy@oregonstate.edu

Dylan Liu
Oregon State University
Corvallis, Oregon, USA
liudy@oregonstate.edu

Samarendra Hedaoo
Oregon State University
Corvallis, Oregon, USA
liudy@oregonstate.edu

Margaret Burnett
Oregon State University
Corvallis, Oregon, USA
burnett@eecs.oregonstate.edu

Anita Sarma
Oregon State University
Corvallis, Oregon, USA
anita.sarma@oregonstate.edu

ABSTRACT

Online computer science (CS) courses have broadened access to CS education, yet inclusivity barriers persist for minoritized groups in these courses. One problem that recent research has shown is that often inclusivity biases ("inclusivity bugs") lurk within the course materials themselves, disproportionately disadvantaging minoritized students. To address this issue, we investigated how a faculty member can use AID—an Automated Inclusivity Detector tool—to remove such inclusivity bugs from a large online CS1 (Intro CS) course and what is the impact of the resulting inclusivity fixes on the students' experiences. To enable this evaluation, we first needed to (Bugs): investigate inclusivity challenges students face in 5 online CS courses; (Build): build decision rules to capture these challenges in courseware ("inclusivity bugs") and implement them in the AID tool; (Faculty): investigate how the faculty member followed up on the inclusivity bugs that AID reported; and (Students): investigate how the faculty member's changes impacted students' experiences via a before-vs-after qualitative study with CS students. Our results from (Bugs) revealed 39 inclusivity challenges spanning courseware components from the syllabus to assignments. After implementing the rules in the tool (Build), our results from (Faculty) revealed how the faculty member treated AID more as a "peer" than an authority in deciding whether and how to fix the bugs. Finally, the study results with (Students) revealed that students found the after-fix courseware more approachable - feeling less overwhelmed and more in control in contrast to the before-fix version where they constantly felt overwhelmed, often seeking external assistance to understand course content.

ICER '24 Vol. 1, August 13–15, 2024, Melbourne, VIC, Australia
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0475-8/24/08
https://doi.org/10.1145/3632620.3671117

CCS CONCEPTS

• **Human-centered computing → Field studies**; • **Applied computing → E-learning**; • **Social and professional topics → Adult education**; **Gender**.

KEYWORDS

Online CS Education, GenderMag, Inclusivity Bugs, Automated Checker

ACM Reference Format:
Amreeta Chatterjee, Rudrajit Choudhuri, Mrinmoy Sarkar, Soumiki Chattopadhyay, Dylan Liu, Samarendra Hedaoo, Margaret Burnett, and Anita Sarma. 2024. Debugging for Inclusivity in Online CS Courseware: Does it Work?. In *ACM Conference on International Computing Education Research V.1 (ICER '24 Vol. 1), August 13–15, 2024, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3632620.3671117

1 INTRODUCTION

Much has been discussed about the potential of online computer science (CS) courses to revolutionize education by expanding access to a broad range of students and drawing in a large number of individuals eager to learn [2, 42, 51]. For example, during fall 2021, 30% of U.S. post-secondary students were taking at least one online course and 60% were enrolled in entirely online programs [49]. While these courses have indeed broadened access globally, the lack of diversity in computing education has remained remarkably persistent [67].

Particularly for online courses in the field of CS, studies show that most learners tend to be young adult men who already possess at least a bachelor's degree [8, 28, 39]. Whereas, women in online CS education face othering[53], have lower retention [31, 36] and have reported feeling less motivated and less technologically capable [69]. Understanding the importance of closing this observed gender participation gap, many programs pursue initiatives aimed at fostering diversity and inclusion [21, 27, 42, 60]. However, despite the continued efforts by the computing education community, little improvement is observed [30].

Research has shown that even tools and technologies are a contributing factor to the gender disparity [48]. Our recent study [14] has brought to light the presence of "inclusivity bugs" in online CS courseware (the digital course materials) due to its lack of support for cognitive diversity in the very technology/infrastructure used by students. Online CS faculty found inclusivity bugs in their own courseware, such as terminology being unfriendly to newcomers, failure to guide students towards understanding the benefits, and highlighted how repeatedly encountering barriers can demotivate, or disadvantage students from minoritized groups. They used the GenderMag method, an inspection technique that identifies where, what type of users would disproportionately run into barriers, through the lens of five cognitive styles (or facets) [10].

These types of inclusivity bugs matter because when the courseware itself is not inclusive of different cognitive styles, it burdens students with a cognitive "tax" each time they interact with the courseware. This emphasizes the importance of "debugging" online CS courseware in order to create truly inclusive learning environments for all students, particularly those who are underserved by the tools themselves. However, so far, we have learnt about potential inclusivity issues from the instructors' perspectives across a limited set of use cases [14]. To effectively debug online CS courseware, we need a deeper insight into the actual lived experiences of students. Where exactly are they encountering inclusivity barriers? What specific challenges are they facing on the ground?

With this motivation, our empirical investigation took a student-centered approach to allow us to have a holistic understanding of the inclusivity challenges they encountered. In the first stage, we investigated:
RQ1 (Bugs): What inclusivity challenges do students encounter in online CS courses?

Next, to systematically analyze CS courseware, we used the results of RQ1 to design a set of decision rules corresponding to these challenges and developed new capabilities for AID, the automated inclusivity detector [14] to pinpoint "inclusivity bugs" in the courseware. This would allow us to know "where" students would run into inclusivity challenges and why. Subsequently, we teamed up with a faculty member, who was interested in improving the inclusivity of their courseware by using the tool to remove inclusivity bugs from a large online CS1 course. This enabled us to investigate:
RQ2: (Build + Faculty): How can faculty use AID to identify and remove inclusivity bugs embedded within their course?

Finally, we conducted an online user study with students to investigate whether the changes the faculty made actually decreased the inclusivity bugs students encountered through our third research question:
RQ3 (Students): How do faculty's inclusivity fixes impact students' experience with the course?

As the first investigation into the impacts of removing inclusivity bugs within online CS courseware using an automated tool, our findings can inform new automated approaches for making online CS education more inclusive for all learners, especially those from minoritized groups.

2 BACKGROUND & RELATED WORK

2.1 Inclusivity in online CS education

Several factors have been shown to contribute to underrepresentation of certain populations in tech fields. In online CS communities, LGBTQ+ programmers anticipated that, because of the heterosexist climate, few women and LGBTQ+ people would join [20]. Similar problems exist in online CS education too. Studies have shown that women often face "othering" in online learning [53], are less persistent with lectures and assessments [36], have lower retention earlier in CS Massive Open Online Courses [19], and are less likely than others to complete an online career change program for CS [31]. Krause-Levy et al. reported that, when universities shifted to online learning as a result of COVID-19, first-generation and women students felt a lack of sense of belonging [38]. Research has also shown gender differences in student experiences with CS learning platforms, such as Stack Overflow and Piazza [4, 22, 63, 65]. However, only a few previous studies have looked into how to *improve* inclusivity in CS courses, and even fewer in *online* CS courses.

Studies of in-person CS courses have reported that people-oriented tasks and creative expression improve inclusion of women [5, 15, 44]. Pair programming [68, 70], meaningful or socially relevant assignments [6, 9, 43], and leveling the playing field with mechanisms like having everyone start with a language new to all [37] are just a few of the well-known practices increasing recruitment and/or retention across genders in in-person CS education. In online CS, researchers have reported that including gender-inclusive elements in course presentation, improving representation, and using neutral visual designs improve experiences for women in CS [35].

Toward creating inclusive courseware, a number of organizations have created general standards. The Online Learning Consortium's five Pillars of Quality Online Education is used by institutions to identify ways to support successful online learning, including access for all [16]. Another well known set of standards is Quality Matters, a set of 50 standards specifically for online and blended higher education courses [46]. This set includes standards for accessibility and usability, but is intended for course designers and requires a membership to access [45]. However, there still remains a gap to see if debugging inclusivity bugs in courseware would make it inclusive.

2.2 GenderMag and its application in CS education

The GenderMag method [10] is an evidence-based inspection approach designed to find, fix, and avert gender-based cognitive "inclusivity bugs" in technology. GenderMag is grounded in extensive research on how users of different genders tend to interact differently with technology, using different cognitive styles (or, cognitive "facets" in GenderMag). The five facets used in GenderMag are *motivations* for using tech; *information processing style*; *computer self-efficacy*; *learning by process vs. by tinkering*; and *attitude toward risk*.

The "inclusivity bugs" identified by GenderMag are instances of technology product fails. In these instances, the product fails to support these five facets' values, disproportionately affecting people with certain cognitive styles. They are also gender-inclusivity bugs because the facets reflect (statistical) gender differences in problem-solving approaches [3, 10–12, 62, 66].

GenderMag uses three personas: Abi (Abigail/Abishek), Pat (Patricia/Patrick), and Tim (Timara/Timothy). Each of the five facets has a range of values; Abi's and Tim's values lie at opposite ends, and Pat has values within. Since Tim and Abi represent the two extremes of the spectrum, if tech software addresses both these ends, it will also cater to those like Pat, who exhibit a mix of traits from both Abi and Tim. The Abi persona represents facet values which disproportionately skew towards women, Tim represents facet values that disproportionately skew towards men, and Pat provides a third set of values as described below [10]:

- Motivations: Abi and Pat are motivated to use tech only as needed for their task. They rarely have spare time and prefer familiar features so they can focus on the task. Tim is motivated to investigate new, cutting-edge features.
- Information Processing Style: Abi and Pat gather relevant information comprehensively before acting. Tim likes to delve into the first option and pursue it, backtracking if needed.
- Attitude Towards Risk: Abi and Pat are risk-averse with tech. They may avoid using features with an unknown time cost and risks. Tim is risk-tolerant so may use unknown features.
- Computer Self-efficacy: Abi has lower computer self-efficacy, so if a problem arises when they are trying to use an unfamiliar feature, they blame themselves and stop using the tech. Pat has medium self-efficacy and tries alternative ways of succeeding for a while. Tim has higher computer self-efficacy, so if a problem arises, they'll blame the tech, and may spend extra time finding a solution.
- Learning by Process vs. by Tinkering: Abi is a process-oriented learner, so prefers to proceed through tasks step-by-step. Tim and Pat learn by tinkering, and therefore prefer not to be constrained by rigid, pre-determined processes.

A portion of the Abi persona is shown in Figure 1.

The output of a GenderMag evaluation is a set of inclusivity bugs—usability bugs that disproportionately impact users who share the same facets as those of the persona used in the evaluation session. The facets statistically cluster by gender, with Abi's facets statistically more common among women [10]. For instance, a study involving men and women using a search product revealed that women's action failure rates were over twice as high as men's. However, after the product's gender-inclusivity bugs were fixed, failure rates of both the participating genders went down and the difference between these two genders' failure rates disappeared [66].

In the field of CS education, Garcia et al. [26] introduced an inclusive design curriculum to enable faculty to teach GenderMag concepts incrementally across the undergraduate computer science curriculum. Additionally, Garcia et al. [25] conducted an action research study where 13 CS faculty integrated GenderMag concepts into 44 computer science and information technology course offerings in an undergraduate curriculum, finding that it improved course outcomes, particularly for marginalized student groups. Chatterjee et al. [14] carried out a field study that revealed the existence of inclusivity bugs in online CS courseware when evaluated through the lens of GenderMag personas. They developed the AID/Courseware tool to automatically detect a subset of such bugs in courseware. Oleson et al. [50] identified 11 elements of inclusive pedagogical content knowledge (PCK) for teaching GenderMag concepts in courses through their action research investigation. Letaw et al. [41] utilized some of Oleson et al.'s PCK elements when incorporating GenderMag content into two junior-level online CS courses, fostering feelings of inclusion among students and increasing their likelihood of completing the CS major. However, there is still a gap in our understanding within the realm of online CS courses regarding whether students' experiences with courseware are enhanced by addressing inclusivity issues.

2.3 Automated checkers

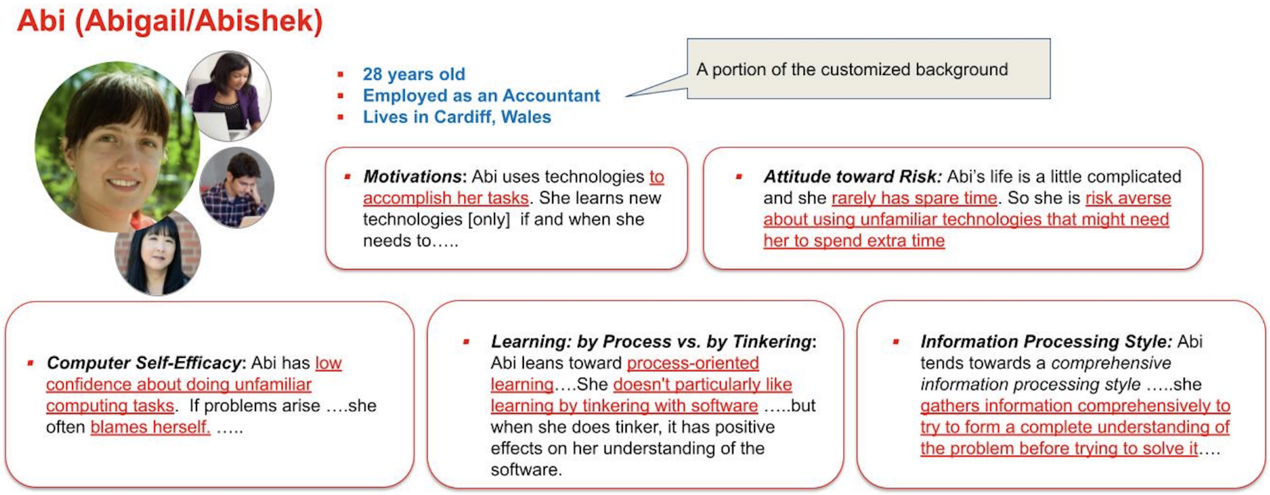
There are tools that automatically check for usability bugs, such as the WebTANGO prototype [32], which predicts user behavior and page navigation time based on complexity. Dingli et al. [17] also proposed a tool for website evaluation using usability guidelines. Various tools also aid accessible design, such as WAVE [33] checks webpage accessibility for visually impaired users, Vischeck [18] provides low-vision simulation, and AATT [34] tests for WCAG conformance. Ally [55] supports content creators in enhancing the accessibility of PDF files.

There are only a few existing tools that support inclusive design practices for cognitive diversity. The GenderMag Recorder's Assistant [47] is a note-taking tool that helps organize outputs like answers to Cognitive Walkthrough questions, notes, and screenshots from GenderMag sessions, aiding humans in conducting the method. Chatterjee et al. [13] investigated automating the GenderMag method to capture inclusivity bugs in open source software.

Most closely related to our work is AID/Courseware, a tool that automates one facet of the GenderMag method from Abi's perspective: information processing style [14] and checks for three decision rules on courseware pages. The first decision rule considers scenarios in which users like Abi might not be able to find all of the information they require to complete their task. The second and third rules relate to website navigation. The former checks if links are labeled with a keyword or phrase and the latter determines whether a destination page provides cues to help Abi understand that they have arrived at the correct location after clicking on a link. However, this tool only automates one of the five GenderMag facets. In our study, we build upon AID/Courseware by expanding the tool's functionalities to address all five of GenderMag's facets from Abi's perspective. In the rest of the paper, we refer to this tool as AID.

3 POSITIONALITY STATEMENT

Understanding researcher positionality is essential to demystify our lens on data collection and analysis [24, 58]. We situate this paper in North America in the 21st century, writing as authors who primarily work as academic researchers. We are of multiple races (Asian, White), with national/ethnic backgrounds from Asian and North American nations. Several of us also have the intersectional identity of women of color. As such, a number of us have experienced lack of representation in computing courses firsthand. At the same time, we recognize the privileges we hold as individuals with access to higher education. Two of us have inclusivity leadership positions,



Abi (Abigail/Abishek)

- 28 years old
- Employed as an Accountant
- Lives in Cardiff, Wales

A portion of the customized background

- *Motivations*: Abi uses technologies to accomplish her tasks. She learns new technologies [only] if and when she needs to…..

- *Attitude toward Risk:* Abi's life is a little complicated and she rarely has spare time. So she is risk averse about using unfamiliar technologies that might need her to spend extra time

- *Computer Self-Efficacy*: Abi has low confidence about doing unfamiliar computing tasks. If problems arise ….she often blames herself. …..

- *Learning: by Process vs. by Tinkering*: Abi leans toward process-oriented learning….She doesn't particularly like learning by tinkering with software …..but when she does tinker, it has positive effects on her understanding of the software.

- *Information Processing Style:* Abi tends towards a *comprehensive information processing style* …..she gathers information comprehensively to try to form a complete understanding of the problem before trying to solve it….

Figure 1: Portions of the GenderMag Abi persona and the facets. The blue portions of the persona are customizable

granting us credibility when collaborating with faculty members. We are committed to using these privileges to holistically enhance the educational experience for students in computing fields.

4 RQ1(BUGS): WHAT INCLUSIVITY CHALLENGES DO STUDENTS ENCOUNTER IN ONLINE CS COURSES?

4.1 Method

Data Collection: To answer RQ1, we collected student discussions of online CS courses posted on the Ed discussion board on Canvas [1], a Learning Management System. The Ed discussion board serves as a platform for students to discuss their learning, ask and answer questions, share challenges, give opinions, and post comments on discussions posted by other participating students.

We reached out to 14 faculty of online CS courses at University <X>, a large public research university in North America. Five out of the 14 courses' faculty granted permission: (a) introduction to Object Oriented Programming (CS2), (b) junior-level introductory software engineering (SE1) course covering requirements, design, and project management, (c) another junior-level introductory software engineering (SE2) course covering testing and debugging, (d) junior-level algorithms (Algs.), (e) junior-level CS Data Structures (DS). The instructors obtained consent from students to use their posts by emailing them, resulting in 126 posts. All data was collected during the 2022-2023 academic year.

Data Analysis: To find what inclusivity challenges students faced, two authors inductively open-coded discussion posts. We organized these posts by the week of the term they were posted in and the specific part of the courseware they referred to (the syllabus, weekly readings, assignments, or exams) and removed posts which were clarification requests, or general comments. For the remaining posts, we analyzed each one to determine (a) how students characterized by GenderMag's Abi facet types might be disproportionately impacted, and (b) whether it mapped to any of the known inclusivity barriers [14] in online CS courseware, as discussed below. We iteratively grouped the findings into categories, until reaching team consensus.

For consistency, we measured inter-rater reliability in line with recommended practices in qualitative research [61]. Specifically, two authors mapped 20% of the discussions each, to the categories and compared their outputs using the Jaccard index [40] which resulted in 100% agreement. Given this level of consensus [40], two of the authors then divided and coded the remaining posts.

4.2 Results (RQ1)

In this section, we first identify the challenges students are discussing in the posts and map them to the six inclusivity barrier types identified by Chatterjee et al. [14], along with the reasons of *why* those barriers occurred for the Abi persona (GenderMag facets). Table 1 presents 39 inclusivity challenges students faced in their courses, grouped into four distinct categories. These are then mapped to the corresponding facets (Column 3), inclusivity barriers (Column 4) and the courses where students experienced these challenges (Column 5).

Category 1: Instances where courseware lacked clarity regarding assignments' next steps were classified as: *lack of step-by-step guidance*. This inclusivity challenge was seen the most (22 out of 39 instances) and was pervasive (spanning across all five courses). This is a challenge since it does not support Abi-like students' process-oriented learning style (LS) and attitude towards risk (RA) - resulting in several situations where students lacked a clear direction of where to focus, and not have enough information to proceed with assignments. For example, in assignment requirements in SE1, the faculty had meant for students to incorporate microservices in the upcoming assignment, not the current one. However, having all the task information needed for the entire project upfront caused the student anxiety about potentially wasting time on a task they wouldn't be able to complete anyway (RA).

[1] https://www.instructure.com/canvas

Table 1: Categories of inclusivity challenges identified from Ed discussion posts. Each category is mapped to corresponding inclusivity barriers [14] and the GenderMag facets (M: Motivations, RA: Risk Averseness, CSE: Computer Self Efficacy, LS: Learning style by process vs. tinkering, IP: Information Processing)

Categories of Inclusivity Challenges	#Count	Facets	Inclusivity Barriers [14]	Courses We Analysed				
				CS2	Algs	DS	SE1	SE2
Lack of step-by-step guidance	22	LS	2 ("focus"), 4("missing information"), 6("blocked by information")	✓	✓	✓	✓	✓
Lack of clear purpose	9	M, RA	2 ("focus"), 4 ("missing information"), 5 ("benefit/pitfall")		✓		✓	✓
No connection to prior learning	3	LS, CSE	5 ("benefit/pitfall"), 6("blocked by information")		✓		✓	
No cues to needed information	5	IP, CSE	1 ("deadend"), 3 ("newcomer-unfriendly"), 4("missing information"), 6("blocked by information")		✓		✓	

> [Ed-105, Assignment 6] "I am getting ready to start some work on my MVP, but I am feeling nervous that I will get going only to realize that incorporating microservices will not be possible."
> <Facet(s): Learning by Process vs. Tinkering, Attitude Towards Risk>

This highlights how ambiguity within courseware can breed uncertainty for process-oriented and risk-averse students.

Category 2: Students expressed feeling lost when tasks did not have a clear objective or purpose. Abi-like students engage with technology primarily to complete specific tasks, rather than for the enjoyment of technology itself - they prefer to concentrate on tasks that are of interest to them. In Ed-120, a student expressed confusion about the aim of the project, questioning whether their efforts to choose a topic will be relevant or aligned with the course's objectives:

> [Ed-120, Assignment 3] "What is supposed to be our aim for our projects...I'm sort of lost on what we are supposed to actually be planning...unsure if any one of them will be relevant or are aligned with what we are doing in this course."
> <Facet(s): Motivations>

This highlights that when students do not understand the relevance of an assignment / tasks to the overall course goals, it can lead to confusion. This also aligns with the principles of Quality Matters which emphasize the importance of clearly linking course materials to learning outcomes [46]. Situations like these, where courseware *lacked clear purpose*, were categorized as an inclusivity challenge because task-oriented students like Abi were unsupported.

Category 3: Students struggled with finding sufficient explanations when the provided course materials did not cover essential concepts. The absence of pointers to already covered course material further exacerbated this challenge. Without proper guidance towards resources, Abi-like students might give up (since the tasks are unexplained and might be a waste of time) and Tims might waste their time tinkering and risk attempting unintended approaches. For example:

> [Ed-6, Assignment 3] "So I tried looking for [information] from the material we've covered, and I didn't find how to calculate...[complexity]...could I get an explanation...in case I just managed to miss it in the modules? Also, [I] ran across the same issue with trying to figure out [another approach]."
> <Facet(s): Learning by Process vs. by Tinkering>

Such instances were categorized as: *assignments did not guide the student towards (atleast one) prior learning.*

Category 4: A major hurdle for students was being blocked by information – too much, not enough / vague, and inconsistently worded duplicate information. Discussion posts revealed that students encountered obstacles in their courseware stemming from ambiguous information and unclear terminology. We classified such instances as *no clear cues to guide students towards necessary information*; A student, for instance, sought clarification on the expectations regarding source citations in discussion posts, as it was not clearly defined:

> [Ed-19, Discussions] "The Week 1 Discussion overview says we need to include citations in our responses to our peers. Would you please elaborate on what this means? Are we supposed to cite third-party content as part of our responses? How many citations do we need per post? How do we indicate we're citing something (e.g. footnote, etc.)?"
> <Facet(s): Information processing style>

5 RQ2 (BUILD + FACULTY): HOW CAN FACULTY USE AID TO IDENTIFY AND REMOVE INCLUSIVITY BUGS EMBEDDED WITHIN THEIR COURSE?

5.1 (BUILD)ing on Automated Inclusivity Detector (AID)

We saw from Section 4.2 that students encountered inclusivity challenges that could disproportionately impact and disadvantage students with Abi-like facet values (See Section 2).

GenderMag enables faculty to systematically identify such challenges in their courses. However, it can be expensive, as it is a manual process and labor-intensive. For example, the creators of GenderMag recommend that at least three evaluators spend one to two hours per session, where a session typically covers only one to three use cases. To reduce the cost of conducting GenderMag, in an earlier study, we built the Automated Inclusivity Detector (AID) [13, 14], a tool that can automatically find inclusivity bugs for the information processing style facet. AID uses decision rules to find these bugs (which maps to student challenges of category 4). Therefore, to capture the instances where students could face inclusivity challenges, we created four decision rules, corresponding to the four inclusivity challenges students faced.

(DR-LS): High level instructions should be followed by step-by-step instructions (Category 1).

(DR-M, RA): Tasks should have a clear purpose (Category 2).

(DR-LS, CSE): Assignments should mention at least one prior learning resource (Category 3).

(DR-IP, CSE): Provide explicit cues to guide students (Category 4).

AID could not capture the first three categories of inclusivity bugs that arise when the remaining four GenderMag facets are not adequately supported. Therefore, we built on top of AID to encompass the facets that were previously not covered. Table 2 shows six decision rules that AID currently checks for, the facet they represent, and inclusivity barriers they capture. The blue highlighted portion details the new decision rules we integrated into AID.

We implemented these rules using natural language processing (NLP) methods, leveraging large language models (LLM) [2]. Specifically, we leveraged Meta Llama2 [64], an open source language model, to have control over the prompt [3] format and specifications, thus being able to specify the behavior of the model. We first scraped all the data available in the courseware to build a corpus of web pages. Next, we used a combination of NLP techniques, such as lexical analysis [56], and part-of-speech tagging [1] to preprocess the scraped pages and pass it on to the language model. We implemented the rules as follows:

(DR-LS) High level instructions should be followed by step-by-step instructions *<Facets: Learning by Process vs. by Tinkering>*: While brief instruction or overviews may suit some learners like Tim, who are comfortable with tinkering and exploring independently, the inclusion of sub-instructions supports learners like Abi, who prefer a step-by-step learning approach. To capture this, we start by first checking if a course webpage has any instructions at all. If there is only a single instruction on the webpage, it was marked as a problem. In cases where multiple instructions were listed, we looked at each paragraph individually and if a paragraph had only one instruction, we considered this to be a violation of the rule (having just one instruction per paragraph might not provide the granular, step-by-step guidance needed to support Abi's learning process effectively).

To determine if a sentence on the webpage is an instruction or not, we used few-shot prompting with Llama2 [64]. Few-shot prompting is a technique where we provide illustrative examples in the prompt to steer the language model to better contextualized performance (in this case to predict whether a sentence is an instruction or not). However, before applying this approach to AID, we conducted a preliminary assessment of Llama2's ability to identify sentences as instructions. We compiled a test set (instructions from assignments) and compared Llama2's classifications on this data. Llama2 achieved a 97% F1-score [4], indicating its high accuracy in recognizing instructional content.

(DR-M, RA) Tasks should have a clear purpose *<Facets: Motivations, Attitude Towards Risk>*: Abi, who prioritizes tasks aligned with their interests and relies on technologies, may be disproportionately affected by tasks lacking clear purposes. Explicit purpose behind tasks enables learners like Abi to assess the task's relevance and decide on its usefulness. We checked for this rules only on pages that contained instructions, as classified by Llama2.

To operationalize this rule, we drew insights from information foraging theory [54], which provides a cost-benefit analysis framework for evaluating the information gained from following a cue. From the students' perspective, we wanted to quantify the perceived purpose behind each task by analyzing its costs versus benefits. For the cost aspect, we checked whether the required time commitment was mentioned by searching for words indicative of time or deadlines in a corpus. As for benefits, we looked for two elements - first, whether the learning outcomes were stated, which we checked for by simply searching for the phrase 'learning outcomes' as per the university's course design guidelines. Second, we used Llama2 to detect if any future learning needs motivating the task were mentioned on the page. Towards that, we prompted examples (few-shot prompting) of future learning needs being explicitly mentioned (e.g., "You will need these basic calculus and math concepts for the first two week's material.") from the course. If both cost and benefit aspects were clearly present, we concluded the task did not violate this rule of making the purpose explicit, otherwise the tool reported a violation.

(DR-LS, CSE) Assignments should mention at least one prior learning *<Facets: Learning by Process vs. by Tinkering, Computer Self Efficacy>:* We checked this rule to identify instances where assignments introduce new concepts or tasks without adequately connecting them to prior knowledge or providing sufficient guidance. Such linking of assignment requirements to prior knowledge empowers learners like Abi by providing a structured approach to complete the task, utilizing familiar step-by-step processes from tutorials, how-to videos, and wizards. We checked this rule only for assignment pages in each module of the courseware. Our process began by building a corpus of keywords from the non-assignment pages. We used the RAKE algorithm [57] for automatic keyword extraction to extract the top 10 ranked keywords, according to recommended practices [59]. Next, to find the most common topics

[2] LLMs refer to pre-trained models that use a large amount of data for pre-training and a number of parameters in the architecture, in the order of billions. In this work, we have considered Meta's Llama2 13B chat model, which is open-source [64]

[3] A prompt is what is inputted to the LLM to generate its response. The prompt consists of three different parts: system message, user message, and assistant message. The system message sets the context of the interaction with the LLM. The system message is used for instructing the model about the input format of the data points and ensuring that the output of LLMs conforms to a specific format

[4] Each sample was passed on 10 times, and correct classification was counted if it succeeded each of the times [52]. We did this to account for consistency given the variability of LLM outputs.

Table 2: Decision Rules (DR) driving AID. The three decision rules (highlighted in blue) are the ones we devised from inclusivity challenges identified from student discussion posts

	Decision Rules driving AID	Facets	Barriers Rule Addresses [14]
DR-IP-1	Keywords from subgoals and associated actions should be present on the webpage	IP	1("deadend"), 4("missing information"), 6("blocked by information")
DR-IP-2	Linked pages should contain keywords from link labels	IP	3("newcomer-unfriendly")
DR-IP-3	Links should be labeled with a keyword or phrase	IP	6("blocked by information")
	New Set of Decision Rules Informed by Student Challenges		
DR-LS	High level instructions should be followed by step-by-step instructions	LS	2("focus"), 4("missing information"), 6("blocked by information")
DR-(M, RA)	Tasks should have a clear purpose	M, RA	2("focus"), 4("missing information")
DR-(LS, CSE)	Assignments should mention at least one prior learning	LS, CSE	5("benefit/pitfall"), 4("missing information")

covered in the assignment pages, we extracted the keywords from that page. Finally, we checked for the assignment keywords in the non-assignment keyword corpus that included pages before the assignment. If there were zero overlaps, we concluded that the assignment page was not effectively drawing on prior knowledge.
User interface design: We used the existing design of AID as a foundation and actively sought feedback from faculty members who would be using this tool. Specifically, we went through four rounds of feedback sessions with the faculty. This iterative process allowed us to refine and enhance the tool's design based on the input and suggestions provided by the faculty members. For example, we updated the UI to (1) show the total number of bugs upfront showing the severity of the problem, (2) highlight the text in the courseware where the bug occurred so it can be easily connected to the list, and (3) connect the bug in the list to the specific part of the course text where it occurred. See Figure 2 (a) and Figure 3 (a) for the sample of the UI design.

One of the inherent challenges we faced was the lack of definitive ground truth data to evaluate the effectiveness of AID's output. This was primarily due to the fact that faculty had, at times, proactively addressed the concerns raised by students in the Ed-discussion boards. However, we took a multi-faceted approach, combining team review, faculty validation, and direct student feedback to establish a robust understanding of AID's performance. As a first step, three researchers conducted five rounds of reviews on AID's output for an online CS1 course, examining bugs against each of the three decision rules, all of whom were familiar with the decision rules. AID flagged 17 pages in the course, leading to disagreements on 3 pages among researchers and with the tool. Recognizing the element of subjectivity, and given our 82.35% agreement rate (14 out of 17), we considered it satisfactory as the tool is not meant to function independently. Faculty members will need to interpret these results in light of their experience.

The faculty member responsible for the courseware leveraged AID and critically evaluated its outputs. As shown in Table 3-Column 4, faculty validated that AID pinpointed legitimate concerns that students have had within the course (also detailed in Section 5.2 below). As for students, they indeed encountered the inclusivity bugs that AID had identified, as detailed in Section 6, further validating its results.

5.2 (Faculty) Removing Inclusivity Bugs

Next, we collaborated with a faculty member from University <X> to help them find inclusivity bugs in their online CS1 course [5]. They were familiar with the GenderMag method and the personas, Abi and Tim. They used AID to identify and address inclusivity bugs within their courseware, proceeding to implement their inclusivity improvements autonomously. We observed the faculty member as they interacted with the tool using a think-aloud protocol, which included three one-hour long sessions.

The faculty's process of fixing inclusivity bugs was iterative. For each inclusivity bug that AID identified, they revisited their (1) original course-design rationale, (2) recalled student's feedback/pain points from past classes and then back to (3) how the inclusivity bug would affect Abi (according to AID's output).

Revisiting original course-design rationale: After agreeing with a certain bug, the instructor took a step back to critically examine the decision-making processes that led to the courseware's current design and structure.

> [Faculty] "I have always thought this that [this course] could probably benefit from some kind of...appendix."
>
> [Faculty] "...the [course] designer thought that students would ignore it if it was with the screenshot, and they might feel that the this tutorial is too lengthy, so we would just keep it to the point...but maybe that place is after the tutorial not before."

Recalling students' painpoints: Multiple times during this process, they revisited real student experiences with that portion of the

[5] we wanted to evaluate the impact of inclusivity bugs and their fixes with freshman courses as here is where they face new concepts and traditionally have the highest CS drop off rates. This along with given class schedules in our university, CS1 was the most suitable class.

courseware - allowing them to gain insights into how the issues manifested in real life learning scenarios.

> [Faculty] "...I have met multiple students who kind of were frustrated with the course we did, so we made some changes"

Back to how the inclusivity bug would affect Abi: During this process, the faculty member formed a deeper understanding of the violations reported and the underlying reasons behind them. They progressively gathered an increased understanding of each of the violations, specifically thinking about the (Abi-like) students who are disadvantaged and why.

> [Faculty] "...that entire paragraph talks about what will happen when you clone. Why should you commit and push and do all that stuff? But not how to do it...which is kind of taken for granted by this paragraph....I agree with that."

However, their engagement with the tool was collaborative and not one of blind obedience. Sometimes they thought critically about the fixes and considered different perspectives, such as whether students themselves are the best judges of their learning needs and whether encountering challenges could potentially have positive effects on learning outcomes.

> [Faculty] "...there could be a devil's advocate kind of designer for courses and they would think. So what if they went to YouTube videos? "

They revisited situations where students voiced extreme concerns over their experience with another course - this feedback from students played a crucial role in shaping the faculty's approach.

> [Faculty] "...I have had a student actually cry and tell me that this is. This course was frustrating because I felt like I always needed help."

Given that these were the initial set of fixes based on the tool's output, the faculty chose to first pilot these changes. As a strategic approach, they prioritized fixing the inclusivity bugs that were most significant or impactful, specifically for the materials used in the course's first week. In these pages, the faculty identified a set of 10 bugs that they addressed with a set of four fixes (see Table 3):

Fix1 & Fix2: The first page that was flagged was the Tool SetUp page. AID reported five violations on this page, where a high-level instruction was provided (Figure 2 (a)), but there were no sub-instructions to guide process-oriented learners through the necessary steps. In this case, the faculty removed the set of instructions from the Tool SetUp page and instead added these details *just in time* within the assignment. This provided more contextualized and timely guidance for process-oriented learners, rather than presenting a set of abstract instructions upfront. Figure 2 (b) shows how these two fixes were implemented.

Fix3: The second and third pages (Module 1- Explorations 1 and 2) were flagged with violations, indicating that the purpose behind the (reading/exploration) tasks on those pages were not clear (Figure 3 (a)). For each page, faculty first brought up why these tasks needed to be done (e.g., the quiz will test you). The instructor used the tool's output to explicitly link these readings to the overall course outcomes, as shown in Figure 3 (b). Note that Quality matters[6] says this very specific thing should be done. However, the instructor overlooked to do it in this instance. This highlights the importance of automated checkers for courseware.

Fix4: The third page (Module1-Exploration2) had violations pertaining to high-level instructions that lacked sufficient guidance. To address this violation, the faculty broke up the content from this page in two distinct pages, similar to Fix1 & Fix2 (See Supplemental [23]). The first page was dedicated solely to the reading material, and the next was for a tool demo for hands-on exploration. Table 3 lists the set of pages that AID reported violations for and outlines the fixes implemented by the faculty member.

6 RQ3 (STUDENTS): HOW DO FACULTY'S INCLUSIVITY FIXES IMPACT STUDENTS' EXPERIENCE?

6.1 Method

Participants: We recruited 20 participants enrolled in an undergraduate program at University <X>. None of them had yet taken the CS1 course that faculty fixed using AID, and were enrolled in the prerequisite course in the term when the study was conducted. We reached out to students taking the prerequisite course via email, inviting them to participate in a study where they would interact with materials related to an upcoming course. This was an opportunity for students to preview coursework they would need to complete in an upcoming term and we hope it motivated the students to approach the task genuinely.

Study Design: We conducted a between-subject study with two treatments- CS1-Original and CS1-Fixed (the post-fix version). Participants were assigned randomly and began by agreeing to the IRB approved consent form and completing a pre-study questionnaire with their demographics and the GenderMag facet survey [7] [29] to self-report their facet values. Next, to get participants to interact with the Week 1 materials, we designed two tasks within the course. First, was to set up the tools based on the information provided in the Tool SetUp page. Second, was to prepare for the first assignment. For each task, we instructed participants to navigate through the course materials while thinking aloud. Intentionally, the task descriptions were at a higher level and not UI-specific to allow participants to navigate the course naturally (See supplemental for script [23]). Each session lasted around 45-60 minutes and was conducted over Zoom using screen-sharing on our machine. All sessions were screen- and audio-recorded. We compensated participants with $20 Amazon gift cards for their time.

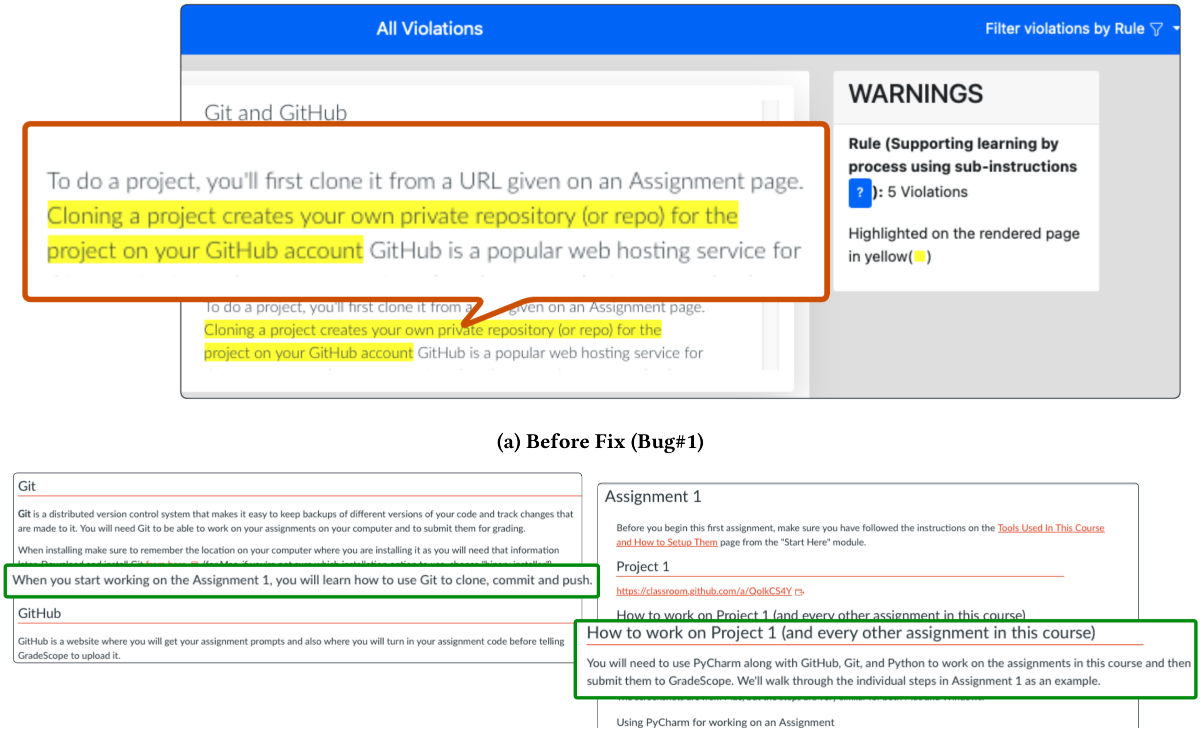
Data analysis: We transcribed each participant's think-aloud audio recordings and used inductive thematic analysis to identify themes [7]. Relevant phrases and sentences were labeled by both authors on a shared document, following an iterative constant comparative approach. Subsequent team meetings were held to discuss and refine the codes. As the analysis progressed, the authors built post-formed codes and associated them with parts of the transcripts. Next, we performed axial coding by comparing and contrasting codes to look for connections and repetitions. This step

[6] Quality Matters standards state that faculty should map course activities to the corresponding learning outcomes

[7] GenderMag Facet survey investigates how a user approaches problem-solving with technology

Table 3: Inclusivity bugs reported by AID in CS1. Col. 2-3: Type of barriers in the page, and facets it affects, Col. 4: Whether faculty agreed with the bugs, Col. 5: The fix implemented by faculty

Courseware Pages where AID found Bugs	Type of Barriers	Facets	Faculty Validated?	Implemented Fix
Bug#1: Tool SetUp Page	Blocked by information: *5 instances*	LS	✓	Removed extraneous information upfront (Fix1), Added step-by-step details just-in-time (Fix2)
Bug#2: Module1-Exploration1	Focus: *1 instance*	M, RA	✓	Added purpose (Fix3)
Bug#3: Module1-Exploration2	Blocked by information: *3 instances* Focus: *1 instance*	M, RA IP	✓	Added purpose (Fix3), Decomposed information into smaller units (Fix4)



To do a project, you'll first clone it from a URL given on an Assignment page. Cloning a project creates your own private repository (or repo) for the project on your GitHub account GitHub is a popular web hosting service for

To do a project, you'll first clone it from a URL given on an Assignment page. Cloning a project creates your own private repository (or repo) for the project on your GitHub account GitHub is a popular web hosting service for

Rule (Supporting learning by process using sub-instructions ?): 5 Violations

Highlighted on the rendered page in yellow( )

(a) Before Fix (Bug#1)

Git

Git is a distributed version control system that makes it easy to keep backups of different versions of your code and track changes that are made to it. You will need Git to be able to work on your assignments on your computer and to submit them for grading.

When installing make sure to remember the location on your computer where you are installing it as you will need that information

When you start working on the Assignment 1, you will learn how to use Git to clone, commit and push.

GitHub

GitHub is a website where you will get your assignment prompts and also where you will turn in your assignment code before telling GradeScope to upload it.

Assignment 1

Before you begin this first assignment, make sure you have followed the instructions on the Tools Used In This Course and How to Setup Them page from the "Start Here" module.

Project 1

https://classroom.github.com/a/QolkCS4Y

How to work on Project 1 (and every other assignment in this course)

You will need to use PyCharm along with GitHub, Git, and Python to work on the assignments in this course and then submit them to GradeScope. We'll walk through the individual steps in Assignment 1 as an example.

Using PyCharm for working on an Assignment

(b) After Fix

Figure 2: Fix1 & Fix2: (a) One out of five bugs in Tool SetUp Page (Table 3) (b) After the fix, the extraneous information was removed from the Tool SetUp page (Fix1-left) and step-by-step guidance was added in Assignment 1 (Fix2-right)

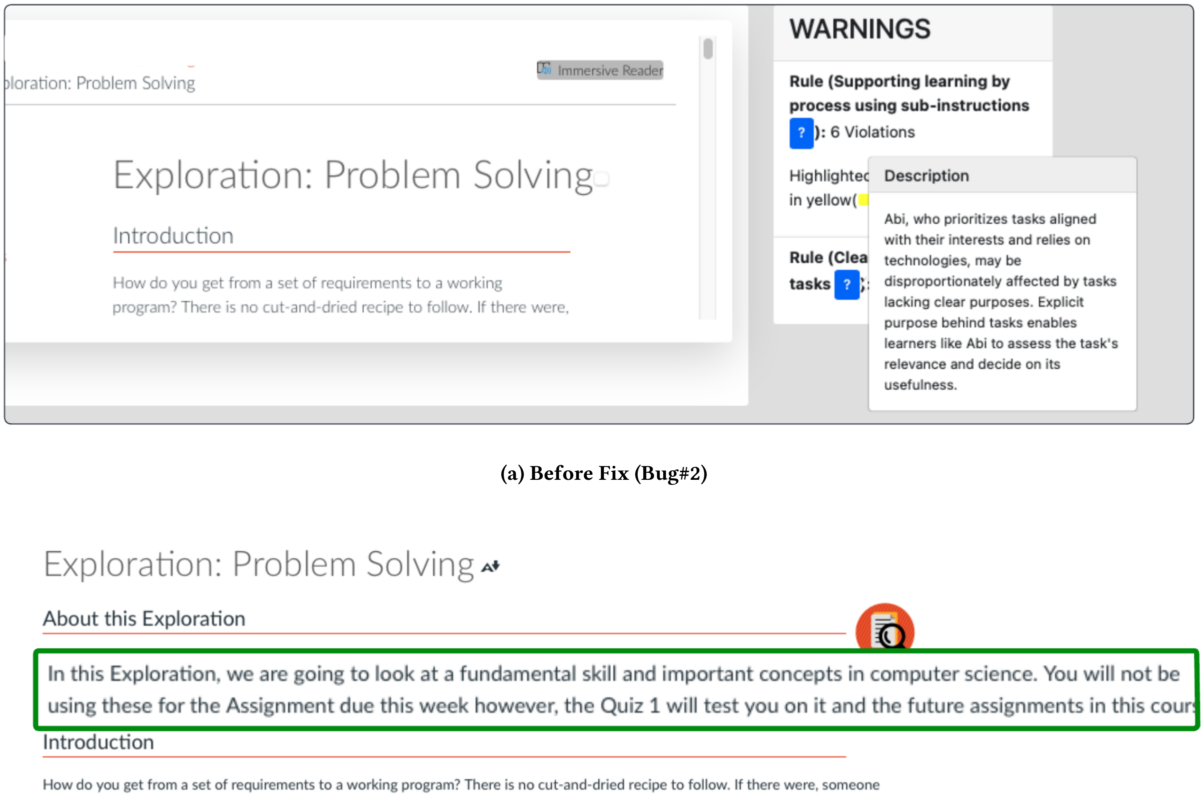
was also repeated three times, before merging them into the final themes.

6.2 Results

Here, we compare and contrast participants' experiences in navigating the first-week of classes in CS1 before and after the inclusivity fixes (discussed in Section 5.2). For clarity, we refer to the version before the fixes as CS1-Original, and the version post fixes as CS1-Fixed.

As Table 4 shows, with the CS1-Original version, participants ran into inclusivity bugs in the courseware 19 times. Faculty's inclusivity fixes to the courseware reduced the number of inclusivity bugs faced by participants to 4; an 80% reduction. Next, we discuss each set of fixes.

6.2.1 Fix1 & Fix2. These inclusivity fixes reduced the number participants facing the bug (Bug#1) from 80% to 20%. Recall, to fix this bug, faculty addressed Abi's process-oriented learning style and removed extraneous information upfront and added specific task-oriented instructions just-in-time in the assignment page (see Figure 2). Figure 4 shows to what extent these fixes helped participants with this facet. Four participants who were process-oriented learners (orange circles in Figure 4a, left panel) faced this bug, which reduced to 0 participants in the post-fix group (a 100% improvement). Additionally, participants with Tim-like learning style were



ploration: Problem Solving

Immersive Reader

Exploration: Problem Solving

Introduction

How do you get from a set of requirements to a working program? There is no cut-and-dried recipe to follow. If there were,

WARNINGS

Rule (Supporting learning by process using sub-instructions ?): 6 Violations

Highlighted in yellow(

Rule (Clea tasks ?

Description

Abi, who prioritizes tasks aligned with their interests and relies on technologies, may be disproportionately affected by tasks lacking clear purposes. Explicit purpose behind tasks enables learners like Abi to assess the task's relevance and decide on its usefulness.

(a) Before Fix (Bug#2)

Exploration: Problem Solving

About this Exploration

In this Exploration, we are going to look at a fundamental skill and important concepts in computer science. You will not be using these for the Assignment due this week however, the Quiz 1 will test you on it and the future assignments in this cour

Introduction

How do you get from a set of requirements to a working program? There is no cut-and-dried recipe to follow. If there were, someone

(b) After Fix

Figure 3: Fix3: (a) (Module1-Exploration1) and (Module1-Exploration2) flagged violations indicating that the purpose behind the tasks were not clear (b) In the fix, faculty added information about why these pages were important.

Table 4: How well each of the inclusivity bug fixes worked, assessed in terms of the number of participants who encountered them in the fixed CS1 version (Column 4). Each version had 10 participants, 8/10 indicates that 8 out of 10 participants encountered the bug.

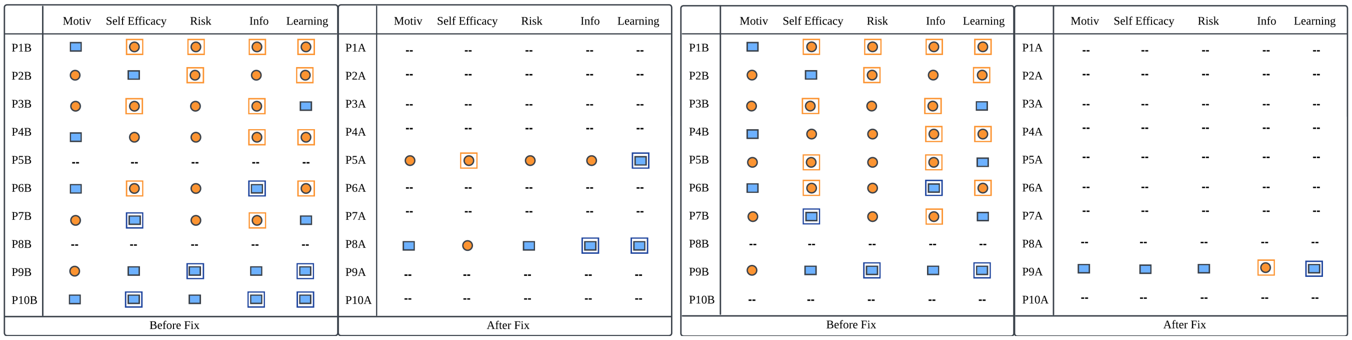
CS1-Original Pages	Bugs encountered in CS1-Original	CS1-Fixed Pages	Bugs encountered in CS1-Fixed
Bug#1: Tool SetUp Page	8/10	Fix1: Tool SetUp Page	0/10
		Fix2: Assignment 1	2/10
Bug#2: Module1-Exploration1	7/10	Fix3: Module1-Exploration1	1/10
Bug#3: Module1-Exploration2	4/10	Fix3: Module1-Exploration2a	1/10
		Fix4: Module1-Exploration2b	0/10
Total Bugs encountered	**19**		**4**

also helped (going from 4 to 2). This shows that the faculty using the output of AID was able to make their coursework better not only for the Abi-like participants, but also for Tim-like participants. The fixes for one type of learning style, did not disadvantage the other, but instead resulted in an improvement for all. Now, we show how the inclusivity fixes improved participants' experiences.

Behavior1 - Overwhelming to Manageable: A *lack of step-by-step guidance (challenge category 1)* along with *no connection to prior learning (challenge category 3)* created hurdles, especially for Abi-like users (process-oriented learners, risk averse).

For example, in CS1-Original, P1B (Abi facets of risk, learning style) felt overwhelmed right from the beginning when they encountered unfamiliar terms and tools (e.g., Git, GitHub, and PyCharm):

> [P1B] "...this is a lot. I don't really know what this is saying. Even to be able to understand this, I have to download this stuff first..."



(a) Bug#1-Fix1 & Fix2

(b) Bug#2-Fix3

Figure 4: (a) For Fix1 & Fix2, 8 CS1-Original participants ran into bugs (left), but only 2 CS1-Fixed participants did (right). (b) For Fix3, 8 CS1-Original participants ran into bugs (left), but only 1 CS1-Fixed participant did (right). Participant ID numbering is from the most Abi-like to the most Tim-like. Circles represent Abi-like and Squares represent Tim-like participants; square outline | square outline: Abi-like | Tim-like facet values participants expressed when they ran into a bug.

While participants in CS1-Original were able to continue their study task, they were still unclear about core concepts like "committing" code. This caused participants to repeatedly run into challenges. One participant articulated:

> [P3B] "...I don't really know what commit means yet?...it hasn't really defined PyCharm either...it wants me to use [tool], but I'm not sure where to get it"

In CS1-Fixed version, none of the Abi-like process-oriented learners encountered the bug (although, one participant with Abi-risk averse facet still faced an issue). Overall, Abi-like participants felt a greater sense of control and found the material to be manageable:

> [P1A] "I feel done with [instructions on Tool-SetUp page]. It's not that hard. It's not like a hard concept to understand."

P6A highlighted how step-by-step guidance was helpful:

> [P6A] "So the first part was running Pycharm and then the welcome screen...then you should see it looking for version control...all these steps are for setting up and making sure where to find it"

Behavior2 - Needing help vs self-explanatory content: In CS1-Original, participants required external assistance when setting up the necessary tools and working through course tasks. Participants (P3B, P4B) resorted to consulting YouTube tutorials to guide them through the process after downloading the tools. One participant, frustrated by the challenges, expressed their intention to seek external help, stating:

> [P6B] "Honestly, I'd probably ask tech. I'd probably message somebody else in the class or try to meet up with them and have kind of work through it with them because I'm struggling..."

When individuals with low Computer Self-Efficacy faced this bug (stemming from *lack of step-by-step guidance*), they attributed their difficulty to their own shortcomings; viewing it as a personal failure rather than realizing the deficiency in the courseware.

> [P3B] "I couldn't find the answer just by briefly going back and skimming where I think the answer would be...I think that's kind of my fault."

In contrast, in the CS1-Fixed version, none of the participants sought external assistance, with one participant emphasizing the benefit of explicit and structured information:

> [P2A] "...having written instructions and screenshots. This is a lot more accessible to me at least. Helps with my attention...the written out instructions, which I found really helpful. I didn't really run into any issues."

6.2.2 Fix3 & Fix4. : These two fixes helped address Bug#2 and Bug#3 (Table 4) by clearly articulating the purpose and expected outcomes of the (reading/exploration) activities. This helped support task-motivated and risk-averse individuals. The faculty also created separate pages for readings (Module1-Exploration2a) and activity (Module1-Exploration2b) to provide more contained, clear instructions of the two kinds of exploration and their purpose (Figure 3) .

Fix#3 for Bug#2 helped participants; 7 in CS1-Original and 1 in CS1-Fixed, (70% to 10%). Counting the orange circles (Abi facets) in the Motivations and Risk columns in Figure 4b-left, shows this bug (Bug#2) disproportionately impacted Abi's facet values: 75% (6/8) of the participants facing this bug reported Abi-like Motivations and/or Abi-like Risk facet values. In CS1-Fixed, none faced the bug.

Bug#3 was also addressed by Fixes#3 and #4. These fixes led to a 30% improvement (4 in CS1-Original to 1 after). Similar to Bug#2, Bug#3 also disproportionately impacted Abi's facet values: 100% (4/4) of the participants facing this bug reported Abi-like Motivations and/or Abi-like Risk facet values (See Supplemental [23] for details). In CS1-Fixed, none faced the bug.

These fixes were helpful to participants as follows.

Behavior3: Ambiguous vs. Well-defined. Participants in CS1-Original were confused about the purpose behind several pages labeled as "Exploration" - perceiving these as optional content, even though it was needed for Quiz 2. P9B (See Figure 4b), who was task-oriented and risk averse asked for confirmation from the researcher conducting the study:

> [P9B] "I don't need to go through [Module1-Exploration1], right?"

In contrast, participants found the content easy to follow in the CS1-Fixed version. Participant not only engaged with the "Exploration" pages, but they also appreciated the clarity of information. P3A reported:

> [P3A] "...very easy to figure out what I'm doing. Especially compared to some classes where this would be like a great paragraph of text and you have to try to break it down yourself"

Behavior4: Students engaged with courseware activities. In CS1-Original, none of the participants reached the hands-on exploration activity at the end of Module1-Exploration2 page, as they were uncertain whether any of the materials in "Exploration" page was required. However, in CS1-Fixed, we observed that all participants completed the activity of using the online editor tool, as faculty had intended. One participant, P2A, commented on how engaging the activities were, saying:

> P2A: "...these are fun activities to do. And it's not like super technical at this point. But it does get [students] thinking"

7 DISCUSSION

A key goal of this work was to investigate how an automated inclusivity checker, such as AID, could be used by faculty to improve inclusivity of their courseware. Our findings provide compelling evidence that AID is highly effective. AID essentially acted as a fault localizer, pinpointing the specific inclusivity bugs ("what"), highlighted where it was ("where") and provided explanations for why they were problematic based on the GenderMag facets ("why"). With this information—the what, where and why—faculty knew how to effectively integrate targeted inclusivity fixes into their courseware.

It is worth noting that AID and the well-established GenderMag method share some fundamental similarities. Humans using this method work directly from the root causes of these bugs - the underlying facets. We believe that AID was also able to capture inclusivity bugs well, because its underlying decision rules are derived from patterns of inclusivity challenges that manifest because of unsupported cognitive facets.

However, unlike the GenderMag approach that focuses on specific use case scenarios, the three decision rules we formulated in this work were designed to cast a broader, courseware-wide net for detecting inclusivity bugs across the entire platform, thereby saving significant time and effort. Even if AID were to incorporate a use case focus for these decision rules, that would require faculty to specify relevant use cases and subgoals that students would have - a non-trivial undertaking.

That said, while AID's comprehensive, courseware-wide purview is a key strength, future work should explore to what extent the two approaches find similar or different set of inclusivity bugs. Perhaps, the two approaches can be chained with a system-wide debugging phase followed by a more detailed use-case based debugging approach.

7.1 AID as a "peer" in an inclusive design process

Our results from RQ2 showed how educators can best leverage this version of AID as partners in an inclusive design process. First, we observed that faculty approached the tool's output with an open yet critical mind - they did not treat the tool's output as gospel, nor did they simply disregard it. Rather, they approached the tool's output as they would a "peer" reviewer - using it to gain new perspectives, but also applying their own expert judgment.

Specifically, the faculty's contextual expertise with students was crucial for evaluating which fixes were appropriate and effective for their specific context. Second, they viewed this not as a one-time DEI box to check off, but as an ongoing practice of continuous improvement. Evolving student needs, new pedagogical approaches, and changing technological landscapes - all underscore why inclusivity must remain a sustained, proactive effort.

7.2 The Fixes: From a gender perspective

Our results showed that faculty's fixes led to a 80% reduction in the inclusivity bugs that participants had encountered in CS1-Original. However, this leaves unanswered whether these fixes actually contributed to the goals of improving the courseware for minoritized groups. To examine this, we discuss the participants' demographics and facets.

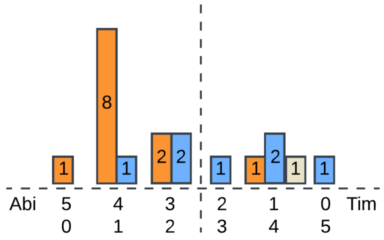


Figure 5: # of women (orange), men (blue), and non-binary or gender diverse (mustard) with each combination of facets (from facet questionnaire), using the x-axis scheme: from 5 Abi facets (left half, orange) to 5 Tim facets(right half, blue). For example: the first column says that 1 participant had 5 Abi facets and no Tim facets. Note that the right half of the graph contains only 1 of the 12 women participants.

Out of the 20 participants, 12 self-identified as women, 1 as non-binary, and 7 as men. Prior research has found that individual differences in preferred facets cluster by gender; with Abi's facets statistically more common among women. Our participant group displayed such a facet distribution as well—the facet values favored by women tended to skew more towards the 'Abi-like' end of the spectrum compared to their men counterparts (See Figure 5).

Recall, participants with Abi's facet values (e.g., in Figure 4a) were impacted by inclusivity bugs more often than Tim's in the original version. So while the curricular refinements produced overall improvements, they had an out sized positive impact for Abi-like participants. This combined with the fact that the women participants in our study were more likely to have Abi facets, shows that the inclusivity fixes helped women.

However, we reiterate the fact that fixing the inclusivity bugs from the perspective of the Abi persona created a much better experience not only for the Abis, but also the Tims. This mirrors the idea of universal design - much like how physical curb cuts intended for wheelchair users (one underserved population) also benefit other populations such as parents with strollers. Our study showed that by addressing inclusivity bugs, using AID, we can cultivate learning environments that are empowering for all students, allowing them to progress without getting unnecessarily bogged down by inclusivity barriers that create cognitive overheads.

8 THREATS TO VALIDITY

As with any empirical study, our work has threats to validity. This section describes the limits of our study scope and measures we took against them.

First, subjectivity may be involved in qualitatively identifying the inclusivity challenges from student posts. To minimize this threat, two authors conducted the analysis after establishing interrater reliability among them. Further, we triangulated the set of inclusivity challenges found by mapping them to the inclusivity barriers found in previous work [14], as shown in Table 1.

Prior work has recommended using the Abi persona first [23], as their facet values tend to be more under-supported in software than other personas. However, fixing bugs for only Abi's persona may leave non-Abi-like students less supported than before. To address this, we evaluated our fixes with both Abi-like and Tim-like students. We recognize that GenderMag personas do not take into account all cognitive facets (such as memory and attention span), which may be pertinent for people. Our work did not account for those kinds of cognitive facets.

For RQ3's user study (Section 6), participants engaged with the online CS courseware remotely, using their own computers. Given the course's online nature, this setup enabled them to interact with the course materials using familiar devices and in familiar environments. Moreover, it increased the likelihood that their thought processes and interaction patterns would reflect their natural behaviors, as opposed to those observed in an in-person lab setting. However, this setup also meant limited control over factors including device specifications, internet quality, and environmental conditions. Nevertheless, since the course was an online computer science offering, conducting the study in a lab environment would have been unrealistic, as it would not accurately represent the typical online learning experience. The remote setup ensured the recruitment of participants who could provide insights into the challenges faced by students in real-world online learning environments.

Another threat is that our study involved two tasks in a single CS course on canvas, which might not generalize to other courses or learning environments. Our investigation in RQ3, was designed as a between-subject study - where each participant had interacted with either the before or after fixed version of the course - to avoid learning effects and participant fatigue. The relatively small number of participants (20 in total) could also be a threat. However, quality instead of size is necessary to increase our confidence in the findings. Although, we reached saturation after the fifth study in both treatments, we continued the study with five additional participants in both treatments to gain diverse and balanced perspectives. Lastly, our participants were from the same university; this could limit the generalizability of the RQ1 and RQ3 results (e.g., because other universities use different course management systems, use different standards for building online courseware, etc.). Thus, caution is advised when interpreting our results to other universities, or educational settings. This limitation can be addressed by additional empirical studies in a variety of educational settings.

9 CONCLUSION

This paper has empirically investigated the impacts inclusivity bug fixes have on student experiences in online CS courseware. Our results revealed that:

- RQ1 (Bugs): Inclusivity challenges arising from a lack of support of cognitive diversity is pervasive. Students faced inclusivity challenges in all the five courses we analyzed; a total of 39 such instances.
- RQ2 (Build + Faculty): AID decision rules, informed by student data, could capture inclusivity bugs that students faced across all GenderMag facets. AID's results lay in its fault localization capabilities that it brought to faculty with the where's and why's behind the inclusivity bugs it identified automatically.
- RQ2 (Build + Faculty): The faculty member fixed the faults AID had localized in their courseware, by changing the courseware as detailed in Section 5.2 and summarized in Table 3.
- RQ3 (Students): Showed that the faculty's fixes helped across the cognitive diversity range of the students in our investigation (Figure 4).

A key takeaway is that these improvements did not come at the expense of any particular population or through compromises that left groups underserved. Instead, the fixes enhanced the experience for all participants. These results provide encouraging evidence that AID's decision rule-based approach to localizing inclusivity bugs may provide a concrete, practical, and effective way to improve the inclusivity of online CS courseware. As echoed by the faculty member's unsolicited email saying:

> [Faculty] "I am glad I am working on this because I am getting a perspective that I haven't had in 3 years of teaching [this course] using this same Canvas site."

ACKNOWLEDGMENTS

The authors wish to thank the students and instructors involved in the study. This work was supported in part by USDA-NIFA/NSF grant number 2021-67021-35344, and by NSF grant numbers 1901031, 2042324, 2235601, 2303043, and 2345334. Any opinions expressed are of those of the authors alone, and do not necessarily reflect the views of the sponsors.

REFERENCES

[1] Steven Abney. 1997. Part-of-speech tagging and partial parsing. In *Corpus-based methods in language and speech processing*. Springer, 118–136.

[2] I Elaine Allen and Jeff Seaman. 2010. *Learning on demand: Online education in the United States, 2009.* Babson Survey Research Group, United States of America.

[3] Manon Arcand and Jacques Nantel. 2012. Uncovering the nature of information processing of men and women online: The comparison of two models using the think-aloud method. *Journal of Theoretical and Applied Electronic Commerce Research* 7, 2 (2012), 106–120.

[4] Rachel Baker, Thomas Dee, Brent Evan, and June John. 2018. *Bias in Online Classes: Evidence from a Field Experiment.* Technical Report. Stanford Center for Education Policy Analysis. https://cepa.stanford.edu/sites/default/files/wp18-03-201803.pdf

[5] William H. Bares, Bill Manaris, Renée McCauley, and Christine Moore. 2019. Achieving Gender Balance through Creative Expression. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (Minneapolis, MN, USA) *(SIGCSE '19)*. Association for Computing Machinery, New York, NY, USA, 293–299. https://doi.org/10.1145/3287324.3287435

[6] Lecia J Barker, Charlie McDowell, and Kimberly Kalahar. 2009. Exploring factors that influence computer science introductory course students to persist in the major. *ACM Sigcse Bulletin* 41, 1 (2009), 153–157.

[7] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.

[8] Lori Breslow, David E Pritchard, Jennifer DeBoer, Glenda S Stump, Andrew D Ho, and Daniel T Seaton. 2013. Studying learning in the worldwide classroom research into edX's first MOOC. *Research & Practice in Assessment* 8 (2013), 13–25.

[9] Michael Buckley, Helene Kershner, Kris Schindler, Carl Alphonce, and Jennifer Braswell. 2004. Benefits of using socially-relevant projects in computer science and engineering education. In *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education.* 482–486.

[10] Margaret Burnett, Simone Stumpf, Jamie Macbeth, Stephann Makri, Laura Beckwith, Irwin Kwan, Anicia Peters, and William Jernigan. 2016. GenderMag: A method for evaluating software's gender inclusiveness. *Interacting with Computers* 28, 6 (2016), 760–787.

[11] Shuo Chang, Vikas Kumar, Eric Gilbert, and Loren G Terveen. 2014. Specialization, homophily, and gender in a social curation site: Findings from Pinterest. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing.* 674–686.

[12] Gary Charness and Uri Gneezy. 2012. Strong Evidence for Gender Differences in Risk Taking. *Journal of Economic Behavior & Organization - J ECON BEHAV ORGAN* 83, 1 (06 2012), 50–58. https://doi.org/10.1016/j.jebo.2011.06.007

[13] Amreeta Chatterjee, Mariam Guizani, Catherine Stevens, Jillian Emard, Mary Evelyn May, Margaret Burnett, Iftekhar Ahmed, and Anita Sarma. 2021. AID: An automated detector for gender-inclusivity bugs in OSS project pages. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. IEEE, 1423–1435.

[14] Amreeta Chatterjee, Lara Letaw, Rosalinda Garcia, Doshna Umma Reddy, Rudrajit Choudhuri, Sabyatha Sathish Kumar, Patricia Morreale, Anita Sarma, and Margaret Burnett. 2022. Inclusivity bugs in online courseware: A field study. In *Proceedings of the 2022 ACM Conference on International Computing Education Research-Volume 1.* 356–372.

[15] Ingrid Maria Christensen, Melissa Høegh Marcher, Paweł Grabarczyk, Therese Graversen, and Claus Brabrand. 2021. Computing Educational Activities Involving People Rather Than Things Appeal More to Women (Recruitment Perspective). In *Proceedings of the 17th ACM Conference on International Computing Education Research* (Virtual Event, USA) *(ICER 2021)*. Association for Computing Machinery, New York, NY, USA, 127–144. https://doi.org/10.1145/3446871.3469758

[16] Online Learning Consortium. 2022. *Quality Framework.* OLC. https://onlinelearningconsortium.org/about/quality-framework-five-pillars/ (accessed Mar. 2022).

[17] Alexiei Dingli and Justin Mifsud. 2011. Useful: A framework to mainstream web site usability through automated evaluation. (2011).

[18] Bob Dougherty and Alex Wade. 2008. *Vischeck.* Vischeck.com. http://www.vischeck.com/ (accessed Mar, 2022).

[19] Rodrigo Duran, Lassi Haaranen, and Arto Hellas. 2020. *Gender Differences in Introductory Programming: Comparing MOOCs and Local Courses.* Association for Computing Machinery, New York, NY, USA, 692–698. https://doi.org/10.1145/3328778.3366852

[20] Brianna Dym, Namita Pasupuleti, Cole Rockwood, and Casey Fiesler. 2021. "You don't do your hobby as a job": Stereotypes of Computational Labor and their Implications for CS Education. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education.* 823–829.

[21] Holly Fiock. 2020. Designing a community of inquiry in online courses. *The International Review of Research in Open and Distributed Learning* 21, 1 (2020), 135–153.

[22] Denae Ford, Justin Smith, Philip J Guo, and Chris Parnin. 2016. Paradise unplugged: Identifying barriers for female participation on stack overflow. In *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering.* 846–857.

[23] Anonymized ForReview. 2024. ICER 2024 submission materials. (3 2024). https://doi.org/10.6084/m9.figshare.25511869.v2

[24] Hana Frluckaj, Laura Dabbish, David Gray Widder, Huilian Sophie Qiu, and James D Herbsleb. 2022. Gender and participation in open source software development. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW2 (2022), 1–31.

[25] 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 (2023), 1–35.

[26] Rosalinda Garcia, Patricia Morreale, Gail Verdi, Heather Garcia, Jimena Noa Guevara, Spencer Madsen, Maria Jesus Alzugaray-Orellana, Elizabeth Li, and Margaret Burnett. 2024. The Matchmaker Inclusive Design Curriculum: A Faculty-Enabling Curriculum to Teach Inclusive Design Throughout Undergraduate CS. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems Proceedings (CHI'24).*

[27] D Randy Garrison, Terry Anderson, and Walter Archer. 1999. Critical inquiry in a text-based environment: Computer conferencing in higher education. *The internet and higher education* 2, 2-3 (1999), 87–105.

[28] Philip J Guo and Katharina Reinecke. 2014. Demographic differences in how students navigate through MOOCs. In *Proceedings of the first ACM conference on Learning@ scale conference.* 21–30.

[29] Md Montaser Hamid, Amreeta Chatterjee, Mariam Guizani, Andrew Anderson, Fatima Moussaoui, Sarah Yang, Isaac Escobar, Anita Sarma, and Margaret Burnett. 2023. How to measure diversity actionably in technology. *Equity, Diversity, and Inclusion in Software Engineering: Best Practices and Insights* (2023).

[30] Karen Hamrick. 2022. *Women, Minorities, and Persons with Disabilities in Science and Engineering.* NSF National Center for Science and Engineering Statistics (NCSES). https://ncses.nsf.gov/pubs/nsf19304/digest/field-of-degree-women#computer-sciences (accessed Mar. 2022).

[31] Hui-Ching Kayla Hsu and Nasir Memon. 2021. Crossing the Bridge to STEM: Retaining Women Students in an Online CS Conversion Program. *ACM Transactions on Computing Education (TOCE)* 21, 2 (2021), 1–16.

[32] Melody Y Ivory. 2000. Web TANGO: towards automated comparison of information-centric web site designs. In *CHI'00 extended abstracts on Human factors in computing systems.* 329–330.

[33] Leonard R Kasday. 2000. A tool to evaluate universal Web accessibility. In *Proceedings on the 2000 conference on Universal Usability.* 161–162.

[34] Prem Nawaz Khan, Cathy O'Connor, and Srinivasu Chakravarthula. 2021. *Automated Accessibility Testing Tool (AATT).* AATT. https://github.com/paypal/AATT (accessed Mar, 2022).

[35] René Kizilcec and Andrew Saltarelli. 2019. Psychologically Inclusive Design: Cues Impact Women's Participation in STEM Education. In *Proceedings of the 2019 CHI Conference on human factors in computing systems (CHI '19).* ACM, 1–10.

[36] René F. Kizilcec and Sherif Halawa. 2015. Attrition and Achievement Gaps in Online Learning. In *Proceedings of the Second (2015) ACM Conference on Learning @ Scale* (Vancouver, BC, Canada) *(L@S '15)*. Association for Computing Machinery, New York, NY, USA, 57–66. https://doi.org/10.1145/2724660.2724680

[37] Maria Klawe. 2013. Increasing female participation in computing: The Harvey Mudd College story. *Computer* 46, 3 (2013), 56–58.

[38] Sophia Krause-Levy, William G. Griswold, Leo Porter, and Christine Alvarado. 2021. The Relationship Between Sense of Belonging and Student Outcomes in CS1 and Beyond *(ICER 2021)*. Association for Computing Machinery, New York, NY, USA, 29–41. https://doi.org/10.1145/3446871.3469748

[39] Sophia Krause-Levy, Mia Minnes, Christine Alvarado, and Leo Porter. 2021. Experience report: Designing massive open online computer science courses for inclusion. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1.* 95–101.

[40] J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics* (1977), 159–174.

[41] Lara Letaw, Rosalinda Garcia, Heather Garcia, Christopher Perdriau, and Margaret Burnett. 2021. Changing the Online Climate via the Online Students: Effects of Three Curricular Interventions on Online CS Students' Inclusivity. In *Proceedings of the 17th ACM Conference on International Computing Education Research.* ACM, Online, 42–59.

[42] Patrick R Lowenthal and Charles B Hodges. 2015. In search of quality: Using quality matters to analyze the quality of massive, open, online courses (MOOCs). *International Review of Research in Open and Distributed Learning* 16, 5 (2015), 83–101.

[43] Jennifer Mankoff. 2006. Practical service learning issues in HCI. In *CHI'06 Extended Abstracts on Human Factors in Computing Systems.* 201–206.

[44] Melissa Høegh Marcher, Ingrid Maria Christensen, Paweł Grabarczyk, Therese Graversen, and Claus Brabrand. 2021. Computing Educational Activities Involving People Rather Than Things Appeal More to Women (CS1 Appeal Perspective).

In *Proceedings of the 17th ACM Conference on International Computing Education Research* (Virtual Event, USA) *(ICER 2021)*. Association for Computing Machinery, New York, NY, USA, 145–156. https://doi.org/10.1145/3446871.3469761

[45] Quality Matters. 2022. *Enhancing Inclusiveness within the Quality Matters Framework*. Quality Matters. https://www.qualitymatters.org/qa-resources/resource-center/conference-presentations/enhancing-inclusiveness-within-quality (accessed Mar. 2022).

[46] Quality Matters. 2022. *Quality Matters*. https://www.qualitymatters.org/ (accessed Mar. 2022).

[47] Christopher Mendez, Zoe Steine Hanson, Alannah Oleson, Amber Horvath, Charles Hill, Claudia Hilderbrand, Anita Sarma, and Margaret Burnett. 2018. Semi-Automating (or not) a Socio-Technical Method for Socio-Technical Systems. In *2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 23–32.

[48] Christopher Mendez, Hema Susmita Pedala, Zoe Steine-Hanson, Claudia Hilderbrand, Amber Horvath, Charles Hill, Logan Simpson, Nupoor Patil, Anita Sarma, and Margaret Burnett. 2017. Open source barriers to entry, revisited: A tools perspective. (2017).

[49] NCES. 2021. *Digest of Education Statistics, Table 311.15.* National Center for Education Statistics. https://nces.ed.gov/programs/digest/d20/tables/dt20_311.15.asp (accessed Mar. 2022).

[50] Alannah Oleson, Christopher Mendez, Zoe Steine-Hanson, Claudia Hilderbrand, Christopher Perdriau, Margaret Burnett, and Amy J. Ko. 2018. Pedagogical Content Knowledge for Teaching Inclusive Design. In *Proceedings of the 2018 ACM Conference on International Computing Education Research* (Espoo, Finland) *(ICER '18)*. Association for Computing Machinery, New York, NY, USA, 69–77. https://doi.org/10.1145/3230977.3230998

[51] Shailendra Palvia, Prageet Aeron, Parul Gupta, Diptiranjan Mahapatra, Ratri Parida, Rebecca Rosner, and Sumita Sindhi. 2018. Online education: Worldwide status, challenges, trends, and implications. , 233–241 pages.

[52] Alicia Parrish, Angelica Chen, Nikita Nangia, Vishakh Padmakumar, Jason Phang, Jana Thompson, Phu Mon Htut, and Samuel R Bowman. 2021. BBQ: A hand-built bias benchmark for question answering. *arXiv preprint arXiv:2110.08193* (2021).

[53] Krystle Phirangee and Alesia Malec. 2017. Othering in online learning: An examination of social presence, identity, and sense of community. *Distance Education* 38, 2 (2017), 160–172.

[54] Peter Pirolli and Stuart Card. 1999. Information foraging. *Psychological review* 106, 4 (1999), 643.

[55] Debashish Pradhan, Tripti Rajput, Aravind Jembu Rajkumar, Jonathan Lazar, Rajiv Jain, Vlad I. Morariu, and Varun Manjunatha. 2022. Development and Evaluation of a Tool for Assisting Content Creators in Making PDF Files More Accessible. *ACM Trans. Access. Comput.* 15, 1, Article 3 (mar 2022), 52 pages. https://doi.org/10.1145/3507661

[56] James Pustejovsky and Branimir Boguraev. 1993. Lexical knowledge representation and natural language processing. *Artificial Intelligence* 63, 1-2 (1993), 193–223.

[57] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. *Text mining: applications and theory* (2010), 1–20.

[58] Ali Akbar Septiandri, Marios Constantinides, Mohammad Tahaei, and Daniele Quercia. 2023. WEIRD FAccTs: How Western, Educated, Industrialized, Rich, and Democratic is FAccT?. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*. 160–171.

[59] Nikhil Singh, Guillermo Bernal, Daria Savchenko, and Elena L Glassman. 2023. Where to hide a stolen elephant: Leaps in creative writing with multimodal machine intelligence. *ACM Transactions on Computer-Human Interaction* 30, 5 (2023), 1–57.

[60] Christine K Sorensen and Danilo M Baylen. 2009. Learning online. *DISTANCE LEARNING EDITORS AND EDITORIAL ADVISORY BOARD* 7 (2009).

[61] Steven E Stemler. 2004. A comparison of consensus, consistency, and measurement approaches to estimating interrater reliability. *Practical Assessment, Research, and Evaluation* 9, 1 (2004), 4.

[62] Simone Stumpf, Anicia Peters, Shaowen Bardzell, Margaret Burnett, Daniela Busse, Jessica Cauchard, and Elizabeth Churchill. 2020. Gender-inclusive HCI research and design: A conceptual review. *Foundations and Trends in Human–Computer Interaction* 13, 1 (2020), 1–69.

[63] Adrian Thinnyun, Ryan Lenfant, Raymond Pettit, and John R Hott. 2021. Gender and Engagement in CS Courses on Piazza. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. 438–444.

[64] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[65] Bogdan Vasilescu, Andrea Capiluppi, and Alexander Serebrenik. 2014. Gender, representation and online participation: A quantitative study. *Interacting with Computers* 26, 5 (2014), 488–511.

[66] Mihaela Vorvoreanu, Lingyi Zhang, Yun-Han Huang, Claudia Hilderbrand, Zoe Steine-Hanson, and Margaret Burnett. 2019. From Gender Biases to Gender-Inclusive Design: An Empirical Investigation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. ACM, Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3290605.3300283

[67] Jayce R Warner, Carol L Fletcher, Nicole D Martin, and Stephanie N Baker. 2021. Applying the CAPE framework to measure equity and inform policy in computer science education. *Policy Futures in Education* (2021), 14782103221074467.

[68] Linda L Werner, Brian Hanks, and Charlie McDowell. 2004. Pair-programming helps female computer science students. *Journal on Educational Resources in Computing (JERIC)* 4, 1 (2004), 4–es.

[69] Julia Yates and Anke C Plagnol. 2022. Female computer science students: A qualitative exploration of women's experiences studying computer science at university in the UK. *Education and Information Technologies* 27, 3 (2022), 3079–3105.

[70] Kimberly Michelle Ying, Lydia G Pezzullo, Mohona Ahmed, Kassandra Crompton, Jeremiah Blanchard, and Kristy Elizabeth Boyer. 2019. In their own words: Gender differences in student perceptions of pair programming. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. 1053–1059.