Diversity, Equity, and Inclusion in Computing Science: Culture is the Key, Curriculum Contributes

Giulia Toti[*]
giulia.toti@ubc.ca
Department of Computer Science
University of British Columbia
Vancouver, BC, Canada

Peggy Lindner[*]
plindner@central.uh.edu
Department of Information Science Technology
University of Houston
Houston, TX, USA

Alice Gao[*]
ax.gao@utoronto.ca
Department of Computer Science
University of Toronto
Toronto, ON, Canada

Ouldooz Baghban Karimi[*]
ouldooz@sfu.ca
School of Computing Science
Simon Fraser University
Surrey, BC, Canada

Rutwa Engineer
rutwa.engineer@utoronto.ca
Department of Mathematical and Computational Sciences
University of Toronto Mississauga
Mississauga, ON, Canada

Jinyoung Hur
jhur10@illinois.edu
Department of Computer Science
University of Illinois Urbana-Champaign
Urbana, IL, USA

Fiona McNeill
f.j.mcneill@ed.ac.uk
School of Informatics
University of Edinburgh
Edinburgh, Scotland, UK

Shanon Reckinger
shanon@uic.edu
Department of Computer Science
University of Illinois Chicago
Chicago, IL, USA

Rebecca Robinson
rebecca.robinson@monash.edu
Department of Human Centred Computing
Monash University
Clayton, Victoria, Australia

Anna Sollazzo
anna.sollazzo@glasgow.ac.uk
School of Computing Science
University of Glasgow
Glasgow, Scotland, UK

Richard Wicentowski
rwicent1@swarthmore.edu
Computer Science Department
Swarthmore College
Swarthmore, PA, USA

Abstract

Undergraduate computer science programs worldwide struggle to attract and retain underrepresented students for many reasons. Culture, stereotype threats, uneven gender and racial representations, lack of role models, and uncertain career prospects for minority groups are among the many reasons behind this situation. Many computer science programs are trying to change course through strategies to foster equity, diversity, and inclusion (EDI), aimed at improving outreach, recruitment, admissions, and retention of underrepresented students. EDI approaches may also include modifications to the undergraduate computer science curriculum. However, if not properly planned, these modifications risk amplifying existing stereotypes rather than producing positive change [38].

In this study, through an extensive literature review, a rigorous curriculum analysis of 49 computer science programs across the globe, and qualitative and quantitative analysis of surveys and interviews bringing in the voices of 613 students and 30 educators participating from around the world, we explore equity, diversity, and inclusion in the computer science curriculum. We highlight the role of inclusive content and course design, discuss program flexibility, and the impact of inclusive courses and program design in attracting and retaining historically marginalized students. Finally, we provide concrete steps to make computing science undergraduate curricula more appealing to a diverse audience.

[*]Working Group Leaders

ITiCSE-WGR 2024, Milan, Italy

ACM ISBN 979-8-4007-1208-1/24/07
https://doi.org/10.1145/3689187.3709611

CCS Concepts

• **Social and professional topics → Computer science education**.

Keywords

Computer Science, Computing Science, Undergraduate Curriculum, Equity, Diversity, Inclusion, EDI

ACM Reference Format:

Giulia Toti, Peggy Lindner, Alice Gao, Ouldooz Baghban Karimi, Rutwa Engineer, Jinyoung Hur, Fiona McNeill, Shanon Reckinger, Rebecca Robinson, Anna Sollazzo, and Richard Wicentowski. 2024. Diversity, Equity, and Inclusion in Computing Science: Culture is the Key, Curriculum Contributes. In *2024 Working Group Reports on Innovation and Technology in Computer*

Science Education (ITiCSE-WGR 2024), July 8–10, 2024, Milan, Italy. ACM, New York, NY, USA, 51 pages. https://doi.org/10.1145/3689187.3709611

1 Introduction

Recommended approaches to achieve equity, diversity, and inclusion (EDI) in undergraduate Computer Science (CS) programs [5, 6, 76, 93] suggest that the best way to achieve this is through culture change. At the same time, efforts on other elements, including advertisement, recruitment, admissions, and program enhancements for retention, are impactful. Research evidence suggests that curriculum structure and changes [65, 76] can be essential in attracting and retaining historically marginalized students in CS programs.

In this work, we investigated existing efforts on equity-minded curriculum design in undergraduate CS programs and their adoption by educators and institutions. We examined their effectiveness in providing access, motivation to enroll, cultural relevance, and engagement to retain. We ask:

RQ1: What are examples of EDI-centered curriculum design efforts in undergraduate CS programs?
RQ2: What are the outcomes of these EDI-centered undergraduate CS curriculum design efforts?
RQ3: How can undergraduate CS programs adopt EDI-centered curricula?

To answer these questions, we first performed an extensive literature review on the undergraduate CS curriculum and interventions to improve it. We found ***six major categories of efforts*** designed to broaden participation in computing through curricular interventions. For each category, we reviewed examples to understand their curricular focus, impact, adoptions, and contexts. When data was available, we also explored their measurable success in achieving EDI outcomes. The literature highlighted examples of interventions that have been adopted in varying degrees across different institutions and regions, with some cases showing evidence of replication. However, gaps remain in evaluating broader impact and scalability. These findings underscore the diverse challenges of implementing effective EDI-focused curricular interventions and emphasize the importance of tailoring them to institutional and regional contexts.

Following the literature review, we moved to the analysis of existing curricula, choosing three major geographical regions to focus on: Australia, North America, and Europe. We made this choice because the vast majority of the literature accessible to the authors was based in these regions, and because they share cultural similarities. Existing research [64, 111] suggests that demographic inequalities in computing and STEM differ considerably across different parts of the globe. Therefore, the extent to which insights from this literature apply to other global regions, particularly those with significant cultural differences, is difficult to determine. We therefore urge caution in applying insights from this paper outside the three regions we focus on. Exploration of other global regions is important but outside the scope of this work.

We analyzed forty-nine highly-ranked undergraduate CS programs in our three selected regions, focusing on how well their curricula align with the ideas and interventions in the literature. We examined equity-minded approaches to curriculum development with a special focus on introductory CS courses, availability of interdisciplinary options, offering of upper-division courses, co-op placement opportunities, choice of electives, and the possibility of engaging in research or implementation projects. Throughout the analysis of curriculum efforts in the selected institutions and the diversity of their student populations, we assessed the effectiveness of these efforts. Our data sources for this analysis comprised publicly available information on degree requirements, courses offered, course information, syllabi in relevant programs, experience reports, and public reports of student populations in these institutions. We observed programs adopting the interventions in varying degrees with significant regional differences regarding how the undergraduate CS curricula provide fair access and motivation for underrepresented students. This curriculum analysis complemented our literature review towards answering **RQ1** and **RQ2**.

To further explore **RQ2** from the students' perspective, and answer **RQ3**, we complemented our literature review and curriculum analysis with Student and Educator surveys and interviews. We adopted mixed methods analysis and brought in the voices of 613 students and 30 faculty participants from across the globe. We used qualitative analysis on open-ended questions and quantitative analysis on Likert-scale and multi-select data. The analysis helped us better understand educators' perspectives, adoption, and observations on curriculum effectiveness, and student experiences with and opinions on the CS curriculum, and their suggestions to improve it. We reproduced our previous findings [5, 6] on the significance of differences in the assessment of CS programs' culture among gender identities. Furthermore, we expanded prior findings across racial diversity and accessibility needs, suggesting significant differences within all minoritized groups in their expressions of the possibility of success for them as a computer scientist. We also found significant differences in experiences with CS culture within each group and significant differences among gender identities and people with and without accessibility needs with their observations of elements of discouragement in CS curriculum.

We then explored the open-ended answers through thematic analysis on details of student course and curriculum experiences, and suggestions to improve the CS curriculum. Our findings reiterated some of our quantitative findings on course categories and subjects of interest. Our findings also elaborated on how content, practicality, and course design can impact students' feelings of belonging, enjoyment, and engagement with the content. Through our qualitative analysis, we gathered preferences and ideas on improving student engagement with the curriculum and performed a sentiment analysis towards EDI solutions. We complemented our learning through statistical and contextual analysis on educator surveys and interviews. We focused on observations on course-level and organizational efforts, educator suggestions for improvement, and indicated needs for support within organizations to better achieve diversity and inclusion through CS curriculum.

Our curriculum work is inspired by Paulo Freire's [37] learner-centericity in curriculum, encouraging active engagement of the students in the learning process, dialogue, affirming human dignity, and liberation through education. We examine curriculum in relation to learner's needs and how it actively engages the students based on their interests, experiences, culture, and personal and societal needs. We examine curriculum based on opportunities provided to students to expose and explore, actively challenge, and build relevant technical, cultural, and social justice competencies.

We present our work in four major sections: a comprehensive literature review on diversity and inclusion considerations in CS curriculum; a study of EDI-related curriculum efforts; analysis of student voices; and educators' feedback. We incorporate our methodology for study design, data collection, and analysis within each section and conclude each section with major findings, a discussion of results, and threats to validity. We then summarize our findings in a final analysis and provide recommendations to improve CS curriculum to achieve diversity and inclusion. The rest of this paper is organized as follows. In section 2, we define and clarify our terminology and context throughout the paper. In section 3, we explore the works and recommendations for broadening participation in CS through curriculum interventions. In section 4, we present our process of choosing 49 universities across the globe and exploring their CS curriculum. Section 5 presents our student survey and interview design, participant recruitment, and results of quantitative and qualitative analysis. Section 6 follows with survey and interview design, participant recruitment, and results for educators. We discuss the threats to the validity of our work in section 7. While we conclude each section with findings and a discussion, section 8 summarizes all of our findings and discusses our recommendations for adopting an EDI-centered CS curriculum. Finally, section 9 discusses the future directions to explore.

2 Context and Terminology

Throughout this paper, we use terminology that may have different meanings depending on context and the region across the world. In this section, we explain some of our terminology choices and try to clarify the context and possible differences in interpretation.

Racialized: The Oxford dictionary defines this adjective as "affected or influenced by racism" [82]. We chose the term as a more inclusive alternative to the usage of BIPOC (Black, Indigenous, and People Of Color), Visible Minority, and other variations used to study or explain racial background. Our choice is based on recommendations [114] and the possible impact of terminology in excluding one or more populations or possible associations with disrespectful or traumatic historical meanings.

EDI: We use this term to refer to Equity, Diversity, and Inclusion (EDI). We are aware that DEI, and other variations that also include justice (i.e., JEDI or EDIJ), are in use and that the usage is different across regions included in our study. Our work mentions but does not explore justice-centric solutions, therefore we limited our scope to EDI throughout our study. We also defined the term before use throughout our study.

We also realized that terms course, unit, and credit are used with different meanings, sometimes interchangeable and confusing, across the regions in our study. We are aware of possible differences in interpretation. We have performed an analysis to elaborate on the possible impacts of it on our work. We present possible threats to validity in section 7.

3 Literature Review

In this section, we provide an overview of the challenges identified in addressing equity, diversity and inclusion (EDI) initiatives in CS undergraduate curricula. In our literature review, we focus on recent reports and interventions and present a handful of examples in each category of initiatives as they emerged in the scope of our research. Gathering comprehensive evidence of all initiatives is outside the scope of this paper.

3.1 Needs for Changes in the CS Curriculum

With the continuous evolution of the discipline, the undergraduate CS curriculum undergoes frequent updates and revisions to stay aligned with the advancements in the field. Recent examples of this include the increased demand for Data Science and AI courses [13], or the emerging focus on creating more inclusive software [124]. Moreover, CS majors are dealing with some difficulties, such as high levels of attrition (students switching majors or dropping out altogether [43, 110]) and lacklustre pass rates (a concern shared with other STEM majors [106]). Studies report more negative impacts on students belonging to marginalized groups: CS majors are still struggling to achieve better gender and racial representation [9, 14, 23, 36, 110], and issues of isolation [8, 26, 101, 103], stereotype threats [14, 61, 88], and struggles with in-class climate [51, 63, 89, 100] are all factors that can hinder less represented students' chances to complete a CS program (or even start one).

Numerous interventions have been designed and deployed to address these issues. Many of them can be considered pedagogical in nature: these are strategies and techniques that the instructors can use to deliver their content and facilitate students' learning (e.g. various examples of Active Learning [15]), but are not connected to the course content. We do not cover this kind of intervention. Others, which we will focus on, are more closely correlated to the curriculum, as they impact *what* is taught in the classroom, and not just *how* it is taught. An example would be which programming language to use in a CS1 course.

We provide an overview of recent interventions applied to the CS curriculum to address the specific needs of the student populations and the changing requirements of industry, where this is relevant to diversity. Adopting an inductive approach, we classify the interventions into six major groups, each addressing a distinct need for broadening participation:

(1) *Levelling the playing field to help students with different technical backgrounds:* This group of interventions tackles the needs of students entering a CS major with varying levels of programming experience.
(2) *Integration of capstones, research, and industry projects in the curriculum:* Allowing students to apply the content learned in the classroom and connect with real-world applications is a way to improve retention and a sense of belonging.
(3) *Simplifying curricular complexity:* Higher curricular complexity is a factor behind high attrition rates. These interventions aim at reducing this problem.
(4) *Creating interdisciplinary routes into computing:* Combining computing with subjects with higher diversity or reimagining CS courses around interests and subjects that attract more diverse students can attract a more diverse student population.
(5) *Challenging stereotypes through diverse examples and applications:* Students from underrepresented groups report a lack of interest and engagement with CS material tailored to a

Table 1: Examples of Curriculum Interventions: Leveling the Playing Field in Introductory Programming Courses

Project	Institution	Curriculum Impact	Year
Themed CS0	CalPoly	Improved performance and retention [49]	2012
	UIC	Media-themed CS0: Low DFW and better student success [108]	2008
CS0 Bridge Course	Ohio	Improved performance for students with no experience [69]	2016
CS0.5 for Non-CS Majors	UBC	Improved Student Outcomes [32]	2018
Breadth-First CS	HMC	One of several interventions that increased the percentage of women in CS to 40% [3]	2012
Splitting CS1 Sections	UVA	Comparable CS2 outcomes regardless of prior programming experience [29, 57]	2011
(by previous experience)	JMU		2017
Objects-first Programming	UWG	Comparable CS1 outcomes regardless of prior programming experience [116]	2004

CalPoly: California Polytechnic State University, **UIC:** University of Illinois Chicago, **Ohio:** Ohio University, **HMC:** Harvey Mudd College, **UVA:** University of Virginia, **JMU:** James Madison University, **UWG:** University of West Georgia, **UBC:** University of British Columbia, **WPI:** Worcester Polytechnic Institute.

more stereotypical CS population. These interventions aim to change that.

(6) *Cultural and social justice competency:* Increase focus on EDI within CS curricula and on how technology has historically been driven by industry requirements to meet the needs of a few. Solutions try to encourage educators to move toward the development of technology that benefits all.

3.2 Interventions to Improve CS Curriculum

In this section, for each of the six major groups of interventions, we provide details about the context that prompted this change request, an overview of recent example interventions in the category, and their impact. We present our observations with particular attention to how these interventions affected underrepresented CS students.

3.2.1 Leveling the Playing Field in the Introductory Programming Courses. One challenge many CS programs face is that admitted students come in with varying levels of programming experience and computing knowledge [90]. Students have reported stress, low self-confidence, and feelings of intimidation when taking CS1 with other students with more prior programming experience [112]. Another challenge universities face is having low pass rates in their CS1 course, particularly among underrepresented groups [78]. Recent research shows a significant gap in course performance when comparing students with and without prior programming experience [20].

Bridge courses: Creating bridge courses to help students with little to no programming experience has helped improve the experience of those without tech privilege [49, 69, 108]. One study found that a CS0 course for CS majors, which covered introductory programming in Python, computational thinking, and motivational content for students with limited programming experience, significantly improved the CS1 performance of students with no prior programming experience, compared to those with some programming background [69]. Themed CS0 courses have been found to result in increased student retention and increased academic performance post-CS0 [49]. Media computation has also been used as a tool for non-CS majors to provide a more engaging curriculum, with the results showing very low DFW rates (the rate of students who got a D, failed or withdrew from the course) and better student success [108]. Considering that a larger percentage of women take CS courses as non-majors than as majors [102], it is important to continue offering accessible introductory programming courses for non-majors or avoid creating barriers that prevent non-majors from enrolling in CS courses designed for majors.

Splitting CS1 based on prior experience: Negative consequences of differences in previous programming experience before enrolling in university disproportionately affect women and underrepresented minorities [68]. Splitting CS1 based on prior programming experience can help in achieving comparable CS2 outcomes regardless of prior experience [29, 32, 57], as well as a way of addressing misconceptions about the need for a strong programming background to enter university [3]. However, splitting CS1 based on prior programming experience does not always improve retention of women and those from other underrepresented groups [57]. Others reported that teaching an objects-first CS1 course levels the playing field, where those without prior experience do as well as those with experience [116].

Programming Language Choice in CS1 : Choice of programming language can help level the playing field with students taking CS1. Students from underrepresented groups are less likely to have learned a new language before starting their university program [125]. Block-based programming may be useful for novices. Block-based programming is less common than textual programming and is used primarily with children [91]. Coffman et al. [28] found that students at their university perceived text-based programming to be more valuable than block-based programming, but they also found that some students from underrepresented groups struggled more with text-based programming than students from majority groups. One study found that 88% of universities were using C++, Python, Java, and C [105] in their introductory courses, with the growing trend that Python is becoming more popular than others. Some universities have moved to less common programming languages, like Racket or Pyret in CS1, to help mitigate the challenges of students coming with previous programming experience in the language chosen for that class. Harvey Mudd uses a locally-created Karel-like language in the initial weeks of CS1 and then uses Python as a functional language to create a situation where all students are facing new languages and/or paradigms [3].

3.2.2 Capstone, Research, Industry Projects Integrated Into Curriculum. Undergraduate research impacts recruitment and retention in computing positively [104]. The National Science Foundation (NSF) has supported undergraduate research programs since 1953

Table 2: Examples of EDI-Centered CS Curriculum Efforts: Capstone, Research, Industry Projects Integrated Into Curriculum

Project	Institution	Curriculum Impact	Year
Research Experience for Undergrads (REUs)	MI	Women and underrepresented minority groups had higher than the mean scores on several measures of assessment factors [96]	2018
Intensive Research Experiences (IREs)	MI	Improved attitudinal constructs, especially for women of color [97]	2020
Public Speaking in Capstone Courses	UCR	Beneficial for students from underrepresented groups [47]	2020
Systemic Justice Themed Capstone	MIT	Authentic and relevant to African American students [118]	2023
IoT-based Capstone Projects	TAMUCC TAMUK	IoT based project-based learning in a remote setting is as effective for historically underrepresented students [75]	2023

MI: Multi-Institution, **UCR:** University of California, Riverside, **MIT:** Massachusetts Institute of Technology, **TAMUCC:** Texas A&M University - Corpus Christi, **TAMUK:** Texas A&M University - Kingsville

and conducted a two-year study into the impact of undergraduate research on computing students. This study indicated significant benefits for women and underrepresented minority groups, with the former group showing higher scientific identity and scientific leadership than the mean and the latter group showing higher self-efficacy, academic help-seeking, research skills and leadership skills [96]. This is a strong indication that a focus on research projects could be beneficial for underrepresented groups. However, further research is required to understand this more fully — it is possible for selection bias to be affecting these populations of undergraduate students, causing them to already have higher scores than the mean in these attributes. Intensive Research Experiences (IREs) are workshops targeted at undergraduate women which expose them to pathways to a career in CS research [97]. These IREs are unique because they are more scalable than other research opportunities, and have been shown to be an effective way of providing undergraduate women with peer engagement and feel like they fit into CS research, particularly for women of color.

Internships are important links between college education and a successful transition into the job market, with those who have undertaken an internship 48.5% less likely to face graduate underemployment than those who do not (controlling for factors such as gender, ethnicity and college) [48]. Given that women and those from ethnic minorities are less likely to get paid internships [83, 84], this suggests an important role for similar activities within degree curricula. Break Through Tech empowers, trains, and connects a new generation of diverse tech talent (with a focus on Black, LatinX, Native Americans, low-income women, and gender non-conforming individuals) into the US workplace through their Sprinternship program. These programs have been done in partnership with a home institution (CUNY System in New York City, University of Illinois in Chicago, etc.). Sprinternships are three-week micro-internships in partnership with a tech company. They showed that only 4% of students without Sprinternships went on to secure a full summer internship, while 65% of students with a Sprinternship went on to secure one [81]. It is also important, however, to be mindful of the fact that internships may also expose minoritized students to hostile environments and behaviors, such as having their competence underestimated or having to alter their appearance to avoid attracting attention [109], with potential consequences for their long term studies and careers. Others have shown that Black women working in tech internships must develop coping strategies to manage, among other things, the constant need to prove their competency [94].

Capstone projects are a valuable contribution to a degree for helping students develop the hard and soft skills that are required in industry [113]. These can be beneficial for underrepresented groups and can foster a sense of inclusion [12], but there are also risks associated with them. When students are working in groups, unless these are well managed, roles within them can be allocated along gender roles, with male students doing the majority of the coding and female students taking on more admin-based roles [17]. This can lead to a view among male students that they are more valuable to the project. Women are also more likely to undervalue their own contribution, even where their peers rate it highly [11].

Haji Amin Shirazi et al. [47] found that integrating oral communication and public speaking skills into their capstone course was particularly beneficial for students from underrepresented groups. Wallker [118] developed a systemic justice capstone project that is relevant and authentic to that is relevant and authentic to African American students. Mehrubeoglu et al. [75], through interdisciplinary capstone projects at Hispanic-serving higher education institutions, assessed student learning and engagement. One of their findings was that historically underrepresented students performed as well as students of other backgrounds in project-based learning, and the remote settings did not hinder them.

3.2.3 Simplifying Curricular Complexity. Another issue in retaining diverse students is curriculum complexity, particularly compulsory courses where failure leads to exclusion from the program. Facilitating different paths into the degree program leads to more options for student success, which is advantageous for those who may have less experience coming into the degree or who may be drawn to less stereotypical subjects.

Slim et al. [107] frame degree attainment - arguably the most important measure of student success - as the degree to which students can successfully navigate the various requirements associated with their degree program. The more complex, demanding, and inflexible the requirements, the lower the likelihood of success.

Lionelle et al. [65] analyzed the relationship between the curricular complexity and the attainment of women in 60 CS programs. They found that the degree complexity and the factors that block and delay progression inversely correlate with the representation of women in the program. Core recommendations to reduce this dropout include: (1) providing increased flexibility in pathways to avoid

Table 3: Examples of EDI-Centered CS Curriculum Efforts: Curricular complexity

Project	Institution	Curriculum Impact	Year
Establishment of BA in CS: Removal of math & physics requirements	FIU	Enrollment with greater diversity, particularly resulting in the inclusion of Black students [99].	2022
Liberal Arts CS Curriculum	MI	Inclusion of Liberal Arts flexible curriculum and a process for adherence to CS Curriculum Guidelines. No discussion on under-representation [52].	2023
Modular TA-Oriented Curriculum	NEU	Modularity for better TA training and support. No evaluation results [79].	2023
Hidden Curriculum	UCSD	Created a peer-written hidden curriculum guide and generalized on learner-centred design workflow. Positive benefits among participants [80].	2023
CS Degree Attainability & Curricular Complexity Barriers	MI	Degree attainability [107], and curricular complexity, blocking factors and delay factors impacting representation of women in the program [65].	2024

FIU: Florida International University, **MI:** Multi-Institution, **NEU:** Northeastern University, **UCSD:** University of California San Diego

failure in a single course blocking an entire degree (a "choke point“), (2) minimizing the length of required course sequences, (3) offering flexibility for completing math prerequisites, and (4) communicating degree pathways clearly and early with students, providing sufficient support for them to plan their studies effectively.

Students may try to navigate curricular complexity through information from their peers. Nakai and Guo [80] developed guidelines for underrepresented students who do not have the sufficient exposure to the "hidden curriculum" (not explicitly taught rules, values and expectations) and face information asymmetry within the field of CS, to assist them professionally and emotionally.

Florida International University established a BA degree in CS that did not require Calculus and Physics to meet the graduation requirement, and was able to attract a more racially diverse population, although the increase was statistically insignificant for women [99]. Unfortunately, many studies on (reducing) curricular complexity are not explicit on its motivation or implications on EDI issues in computing education. Tychonievich and Sherriff [115] describe an eight-year curriculum overhaul at the University of Virginia that aims to reduce curricular complexity (among other improvements). Holland-Minkley et al. [52] present guidelines for the CS curriculum in the context of liberal arts. Muzny and Shah [79] introduce modularity in the curriculum for TAs. These studies do not, however, discuss the implications of the overhaul on EDI.

3.2.4 Creating Interdisciplinary Routes into Computing. Preconceptions and stereotypes around computer science abound and are a major reason for the lack of uptake among those that do not fit the stereotype [70, 71]. This is a significant problem for diversity, with groups such as women and students who are not white or Asian often feeling excluded. A potential solution to encourage such students to study computing is combining computing with subjects with higher diversity or reimagining CS courses around interests and subjects that are less prone to this kind of stereotyping.

Allowing students to integrate computing with other subjects — often referred to as a “CS + X” degree — is an approach that appears to have diversity implications. This can be implemented in different ways. Sometimes, the focus of an entire course is on integrating computing with another subject. The computing courses these students take contain significant computing material rather than just introductory material or computing used as a tool for facilitating other subjects, but are specifically created with these students in mind. Such an approach is more resource-intensive but has been successfully implemented in several places — for example, the Computing in the Arts degree at the College of Charleston [9], which integrates computing with an art specialism, focusing on multiple synthesized courses. A six-year longitudinal study shows that this course graduates are 45.6% female. In the ten-year study of Dodds et al. [33], they found that their biology-themed CS1 was popular among women students. Harvey Mudd College has a *green* route option for CS1, which allows students without prior experience to choose an introductory course co-designed with biosciences, as well as the *gold* route — an option for novice programmers taught entirely within CS. They found that the green route was more attractive to women and that those who chose this route were just as likely to choose to major in CS as those who chose the more traditional gold route [3]. Interdisciplinary minors, such as the Applied Computing for Behavioral and Social Sciences minor at San Jose State University, were offered by many institutions. The courses for these minor programs commonly exhibited a higher representation of historically underrepresented students, namely female, Black, or Hispanic students, as they are intended to expand pathways to computing [24, 25, 60].

The University of Michigan College of Literature, Science and the Arts (LSA) is exploring ways in which computing curricula for liberal arts students can be co-designed with liberal arts students and faculty so as to bring in the expertise of computing scientists whilst centering the needs and interests of the target audience [45].

More commonly, students can build combined degrees with modules from computing and from one or more other subjects. The computing modules that they take are not particularly adapted to them and will be done with other students who are pure computing students. An intermediate approach is to allow students to primarily take courses from each of their subjects, but to add one or more synthesized courses aimed specifically at these joint major students, and/or to require a capstone project that focuses on both subjects.

Unsurprisingly, the extent to which an interdisciplinary degree can attract diverse students is dependent on what the “X” is. Evidence around this mostly focuses on female students: combined degrees where the “X” is female-dominated, such as the arts [7, 123] and media are much more gender-balanced, reporting a gender balance close to equal, more than double the percentage of women in straight computing degrees.

Table 4: Examples of EDI-Centered CS Curriculum Efforts: Interdisciplinary routes into computing

Project	Institution	Curriculum Impact	Year
CS + X degree	UIUC	Increased female enrolment in many cases — e.g., from 22.5% to 39%[18]	2022
Computing in the Arts degree	Charleston	6-year study shows over 46% of graduates are female [9]	2018
Teaching CS through a media lens	GATECH	Predominantly female enrollment in interdisciplinary minors [95]	2004
Bioscience-themed CS1 option	HMC	More attractive to women and same percentage in those choosing to major in CS as standard CS1 [3]	2012
Interdisciplinary minor	SFSU SJSU and others	Greater representation of female and URM students in courses for these minors, and support student learning and experience in computing courses [24, 25, 60]	2014 2018 2020

UIUC: University of Illinois Urbana-Champaign, **Charleston:** College of Charleston, **GATECH:** Georgia Institute of Technology, **HMC:** Harvey Mudd College, **SFSU:** San Francisco State University, **SJSU:** San José State University

Northeastern University has a very broad program of combined degrees, with 44.6% of computing students pursuing a combined major. 39% of these students are female, as opposed to 22.5% in single-major computing [18]. However, they did not find any racial and ethnic diversity between joint- and single-major computing students. There is some evidence that CS + X degrees where the "X" is a male-dominated subject, such as physics, are even more male-dominated than pure computing students, although the data is sparse and dated [19].

3.2.5 Challenging Stereotyping through Diverse Examples and Applications. Students from underrepresented groups have reported a lack of interest and engagement with CS material tailored to a more stereotypical CS population [27]. For example, women more often value social and artistic expression, often not captured as part of the diversity of interests in computer science [72]. Also, women in CS are more likely than men in CS to be undecided in their career plans [56]. One approach to improving interest and sense of belonging is to broaden the types of examples and applications used in the curriculum. The National Centre for Women & Information Technology's (NCWIT) *Make it Matter*[1] states that *"all students are more motivated, perform better, and and more likely to persist when they can see how a lesson connects to their experiences, interest, goals, and values"*. They provided four recommendations: (1) use meaningful and relevant content, (2) make interdisciplinary connections to CS, (3) address misconceptions about the field of CS, and (4) incorporate student choice, which is also used as the framework for the Morrison et al. [76] systematic literature review exploring the evidence of impact for initiatives that have been introduced to broaden participation in CS. This section will explain these types of interventions and who they serve.

Women: Women studying CS tend to find educational activities themed around people more appealing than the educational activities themed around things [44, 67]. Media computation-themed CS1 courses are more interesting to women students compared to traditional CS1 courses [95]. Using gender-neutral language and adopting project topics that pique women's interest improves the sense of belonging for women in CS programs [121]. Mckeever and Lillis [74] found that women were more attracted to hybrid-type computing problems, such as those related to media, business, art, management, and more.

Research has shown that some elective courses in the CS curriculum are more popular with women, including mathematical and statistical foundations, human-computer interactions, society, ethics, and professionalism [16].

Underrepresented Race/Ethnicity Groups: Barretto et al. [10] found that students of underrepresented race/ethnicity were approximately six times less likely to take a traditional ML/AI course than those from the majority race/ethnicity. Also, students of underrepresented races and ethnicities are more likely to report interest in the social, cultural, and political impacts of ML/AI rather than the technical aspects of ML/AI.

Culturally responsive pedagogy is a relatively new field aimed at improving the achievements of students from underrepresented racial and ethnic groups [42]. The work of Magerko et al. [66] set out to teach CS1 coding concepts in Python via a learning environment developed called Earsketch, which combines programming and music production. They found improvement in CS1 content knowledge and all psychosocial constructs for all students, including women and students from underrepresented race/ethnic groups [66]. Similarly, Emdin et al. [35], promote using a third space, where classroom and students' emotional experiences are integrated, such as hip-hop-based STEM instructional interventions, as both pedagogy and therapy for Black students.

Culturally Responsive Computing (CRC) examines the cultural relevance of the curriculum and promotes equity through culture-inclusive teaching practices [34]. This results in the inclusion of vernacular and civic knowledge, brings socio-economic relevance to the curriculum, and improves student performance and engagement. Adopting CRC has the potential to increase the participation of students from historically marginalized backgrounds. In LilyPad, Arduinos with Native American youths utilize sewable microcontrollers to create traditional textiles [54]. Davis et al. [31] introduced a study on Culturally Situated Design Tools (CSDT), which teaches CS1 concepts with cultural contexts, such as African American cornrow braiding, or Native American, African American and Appalachian quilting traditions. Lastly, *Impossible Project*[2] developed a two-week anti-racist CS curriculum at the University of Buffalo. The group landed on illustrating racial bias in tech through a case study of PredPol, a predictive policing company that utilizes an algorithm to predict crime.

[1]https://ncwit.org/engagement-practices-framework/

[2]https://www.daliamuller.com/impossible-project

Table 5: Examples of EDI-Centered CS Curriculum Efforts: Challenging Stereotyping through Diverse Examples and Applications

Project	Institution	Curriculum Impact	Year
Earsketch	K12	Pre/post survey and interviews show improved performance and attitudinal constructs across all gender and ethnicities [66]	2016
Hip-Hop Based STEM	K12	Creating a learning space integrating classroom and students' emotional experience increased minority youth's engagement with STEM education [35]	2016
CS through traditional crafting	K12	Challenges and benefits discussed but outcomes not evaluated [54]	2014
Cultural Computing Curriculum	K12	Providing cultural contextual learning improved students' performance but did not change their attitude to culture or CS [31]	2019
The Impossible Project	Buffalo	No impact reported (Anti-Racist CS Curriculum, US Child Welfare Case Study)	2021

K12: Kindergarten through twelfth grade, **Buffalo:** University of Buffalo

Universal Design for Learning: Universal Design for Learning (UDL) focuses on the importance of providing multiple examples of concepts to make them relevant to individuals with diverse backgrounds, gender, ethnicity, race, experiences, etc. [21]. UDL is a holistic approach to educational design that provides guidelines for course and curriculum design to accommodate the widest range of learners, including students with disabilities [98]. Allen et al. [2] used UDL principles to improve CS1 for English Language Learners and showed how it improved their performance when compared to stronger English speakers.

3.2.6 *Cultural and Social Justice Competency.*

One problem with diversity in computer science is that very few people, especially students, are educated in the non-technical aspects of the diversity problem. Diversity, Equity, and Inclusion initiatives are becoming more common in higher education in the US [30] and beyond, most targeting faculty and staff. The lack of such initiatives for students means that these problems are not well understood by the student body. For example, there is work indicating that many CS students (especially men) are unaware of the gender gap in CS or do not think the gender gap is a problem [126]. Therefore, CS students need education and training in EDI issues.

In this section, the initiatives we discuss focus both on educating CS students on how they can be part of the solution to create a more equitable environment within the field, and on how, in their future careers, they can be part of creating technology that addresses the needs of diverse and disparate groups of people rather than catering exclusively for "typical" technology users, as technology has had a history of doing.

The integration of EDI in CS curriculum is needed, the gap still exists, with limited commitment in implementation and integration [122]. This section will describe interventions that add EDI curriculum to CS programs as standalone courses or EDI-flavored modules infused into technical courses.

Cultural Competency: The work from Washington [119] determined that cultural competence should be a non-technical focus for computing departments to ensure companies have talent pools that better understand the importance of EDI issues [119]. Washington [119] argued that with no formal courses that focus on the non-technical issues affecting marginalized groups and how to address and eradicate them, students are indirectly taught that the current status quo in computing departments and industry is not only acceptable but also unproblematic. The authors also believe more students from marginalized groups can be retained in the major through the addition of the cultural competence curriculum. The cultural competence curriculum described in a subsequent paper [120] focuses on exploring EDI challenges in computing through an introduction and discussion of identity as a social construct, its impact on computing, and the resulting impact on technology.

The Cultural Competence in Computing (3C) Fellow program has expanded this work to countless universities that have initiated cultural competence curricula into their CS programs' curricula. One such example is a work from Kong and Pollock [59] using the 3C program to develop a course called "Race, Gender, and Computing" at the University of Delaware's CS program.

Justice-Oriented Curriculum: Justice-centered computing focuses on the pressing ethical need for technology to benefit all, recognizing that historically, technology has been driven by the requirements of an industry that benefits the few. This is often at significant cost to other groups, such as via environmental degradation. Justice-focused computing requires both that technology is driven by diverse people who have an understanding of the needs of a broad range of people and that all computer science students are explicitly educated in the creation of inclusive technology [62]. The research looked at the top 20 universities in the US and found that less than 50% of the top-ranked universities had courses that explicitly focused on ethical issues, gender, race, accessibility, and environmental justice and that for those that did, it accounted for less than 5% of the curriculum [77].

Other EDI Curriculum: Artze-Vega et al. [4] provides an equity-minded teaching guide for teachers to enable all students, especially historically underserved students, to have an equal chance at success. They include steps to guarantee both relevance and rigour of course content, transparency of expectations for success, support and scaffolding to facilitate learning, and the use of the right tools and platforms. Gachago et al. [39] encouraged an equity-oriented approach to the process of learning design itself, noting the benefits of ensuring that those who participate in curriculum design are themselves from a diverse set of backgrounds. They advocate for learning design processes that are characterized by collaboration and are cognizant of the traditional power structures that have existed in this space. Peña et al. [92] also addressed the importance of gender diversity in both teaching staff and decision-making roles in faculty.

Table 6: Examples of EDI-Centered CS Curriculum Efforts: Cultural and Social Justice Competency

Project	Institution	Curriculum Impact	Year
Cultural Competency CS Course	MI	Discusses the need for computing departments to focus on cultural competence to create more inclusive environment beyond universities [119]	2020
CS through inclusive design	HSI	Improved outcomes & retention, especially amongst minoritised students [41]	2024
Case studies to teach AI ethics	Waseda	Students developed deeper understanding of ethical issues in AI, but study does not provide an insight with respect to EDI [50]	2022
Race, Gender, and CS Course	UDEL	Piloted a class discussing how technologies do not benefit women and minorities as much and observed positive reflections among students [59]	2023
Gender-inclusive teaching	UPC	Suggestions for gender-inclusive teaching. Not evaluated [92]	2021
GenderMag	MI	Framework for software engineers to develop gender-sensitive software without specific knowledge of gender issues; improved student grades [22, 41]	2016 2024

MI: Multi-Institution (International), **UDEL:** University of Delaware, **UPC:** Universitat Politècnica de Catalunya, **HSI:** Hispanic-Serving Institution, **Waseda:** Waseda University

Inclusive Design in Software An important aspect of cultural and social justice is training students to design more inclusive software. A curriculum emphasizing inclusive design aims at making students aware of technology barriers impacting numerous minoritized populations and teaching them how to design better solutions. Oleson et al. [85] explained that this is no easy task because students (and instructors) are often trained to *engineer* programs rather than *design* them, and remain unaware of their blind spots, believing that some “common sense” is all that they require to design inclusive software. The same research shows that students encounter several hurdles on the path to truly inclusive software design, including difficulties taking the perspectives of others or sticking to conventional design patterns even when those are at odds with the design goals they are trying to achieve.

To address this issue, Burnett et al. [22] introduced GenderMag, a systematic method to evaluate software from a gender-inclusiveness perspective. The method was tested in an academic environment to facilitate the teaching of inclusive design principles and was found to be well-accepted by instructors and effective at improving students’ grades, as well as instructors’ evaluations [41]. They found that incorporating inclusive design improved course outcomes and students’ retention, especially for marginalized groups [40].

Patel et al. [87] argued that the disconnect between computer science core curricula and the specific courses that cover societal aspects, such as ethics, EDI, and design for human interaction, trains a workforce that is not ready to implement inclusive software.

Ethics: Inclusive design in software is an ethical imperative as technology increasingly dominates our lives, but this is only one of the ethical dimensions that computing students need to be aware of. Important ethical questions include the development of hardware from materials, the mining that leads to ecological damage, the reliance of many forms of technical development on underpaid and ethically questionable labour in the developing world, and the climate implications of large data storage. An increasingly pressing ethical area concerns the increasing use of AI, especially Large Language Models (LLMs) such as ChatGPT.

Aler Tubella et al. [1] explored the existing literature around ethical AI and interviews 11 experts across five European countries to help define educational strategies, competencies and resources needed for the successful implementation of Trustworthy AI in Higher Education (HE). The main concerns raised were technical robustness and safety, as well as privacy and data governance.

While most of the existing literature on teaching AI focuses on the need to teach diverse students from all disciplines to improve their understanding of AI, the imperative to educate CS students - the future developers of AI - about AI and ethics is well-evidenced. McDonald and Pan [73] explored how CS students are not inclined to think deeply about ethical considerations until prompted to do so, and even then, their thinking about ethics exposes biases. Jang et al. [53] discovered gendered differences in attitudes to fairness in AI, with female students having stricter ideas of fairness, as did students with more education in AI.

Hishiyama and Shao [50] discussed the efficacy of using case methods to teach AI ethics. This approach appeared to hold the students’ interest and lead to a deeper understanding of the issues involved.

3.2.7 Summary and Highlights of the Gaps. Six themes emerged from our exploration of the literature on curriculum interventions related to the benefit or intent benefit to a more diverse student population. We presented the interventions that align with each theme in this section and summarized them in Tables 1-6.

Many reported interventions only happened in K12, and have not been adopted in undergraduate curriculum. Some did not observe or did not report long-term impact. Many interventions are not widely adopted or reproduced. Also, we suspect many valuable efforts and experiences in this domain may not have been published because they are in their early stages. Furthermore, existing interventions tend to address only a single dimension of diversity. Consequently, existing literature lacks guidelines for a holistic, comprehensive, EDI-centered curriculum design, considering the race, gender, ethnicity, age, access needs, and socioeconomic backgrounds of student populations. There is a pressing need to explore the design of an equity-minded CS curriculum and to determine an effective approach to achieve it.

In section 4, we show whether and how existing curricula from selected universities worldwide align with the presented themes.

Table 7: Literature Themes and Curriculum Analysis

Literature Themes	Curriculum Analysis
Levelling the playing field to help students with different technical backgrounds	Multiple Versions of CS1
Integration of capstones, research, and industry projects in the curriculum:	Capstone, Thesis, Research, and Internship
Simplifying curricular complexity	Curriculum Complexity
Creating interdisciplinary routes into computing:	Interdisciplinary Options
Challenging stereotypes through diverse examples and applications	Content Level Interventions
Cultural and social justice competency	

4 Curriculum Analysis

4.1 University Selection Criteria

To answer our research questions **RQ1** and **RQ2** concerning the approaches to addressing EDI in the undergraduate computer science curriculum, and the outcomes of these EDI-centered curriculum efforts in terms of attracting or retaining more historically marginalized students in the field, we conducted an extensive curriculum analysis. Our curriculum analysis expands our literature review by exploring public information from institutions worldwide on their related curriculum efforts.

We chose three major geographical regions to focus on: Australia, North America and Europe. We made this choice because the vast majority of the literature was based in these regions, and because of their cultural similarities. For each selected region, we compiled the 10 highest-ranked research-oriented CS programs according to the *Times Higher Education World University Rankings 2024* for Computer Science[3], and analyzed publicly available data about the CS curriculum for each university in this list. The rationale behind this choice is that the effect on public perception of these universities' high rankings would result in them receiving a more diverse pool of applicants to their CS programs due to students' desire to pursue degrees in prestigious institutions [5, 6], and that their high reputation would lend weight to their impact in defining curriculum in other universities in their region.

Where publicly available, we have collated information about the student population for each of these programs, such as the percentage of women, international students, or BIPOC students enrolled. This information is presented in Tables 8–11.

We also included an additional selection of institutions from the US, including Liberal Arts Colleges with high enrollment of women, Historically Black Colleges and Universities, Hispanic Serving Institutions, and Women-only Institutions. In the paper, we will use the term *equity-focused institutions* to refer to this particular group. These institutions play a distinctive and crucial role in the landscape of higher education in the United States. They also represent an opportunity to study EDI-related curriculum efforts beyond highly ranked research institutions due to their reputation and the selective student population they serve. Furthermore, guidelines for a liberal arts perspective have been discussed recently. Holland-Minkley et al. [52] proposed a six-step integrated design process to align common CS curricular recommendations such as CS2023 [86] with liberal arts programs. Guzdial and Evrard [46] identified three themes of computing use in liberal arts and sciences: computing for discovery, expression, and justice. These themes were used to develop a new program and courses to support computing education for liberal arts and sciences students to support non-EDI intervention better.

Table 7 illustrates how our approach to data collection from publicly available curriculum information was influenced by the literature review. We defined categories for which we could reasonably assume to find data in publicly available sources for each institution (e.g., department websites, course catalogs, syllabi), linking them to the six themes outlined in Section 3.1. In one case, we

[3]https://www.timeshighereducation.com/world-university-rankings/2024/subject-ranking/computer-science

Table 8: Percentage of women and international enrolments in undergraduate CS programs in Australian universities

University	Women[1]	International[1]
ANU	24%	47%
Macquarie	19%	24%
Monash	24%	69%
Swinburne	18%	61%
Adelaide	15%	48%
Melbourne[2]	N/A	N/A
UNSW	23%	21%
Queensland	24%	41%
Sydney	24%	53%
UTS	24%	27%

[1] **Source:** https://www.education.gov.au/higher-education-statistics/resources/student-enrolments-pivot-table-2022, 2022.
[2] **Note:** Data for this university is unavailable as CS is only offered as a major within other degrees, not as a separate degree program.

Table 9: Percentage of women in undergraduate CS programs in Canadian universities

University	Women	Source
McGill	25%	[217]
McMaster	20%	[213]
SFU	20%	[215]
Montréal	N/A	N/A
UQAM	N/A	N/A
Alberta	N/A	N/A
UBC	32%	[216]
UofT	23%	[213]
Waterloo	28%	[213]
Western	18%	[213]

Table 10: Percentage of women and international enrolments in undergraduate CS programs in European universities

University	Women	International	Source
EPFL	16%	58%	[208]
ETH Zürich	18%	43%	[218]
ICL	25%	61%	[210]
KULeuven	N/A	N/A	N/A
PSL	N/A	N/A	N/A
TUM	18%	40%	[211]
UCL	22%	61%	[209]
Oxford	N/A	56%	[214]
Cambridge	20%	38%	[212]
Edinburgh	N/A	48%	[209]

Table 11: Percentage of women, international, and BIPOC enrolments in undergraduate CS programs in US universities

University	Women[1]	International[1]	BIPOC[1]
Caltech	36%	8%	70%
CMU	46%	24%	61%
Cornell	38%	16%	59%
Harvard	40%	16%	60%
MIT	37%	15%	62%
Princeton	40%	17%	60%
Stanford	34%	15%	64%
Berkeley	28%	18%	69%
UIUC	25%	26%	47%
UWashington	46%	23%	55%
Carleton	39%	17%	24%
Harvey Mudd	38%	10%	63%
Swarthmore	37%	25%	41%
Wellesley	100%	12%	77%
Howard	44%	31%	66%
Morehouse	0%	14%	86%
Spelman	100%	0%	100%
UCI	23%	21%	67%
Austin	24%	8%	69%

[1] **Source:** https://collegefactual.com, 2020–2021.

decided to group two themes ("Challenging stereotypes through diverse examples and applications" and "Cultural and social justice competency") under the common umbrella of "Content Level Interventions": this was done because, for both themes, data was to be collected through the analysis of courses and related high-level available information (e.g. course catalogs descriptions) at selected institutions. Further analysis may have made possible keeping the two themes separate, but this would have required access to more detailed information about the courses, like the type of examples or assignments used, which was beyond the scope of this study. Instead, our data collection focused on information about degree requirements, entry-level programming courses, and course-level considerations. We also looked at additional curriculum-related factors impacting student populations, such as entry pathways, interdisciplinary options, and specific content topics.

In the following, we present our analysis results in the five categories for institutions in different regions and describe differences in regional approaches and overall findings.

(1) Multiple Versions of CS1
(2) Capstone, Thesis, Research and Internship Options
(3) Curriculum Complexity
(4) Interdisciplinary Options
(5) Interesting Courses

4.2 Multiple Versions of CS1

We reviewed the details of each university's CS1 offering as provided on each institution's website, to see whether multiple versions (streams) were offered, and if so, for what purpose. The reasons we found for offering more than one version of CS1 include:

- having a course for an advanced stream;
- having a bridge course, to provide additional support to students without prior programming experience and perhaps allow students to enter the program who may not otherwise meet the minimum requirements; or
- other reasons, such as wanting to provide different programming language options.

A summary of this data is given in Table 12. In some cases, there was more than one additional stream in a particular category; in these situations multiple checkmarks have been recorded in the relevant column. Under the column "Other" we also included separate CS1 courses delivered by the department hosting the CS major but tailored specifically for the needs of students enrolled in a degree other than CS. These courses do not affect students already in a CS stream, but they may provide opportunities for students in other majors or undeclared majors to explore CS as an option.

4.2.1 Australian Institutions. In Australia, the approach varies. ANU, Monash University, and the University of Sydney all provide advanced CS1 streams. The University of Adelaide and UTS provide separate streams, but only for students enrolled in degrees other than CS (Bachelor of Information Technology and Bachelor of Engineering respectively). Macquarie University and UNSW both provide bridging streams. In the case of Macquarie this is done through a separate division of the university (Macquarie University College) dedicated to providing pathway programs to students who would not otherwise meet the admission requirements of the university. In the case of University of NSW, this is through their "Gateway Admission Pathway", in which students who do not meet the minimum requirements for admission are allowed to enroll in a Diploma, the completion of which is equivalent to the first year of studies in the corresponding degree.

No Australian universities in our analysis provide more than one additional CS1 stream, and Swinburne, University of Melbourne, and University of Queensland do not provide any additional options.

4.2.2 Canadian Institutions. In Canada, three universities provide bridging courses: the University of British Columbia, the University of Toronto, and the University of Waterloo. Two universities provide advanced streams: Western University and the University of Waterloo. The University of Waterloo is the only university in this region to provide three streams designed for three different levels of prior experience — a bridging stream, a standard stream, *and* an

Table 12: Curriculum Analysis: CS1 Streams

University	Advanced	Bridging	Other
ANU	✓		
Macquarie		✓	
Monash	✓		
Swinburne			
Adelaide			✓
Melbourne			
UNSW		✓	
Queensland			
Sydney	✓		
UTS			✓
McGill			✓✓✓
McMaster			
SFU			✓✓✓
Montréal			
UQAM			
Alberta			
UBC		✓	
UofT		✓	
Waterloo	✓	✓	
Western	✓		✓✓
EPFL			✓
ETH Zürich			
ICL			
KULeuven			✓✓
PSL			
TUM			✓
UCL			
Oxford			
Cambridge			
Edinburgh			
Caltech	✓		
CMU		✓✓	
Cornell			✓
Harvard			✓
MIT	✓		
Princeton			
Stanford	✓	✓	✓
Berkeley			✓
UIUC			
UWashington	✓	✓	
Carleton			
Harvey Mudd			✓
Swarthmore			
Wellesley	✓		✓
Howard			
Morehouse			
Spelman		✓✓	
UCI			
Austin	✓		✓

All universities offer at least one CS1 stream. This table indicates the number of additional streams.

advanced stream. This particular 3-stream approach was only seen in North American universities, not in Europe or Australia.

Some Canadian universities also have separate streams based on reasons other than the level of prior programming experience. For example, McGill University has four CS1 streams, depending on whether the student is taking the unit as part of a standard CS degree, a life sciences degree, a physical sciences degree, or a spatial sciences degree. Simon Fraser University provides two additional options with different programming approaches (scientific programming or systems-oriented), as well as a stream designed only for Engineering students. Western University provides separate streams for Data Science and Engineering students.

4.2.3 *European Institutions.* Most of the universities we analyzed across Europe only listed a single version of CS1 for students to enroll in. Three universities offer CS1 courses for students outside the CS major: KU Leuven provides separate streams for Engineering students and for those taking CS as part of a Math, Physics, or Biomedical Science program; TU Munich offers an introductory CS course for students in math; and EPFL offers to all first-year Bachelor students a course to introduce them to fundamental ideas of computer science and communication systems. This last course in particular is delivered to about 1500 students in seven different instances, and the programming language is chosen to meet the needs of each particular section. None of the European universities indicated on their website any separate CS1 streams provided for advanced students or any bridging courses for CS1.

4.2.4 *US Institutions.* Among universities in the US, four offer advanced CS1 streams (Cal Tech, MIT, Stanford, and University of Washington), and three offer bridging streams (Carnegie Mellon, Stanford, and University of Washington). Stanford University and the University of Washington offer bridging *and* advanced streams. In addition to these three streams, Stanford offers a separate fourth pathway for non-CS students.

Carnegie Mellon also adopts the three-stream approach based on the level of prior experience—however, two of the three would be considered bridging streams (rather than one bridging and one advanced). The standard Carnegie Mellon CS1 course assumes students have completed and obtained a score of 5 on the AP Computer Science A exam. A bridging course exists for those without that requirement, which is nonetheless described as "fast-paced". A second, "slower-paced" bridging stream is provided for students who want additional support.

Harvard and Berkeley both have unique approaches that are worth noting. Harvard offers two different streams based on language choice (the standard stream teaches C, while an alternative, more "problem-solving based" stream using Python is also offered). Berkeley offers a separate "self-paced" stream that can be completed over one or two semesters. This stream is intended to attract both students with more programming experience *and* those with no prior experience at all; the idea is that students can work through the required content at a pace that is comfortable for them, whether that be faster or slower than the average student.

University of Illinois Urbana-Champaign offers only a single version of CS1, while Cornell University offers a separate stream for Engineering students only.

4.2.5 Equity-Focused Institutions. Among the nine equity-focused US institutions we considered, only UT Austin and Wellesley provide an advanced stream for CS1, and only Spelman College provides a bridging stream. In fact, Spelman provides two options (one of which is more business-oriented in its approach) for students who do not pass the computing literacy exam that allows them to directly enroll in the standard CS1 stream.

Harvey Mudd, Wellesley, and UT Austin provide streams with a specific applied focus in certain areas, designed for students taking courses in those fields: Harvey Mudd for computational biology students, Wellesley for those enrolled in the natural sciences, and UT Austin for applications in science and business.

The remaining five universities listed in this section (Carleton, Swarthmore, Howard, Morehouse, and UC Irvine) offer only a single stream for CS1.

4.3 Capstone, Thesis, Research, and Internship

As outlined in Section 3.2.2, exposure to research and industry experiences has a predominantly positive impact on students, increasing their motivation to enrol and complete a major, and minoritized groups in CS stand to benefit the most. [96]. For these reasons, the inclusion of these experiences in CS programs should be encouraged. We reviewed the graduation requirements and the course catalogue of the selected CS programs to understand whether each includes forms of extensive project-based work, such as the completion of a capstone project, a thesis, or internship opportunities (in some regions referred to as Co-Ops).

Institutions in different regions use different terminologies to describe their courses and graduation requirements. To be consistent in our analysis, we adhered to the following definitions:

- **Capstone:** culminating team project course, to be completed in the last year. It often involves collaboration with clients or industry partners. Other project-based courses are not included in this definition.
- **Thesis:** individual study or project conducted at the end of a bachelor's degree, presented through the submission of a paper or dissertation and, frequently, an oral presentation. It may have either a research or applied focus.
- **Research:** any course specifically dedicated to the completion of a research study. Often, these courses are planned by an instructor and a single student, with the student receiving credits for working on a mutually agreed research question.
- **Co-Op:** opportunity to work with industry partners, with the goal of gaining experience relevant to one's field. To be included in our analysis, the co-op must be mediated – not just advertised – by the university, and not independently sought by the student.

The results are summarized in Table 13 and discussed in the following paragraphs.

4.3.1 Australian Institutions. Nine out of the ten programs included in our analysis offer or require students to complete a capstone course. The wide availability and consistent requirement of capstone project courses align with a major goal of these programs — helping students develop practical skills to thrive in the industry.

Seven institutions offer a 4-year Honours CS degree (as an alternative to the standard 3-year CS degree), and each one of these

Table 13: Curriculum Analysis: Research & Industry Exposure

University	Capstone	Thesis	Research	Co-Op
ANU	Req.Op.	Req.Op.		Req.Op.
Macquarie	Req.			
Monash	Req.Op.	Req.*		Req.Op.
Swinburne	Req.	Req.*		Ava.
Adelaide	Req.	Req.*		Ava.
Melbourne	Req.			
UNSW	Req.	Req.*		
Queensland		Req.*		Ava.
Sydney	Req.	Req.*		
UTS	Ava.	Req.*		Ava.
McGill			Ava.	Ava.
McMaster	Req.			Ava.
SFU	Ava.		Ava.	Ava.
Montréal		Req.*		Req.
UQAM				Ava.
Alberta	Ava.			Ava.
UBC		Req.*	Ava.	Ava.
UofT	Ava.		Ava.	Ava.
Waterloo	Ava.			Ava.
Western		Ava.		Ava.
EPFL	Req.		Ava.	
ETH Zürich		Req.		
ICL		Req.		
KULeuven		Req.		
PSL		Req.		
TUM	Req.	Req.		
UCL		Req.		
Oxford		Req.		
Cambridge		Req.		
Edinburgh		Req.		
Caltech		Ava.	Ava.	
CMU		Req.*	Ava.	
Cornell	Req.		Ava./Req.*	Ava.
Harvard		Req.*	Ava.	
MIT				
Princeton		Ava.	Req.	
Stanford	Req.Op.	Req.Op.	Req.Op.	
Berkeley		Ava.*	Ava./Req.*	
UIUC	Req.		Ava.	
UWashington	Ava.	Req.*	Ava.	Ava.
Carleton	Req.		Ava.	
Harvey Mudd	Req.		Ava.	
Swarthmore	Req.	Ava./Req.*	Ava.	
Wellesley		Req.*	Ava.	
Howard	Req.			Ava.
Morehouse	Req.		Ava.	
Spelman	Req.	Req.*	Ava.	
UCI	Req.	Req.*	Ava.	
Austin		Req.*	Ava.	

Ava. = Available. Req. = Required. Req.Op. = Required Option (student chooses one). Ava.* = Available for Honours Program. Req.* = Required for Honours Program.

honours degrees requires a thesis course in which students complete a substantial research project. Although this represents an opportunity for students to engage in research, its relegation to 4-year honours CS degrees could have EDI implications, as honours degrees require separate application processes and one more year of study. These additional time and monetary investment may deter underrepresented students from reaching these research opportunities.

As described so far, most Australian institutions we reviewed do not offer much flexibility regarding the capstone and thesis project courses, which are usually required by the respective programs. However, some programs do provide some flexibility. Monash University requires students to choose between a team project and industry placement options for their 3-year CS degree. ANU asks students in their 4-year CS to complete either a research project course, a software engineering team project course (similar to a capstone), or an internship, making it one of the most flexible program among all the Australian institutions in terms of the choice of experiential learning opportunities.

4.3.2 *Canadian Institutions.*

Half of the ten Canadian institutions have at least one capstone course available. However, almost all the programs do not require students to complete the capstone project, except McMaster University, where the CS program is within the Faculty of Engineering. It is common practice for undergraduate Engineering programs to require students to complete a capstone project in teams. A few Canadian institutions have a thesis course available. In particular, UBC and Université de Montreal offer an honours CS program requiring each student to complete the honours thesis course, while students at Western University can include a project with thesis as one of their courses.

Other research opportunities are available outside of thesis courses. Several institutions (McGill, SFU, UBC, UofT) offer CS courses that allow students to engage in research projects. They come, however, with a caveat: even if the course is in the online catalogue, it may not be regularly offered to the students. The availability of research project courses is highly dependent on the availability of faculty members to supervise these projects. Understanding the actual availability of these research project courses is out of the scope of this work. Other undergraduate research opportunities are available as separate programs rather than courses. Several Canadian institutions on our list, including the University of Toronto, the University of Waterloo, UBC, and Simon Fraser University, have various programs that allow undergraduate students to work on research projects with faculty members. Some examples are the Research Opportunities Program program at the University of Toronto, the Undergraduate Research Assistantship (URA) program at the University of Waterloo, and the Undergraduate Student Research Awards (USRA) program at UBC and Simon Fraser University.

4.3.3 *European Institutions.*

Nine of the ten European institutions on our list require students to complete a substantial individual project in their final year of studies. Five of the nine institutions (University of Cambridge, ETH Zurich, Technical University of Munich, KU Leuven, and Paris Sciences Lettres) refer to this project as a thesis/dissertation, while the rest refer to it as a substantial individual project. EPFL is slightly different, requiring students to complete a substantial project in a team rather than individually. They also have an optional research project course. And TU Munich requires, on top of a thesis, the completion of a "Praktikum", an extensive team project comparable to a capstone. With these two exceptions, other research opportunities outside of the required thesis are not included in the program.

4.3.4 *US Institutions.*

Nine of the ten institutions on our list offer or require a capstone course, a dissertation, or a research opportunity, with the sole exception of MIT, where we could not find any any of these options for the undergraduate CS degree. Based on their website, MIT seems to have a different approach, offering more courses with significant project components.

The US institutions on our list tend to refer to the capstone courses as team projects. Four out of the ten institutions have at least one team project course available, with two of them (Cornell and University of Illinois Urbana-Champaign) requiring it to complete the undergraduate CS degree. However, the emphasis on team projects in US institutions is less than that of Canadian institutions, and even less than Australian institutions.

More than half (7/10) of the programs have at least one senior thesis option. However, the programs differ by whether the senior thesis is available and/or required for the major or the honors programs. Caltech, Princeton and Stanford offer, but do not require, the senior thesis option for the CS major students. UC Berkeley provides the senior thesis option for their honors students but does not require it for degree completion, while it is required for honors students at Carnegie Mellon University, Harvard and University of Washington.

Research based courses are available or even required at nine out of the ten institutions. When marked as available, the same caveat outlined for Canadian institutions remains: supply depends largely on the availability and interests of faculty members, who act as supervisors.

Among the institutions selected, Stanford stands out as providing possibly the most flexible program in terms of project and research opportunities. Their program has three options for the major: a team project, a senior thesis, and an independent research project. Each student in the major must complete one of the three options for their degree.

4.3.5 *Equity-Focused Institutions.*

Capstone courses are much more common amongst the equity-focused US institutions than among their larger counterparts, with team-based or research-based capstones required in seven of the nine institutions on our list. Of those, Morehouse, Carleton, UC Irvine and Harvey Mudd all require two semesters of team projects. Two institutions offered no capstone project: UT Austin and Wellesley. UT Austin does offer elective project-based courses but not as part of the capstone. Wellesley offers elective research courses (independent studies).

None of the equity-focused US institutions requires a thesis for their degree, though about half of the schools (5/9) require a thesis in order for students in their honors programs. Swarthmore allows students to complete a thesis outside of the honors program, but this is not required.

All but one of the equity-focused institutions made research (outside of the capstone) available to students, a significant effort in making research opportunities available to undergraduates, even though a research course is never required.

4.3.6 Co-op and Internship Options. Co-op opportunities allow students to get exposure to industry-relevant topics and help them better understand the application of CS topics and their implications. We examined the course catalogues of all institutions and looked for course descriptions that allowed students to engage in internships or co-op education. The last column in table 13 lists the results.

Our analysis revealed that all the Canadian and most of Australian institutions on our list offered credits to students for industry-related experiences. In contrast, only a few institutions in the US and no institutions in the Europe offer for-credit internship courses.

4.4 Curriculum Complexity

We start by defining a few quantities that we will analyze. We will put the courses required for a CS degree into three categories:

- **Required Core CS Credits**: These are specific CS, Mathematics, and Statistics course credits required by a CS degree.
- **Required Non-CS Credits**: These credits are not CS, Mathematics, or Statistics and are required by the CS degree. These courses are sometimes called general or breadth courses that every student in the faculty/university must complete for their Bachelor's degree.
- **Required CS Elective Credits**: These are CS credits that a student must complete for a CS degree. However, students can choose these credits from a list of specific CS courses.

In addition, we count **the credits of CS elective courses** offered by the department. This number gives us a sense of the number of possible courses that a student can choose to complete their CS elective credits.

Having defined the terms above, we will measure the curriculum complexity for each CS degree using the three metrics below.

- We define **Degree Percentage** below to measure the proportion of required core CS courses in a CS degree.

$$\text{Degree Percentage} = \frac{\text{Number of Required Core CS Credits}}{\text{Number of Credits for the CS Degree}}$$

- We define **Required Percentage** below to measure the proportion of required CS core courses among all the CS courses for the CS degree.

$$\text{Required Percentage} = \frac{\text{Number of Required Core CS Credits}}{\text{Number of Required and Elective CS Credits}}$$

- We define **Elective Flexibility** to measure students' choices when selecting CS electives for their CS degree. First, we compute the ratio between the number of credits for CS electives available to students and the number of CS elective credits required by the program. However, comparing this ratio across institutions may be misleading: a program that allows students to pick 1 out of 4 possible electives would receive a higher score than a program that lets students choose 25 out of 50 possible electives. Therefore, we multiply this ratio by the percentage of CS elective credits students are allowed to take (or 1 - Required Percentage).

$$\text{Elective Flexibility} = \frac{\text{Number of CS Elective Credits Offered}}{\text{Number of CS Elective Credits Required}} \times (1 - \text{Required Percentage})$$

Note that the metrics above report the number of credits rather than the number of courses. Even within an institution, courses may not earn the same number of credits. For example, a course with an attached lab may earn four credits rather than three, or a required course may only meet for half of the semester and earn half of the credits of a typical course. Therefore, data from different institutions are more comparable when we view them through the lens of credits rather than courses.

Moreover, we chose to report percentages rather than the absolute number of credits, because the total credits required by a CS degree vary significantly across institutions. For example, the University of Toronto requires 20 credits, whereas the University of Edinburgh requires 480 credits. Reporting percentages allows us to compare the CS curricula at different institutions fairly.

The results are summarized in Table 14 and discussed in the following sections.

4.4.1 Australian Institutions. Looking at degree percentage, we can divide the Australian Institutions into two groups. At the Australian National University, the University of Adelaide, and the University of Technology Sydney, the CS degrees have a large percentage of CS courses. In particular, the CS programs at the Australian National University and the University of Technology Sydney comprise almost all CS courses. This does not mean that students cannot take courses outside of CS. Instead, the program requires very few non-CS courses to complete the degree. For the other programs, the percentage of core CS courses varies from 67% to 75%. CS courses still dominate these programs since CS courses take up at least 2/3 of the degree.

The required percentage for the Australian programs ranges from roughly 1/3 to 2/3. The Australian National University and Macquarie University programs offer considerable flexibility, where only 1/3 of the courses are required. On the other end, for the programs (Monash University, Swinburne University, University of Melbourne, and University of Sydney), 2/3 of the courses are required.

Regarding elective flexibility, the University of Technology Sydney has much greater elective flexibility than other Australian programs. Of the remaining programs, roughly half (Macquarie University, Swinburne University, University of New South Wales, University of Queensland) have an elective flexibility of 2.8 to 3.9. The other half of the programs have an elective flexibility of less than 2.

4.4.2 Canadian Institutions. For degree percentage, we can divide the Canadian programs into three categories. The University of Montréal and the University of Quebec in Montreal do not require students to take non-CS courses to complete their CS degrees. Next, McGill University, McMaster University and Simon Fraser University require a small percentage of non-CS courses for their programs, ranging from 17% to 25%. Finally, the remaining Canadian institutions are similar, requiring students to take 40% to 55% non-CS courses to complete their CS degrees. Compared to the Australian programs, Canadian programs have more institutions requiring students to take more non-CS courses. Half of the Canadian programs require students to take at least 40% non-CS courses, whereas all the Australian programs we analyze require at most 33% non-CS courses.

The required percentage for the Canadian programs ranges from 35% to 80%. Roughly half of the Canadian programs (McMaster University, University of Montréal, University of Quebec in Montreal,

Table 14: Curriculum Analysis: Complexity

University	Degree %	Req'd %	EF
ANU	100%	34%	1.88
Macquarie	67%	31%	2.81
Monash	69%	64%	1.91
Swinburne	75%	67%	3.91
Adelaide	88%	43%	1.79
Melbourne	71%	65%	0.88
UNSW	75%	58%	3.42
Queensland	67%	50%	3.06
Sydney	75%	67%	1.83
UTS	97%	48%	8.06
McGill	78%	35%	1.32
McMaster	75%	80%	0.50
SFU	83%	58%	1.52
Montréal	100%	80%	1.27
UQAM	100%	73%	1.22
Alberta	45%	56%	0.78
UBC	53%	72%	0.59
UofT	60%	60%	4.25
Waterloo	56%	73%	3.56
Western	52%	77%	2.86
EPFL	72%	71%	0.95
ETH Zürich	92%	92%	0.93
ICL	94%	77%	0.70
KULeuven	73%	100%	N/A
PSL	79%	100%	N/A
TUM	77%	89%	3.16
UCL	71%	100%	N/A
Oxford	100%	63%	0.70
Cambridge	100%	67%	2.58
Edinburgh	79%	71%	0.44
Caltech	39%	62%	3.09
CMU	51%	70%	4.08
Cornell	52%	62%	1.14
Harvard	62%	50%	3.13
MIT	95%	65%	1.95
Princeton	35%	73%	3.36
Stanford	32%	44%	4.31
Berkeley	40%	58%	4.48
UIUC	66%	71%	2.98
UWashington	40%	54%	3.29
Carleton	31%	82%	0.42
Harvey Mudd	41%	82%	1.10
Swarthmore	31%	80%	2.60
Wellesley	31%	60%	2.50
Howard	69%	82%	0.69
Morehouse	46%	84%	0.55
Spelman	53%	81%	0.56
UCI	62%	61%	2.14
Austin	42%	52%	3.20

University of British Columbia, University of Waterloo, and Western University) have more than 2/3 required courses. Moreover, the programs at Simon Fraser University, University of Alberta, and University of Toronto have 56% to 60% required courses. Finally, the program at McGill University only specifies 35% of the courses to be required. Compared to the Australian programs, the Canadian programs are somewhat less flexible. Half of the Canadian programs have more than 2/3 required courses. In contrast, every Australian program has less than 2/3 required courses.

Regarding elective flexibility, the University of Toronto leads the Canadian programs with the largest flexibility, followed by the University of Waterloo and Western University. The remaining Canadian programs have an elective flexibility of less than 2. The elective flexibility in the Canadian programs is similar to that of the Australian programs (not counting the University of Technology Sydney as an outlier).

4.4.3 European Institutions. The degree percentages of the European programs are generally high. Every program has at least 71% core CS courses. In particular, the programs at the University of Oxford and the University of Cambridge do not require students to take non-CS courses. Similarly, the CS programs at ETH Zürich and Imperial College London have 92% to 94% core CS courses. The remaining programs have degree percentages ranging from roughly 70% to 80%. In general, the European programs have higher degree percentages than the Australian and Canadian programs.

Similar to the pattern for degree percentage, the required percentages of the European programs are also relatively high. Three programs (KU Leuven, PSL and University College London) do not leave any room for elective courses. ETH Zürich and TU Munich allow students to fulfill 8% to 11% of the program requirements with elective courses. The remaining five programs have more flexibility and allow students to choose electives for 23% to 37% of the program.

The European programs we analyze tend to have lower elective flexibility than the Australian and Canadian programs. Half of the European programs have an elective flexibility of less than 1. TU Munich and the University of Cambridge have elective flexibility of 3.16 and 2.58, respectively. Our elective flexibility metric is undefined for KU Leuven, PSL and UCL because the programs do not require students to take elective courses.

4.4.4 US Institutions. At a glance, the US programs offer a lot more flexibility than those in the previous sections. Regarding degree percentage, almost all the US programs require students to complete the requirements with at least 34% non-CS courses. The MIT program is an outlier requiring 95% of the program to be CS core courses. On the other extreme, five (or half of the) US programs require less than 40% of the degree to be CS core courses, leaving much room for non-CS courses.

The required percentages of the US programs are similar to that of the Canadian programs, ranging from 44% to 73%. The programs at CMU, Princeton, and UIUC are the least flexible (having a required percentage of 70% to 73%). The programs at Stanford, Harvard and the University of Washington are the most flexible, allowing students to complete 46% to 56% of the requirements with electives.

The US programs generally have greater elective flexibility than those analyzed in previous sections. In particular, eight of the ten US programs have an elective flexibility of 3 or more. UC Berkeley, Stanford and CMU stand out by having an elective flexibility of at least 4. The programs at Cornell and MIT are the least flexible, with elective flexibility below 2.

4.4.5 Equity-Focused Institutions. The CS programs at the equity-focused US institutions have generally low degree percentages, meaning that they require students to take relatively more non-CS core courses to complete their CS degrees. Carleton College, Harvey Mudd and Wellesley tie for having the smallest degree percentage of 31%, requiring students to complete more than 2/3 of their degree with non-CS courses. Howard has the largest degree percentage of 69%, but still requires students to take non-CS courses to complete at least 30% of their degree.

The required percentages of the programs at the equity-focused US institutions are somewhat high, ranging from 52% to 82%. Six of the nine programs have a required percentage of 80% to 82%, which only leaves less than 20% of the program requirements to be electives. The remaining three programs allow students to use electives to fulfill 39% to 48% of the degree requirements.

The elective flexibility of the equity-focused US programs is somewhat lower than that of the other US programs but higher than the European programs. Four programs (Swarthmore, Wellesley, UC Irvine, and UT Austin) have an elective flexibility of 2 or more. Four programs (Carleton College, Howard, Morehouse and Spelman) have an elective flexibility of less than 0.7.

4.5 Interdisciplinary Options

As part of an effort to understand interdisciplinary routes into computing, we reviewed each selected institution's official website to investigate whether its CS department offers either an interdisciplinary major/programs or a specialization within their CS major curriculum. Note that we did not include double majors or double degrees that require additional years of study. In addition, we searched whether each institution offers CS minor programs, as they may contribute to broadening participation in computing. The results are summarized in Table 15.

4.5.1 Australian Institutions. We did not find interdisciplinary degrees to be common, and in cases where they do exist they are fairly limited. Monash University offers an interdisciplinary major with Business Information Systems and the University of Adelaide offers CS and Maths. The University of Melbourne is an interesting case, offering a Computing & Software Systems major as part of either the Bachelor of Science or the Bachelor of Design programs, with the required non-CS courses differing between the two.

Most Australian institutions offered specializations within their CS programs. While many of these specializations were in areas typically housed within the CS department (e.g., AI, Data Science, Systems and Architecture), Macquarie University and NSW also offered business-relevant specializations (e.g., business analysis, e-commerce systems) that are not typically housed within the CS department.

Table 15: Curriculum Analysis: Interdisciplinary Options

University	Interdisciplinary	Specializations	Minor
ANU		✓	
Macquarie		✓	
Monash	✓	✓	✓
Swinburne		✓	
Adelaide	✓	✓	
Melbourne	✓		
UNSW		✓	
Queensland		✓	✓
Sydney		✓	
UTS			
McGill		✓	✓
McMaster		✓	✓
SFU	✓	✓	✓
Montréal	✓	✓	✓
UQAM			
Alberta	✓		✓
UBC	✓		✓
UofT		✓	✓
Waterloo		✓	✓
Western	✓	✓	
EPFL		✓	✓
ETH Zürich	✓	✓	
ICL	✓	✓	
KULeuven		✓	
PSL	✓	✓	
TUM		✓	
UCL		✓	
Oxford	✓		
Cambridge			
Edinburgh	✓	✓	
Caltech		✓	✓
CMU	✓	✓	✓
Cornell		✓	✓
Harvard	✓		
MIT	✓	✓	✓
Princeton		✓	✓
Stanford		✓	✓
Berkeley			✓
UIUC	✓	✓	✓
UWashington		✓	
Carleton			✓
Harvey Mudd	✓		✓
Swarthmore			✓
Wellesley			✓
Howard			✓
Morehouse			
Spelman	✓	✓	✓
UCI	✓	✓	✓
Austin	✓	✓	✓

Availability of interdisciplinary majors/programs, specializations, and CS minor programs. A checkmark in the respective column indicates we found evidence for that offering.

Our analysis revealed that only a few institutions (e.g., Monash University, university of Queensland) among those selected were found to offer CS minor programs.

4.5.2 Canadian Institutions. All Canadian institutions have either interdisciplinary majors/programs or specializations, with the sole exception of UQAM. More specifically, five out of the ten selected Canadian institutions offer interdisciplinary majors/programs. Except for Western Univerisity, all other institutions in this region offering interdisciplinary majors/programs provide majors combined with other science majors. We believe this may be due to the fact that the CS major is offered within the Science Department in most Canadian institutions. A combined CS/Business major is the most common among other remaining options. Singularly, the University of Alberta allows combined CS programs with almost any disciplines.

In terms of specializations, seven out of ten Canadian institutions were found to have CS specializations. While most specializations are in areas typically housed within the CS department (e.g., Software Engineering, Natural Language Processing, Computer Systems, Computer Vision), several institutions (SFU, UQAM, and University of Waterloo) offer specializations that encourage students to take courses outside the CS department, including multimedia, business, and bioinformatics.

Except for UQAM and Western University, all universities in this region offer CS minor programs. Western University, however, offers an interdisciplinary minor in Digital Humanities.

4.5.3 European Institutions. About half of the European institutions offer interdisciplinary majors/programs in CS. Three institutions – the University of Edinburgh, University of Oxford, and Imperial College London – offer interdisciplinary programs, such as CS and Physics, CS and Philosophy, and Mathematics and CS. The University of Cambridge used to offer interdisciplinary options like CS with Mathematics and CS with Natural Sciences, but these programs have been discontinued. Additionally, two of the five institutions on our list – ETH Zürich, and Paris Sciences Lettres (PSL) – offer interdisciplinary programs. For example, ETH Zürich offers a Computational Science and an Engineering major. Interestingly, PSL does not offer a standalone, non-interdisciplinary Bachelor's degree in CS. Instead, PSL provides interdisciplinary programs related to CS, such as Organizational Computer Science (Informatique des Organisations) and Data Sciences, Arts, and Cultures (Sciences des données, arts et cultures). These programs combine CS with diverse fields like mathematics, economics, history, and literature.

Almost all European institutions offer specific CS specializations at the Bachelor's level - with the exception of the Universities of Oxford and of Cambridge, although they do provide multiple electives in the final year. Examples of available specializations include: ETH Zürich's CS program, which offers specializations in areas typically housed within the CS department (e.g., systems & software engineering, information & data processing, theoretical computer science), while its interdisciplinary CS program (Computational Science and Engineering) offers specializations that encourage students to take courses outside the CS department, from Astrophysics to Computational Finance; the École Polytechnique Fédérale de Lausanne, which offers multiple tracks that encourage students to take courses outside the CS department, such as data science, machine learning and bio; and PSL which within their Organizational Computer Science program offers two third year tracks, one in computer methods for business management, and one in data science.

Only one of the ten European institutions – the École Polytechnique Fédérale de Lausanne (EPFL) – offers a minor option in Computer Science, which may reflect the different structure European degrees.

4.5.4 US Institutions. US institutions offer several interdisciplinary options in CS, such as computational biology or computational mathematics. Notably, the University of Illinois Urbana-Champaign has a broad selection of majors with its blended degree program called CS+X, where students may choose to study one of 14 majors while concurrently pursuing CS. These degrees typically do not require additional coursework or semesters but require completing a different set of courses tailored to the interdisciplinary major. Carnegie Mellon University (CMU) is also an interesting case as it offers interdisciplinary CS majors that encouraged students to take courses outside the CS department, such as CS and the arts, and music and technology.

Most US institutions offer specializations or tracks for CS major students, as well as minor options allowing students from other departments and colleges to study CS. While specializations typically focus on areas within computing, there are some exceptions. For example, Cornell University requires CS students to specialize in fields outside of CS (i.e., external specialization), asking students to take nine or more credit hours of upper-level courses (3000+ level) in another discipline.

4.5.5 Equity-Focused Institutions. Less than half of the equity-focused institutions offer interdisciplinary majors/programs or specializations. Harvey Mudd is the only Liberal Arts College that offers joint majors with STEM subjects (e.g., CS+Physics, Mathematical and Computational Biology). Spelman College and UC Irvine offer CS & Engineering programs, while UT Austin offers five undergraduate degrees in CS. Among these five degrees, the Bachelor of Science and Arts in CS requires students to take relatively fewer CS courses, allowing them to take a broader range of liberal arts and humanities courses than the Bachelor of Science in CS program.

Three (UT Austin, UC Irvine, and Spelman College) of the four equity-focused institutions offering interdisciplinary majors are also able to offer specializations for their CS majors. Most of the specializations in UT Austin and UC Irvine are in areas typically housed within the CS department (e.g., Big Data, Computer Systems, Cybersecurity, Algorithms), Having said that, UT Austin's 'Social Impact Stamp' specialization is noteworthy. It requires students to take ethics courses relevant to CS and complete a Social Impact Capstone, allowing them to apply their CS understanding in practice. None of the liberal arts colleges provides specializations, which could be related to having relatively fewer CS electives available (refer to section 4.4). Among the three historically black colleges and universities (HBCUs), Spelman College is the only institution that offers a flexible 'Computer Science Plus X' concentration. This 'Computer Science Plus X' concentration allows students to replace some of the electives and Database Management Systems course with a group of courses from another major (e.g., Economics, Environmental studies), contingent upon approval by the department chair or the junior advisor.

Except for Morehouse College, all equity-focused institutions offer CS minor programs.

4.6 Content Level Interventions

The literature identifies several types of courses that explicitly direct students' attention to make an equitable use of the skills they learn in class and most frequently integrate EDI components. AI has rapidly advanced and become more ubiquitous, and more institutions have seen reason to add ethical considerations to their AI curriculum. HCI courses have begun to more frequently adopt practices such as UDL and CRC, as these approaches connect easily with relevant course concepts. And some schools now offer courses on equitable and responsible use of technology, as part of EDI efforts or to address the reality of computing being entwined with an ever growing number of facets of society.

We surveyed the course listings of our selected institutions for instances of these types of courses, specifically looking for AI courses with a focus on ethical applications, inclusive HCI courses, and other courses with a focus on ethics or social justice (this last category does not include any AI-related courses, as they are covered by the first grouping). For AI and HCI courses, we did not consider 'human-centered' to be sufficient justification for inclusion; we required the course to be specifically focused on ethics, justice, society, culture, or accessibility (and related terms). The course search may have been impacted by the fact that some upper-level electives are sometimes listed with nonspecific or provisional names (e.g. 'Topics in...'); as the scope of our search was limited to available online catalogues, we did not investigate these courses, and based our tallies solely on course titles and descriptions. Table 16 summarizes our findings. As there are less evident regional differences, the rest of this section is organized by course content instead of by region.

4.6.1 Ethical AI. AI courses oriented toward social, ethical, and cultural dimensions have been previously identified as more attractive to students from underrepresented groups [10]. AI courses with a specific focus on ethics are quite common, but they are in no way ubiquitous.

In Australia, six of the ten schools offer an explicitly ethics-focused AI course. Some schools offer these courses outside of Computer Science, e.g. the University of Sydney's "Philosophy of AI" is taught in the Philosophy department, and UTS's "The Ethics of Data and AI" is taught by the Communications department. Similarly, six of the ten Canadian schools offer an ethics-focused AI course. Again, while most of these courses are taught in the Computer Science department, two schools, McGill and Western University, offer courses through the Philosophy department. The University of Toronto's "Ethics of Data Science and Artificial Intelligence" course is available only as a half-credit offering rather than as a full course, though they offer "Intelligence, Artificial and Human" as a first-year course. The University of Alberta's open first-year *AI Everywhere* certificate is specifically framed as interdisciplinary, reflecting the University-wide research area of 'AI4Society'. Eight of the ten European schools offer ethics-focused AI courses, though

Table 16: Curriculum Analysis: Course-Level Interventions

University	Ethical AI	Inclusive Design HCI	Responsible Computing
ANU			
Macquarie	✓		
Monash	✓		
Swinburne			✓
Adelaide		✓	
Melbourne	✓		
UNSW			✓
Queensland	◇		
Sydney	◇		
UTS	◇		✓
McGill	◇		✓
McMaster			
SFU			✓
Montréal			
UQAM			
Alberta	✓		
UBC	✓		✓
UofT	✓		✓
Waterloo	◇		✓
Western	◇		✓
EPFL	◇		✓
ETH Zürich	◇		
ICL			✓
KULeuven	◇		✓
PSL	◇		
TUM			
UCL	◇		
Oxford	◇		✓
Cambridge	◇	✓	✓
Edinburgh	◇		
Caltech	◇		✓
CMU	✓		✓
Cornell	✓		✓
Harvard	✓		
MIT	✓	✓	✓
Princeton	✓		✓
Stanford	✓	✓	✓
Berkeley	◇		✓
UIUC	◇		✓
UWashington	✓		✓
Carleton			
Harvey Mudd			✓
Swarthmore			✓
Wellesley			✓
Howard	✓		✓
Morehouse			✓
Spelman	◇		
UCI	✓	✓	
Austin	✓		✓

Institutions by region offering at least one course in the target areas. A check mark ✓ indicates the course was taught in the Computer Science department. A diamond ◇ indicates the course was taught outside the Computer Science department, for example in a Philosophy department or a Business Management department.

only UCL offered this course within the Computer Science department. All of the remaining schools offered such courses as part of other programs such as Law, Philosophy, or Digital Humanities.

All of US institutions offer an ethics-focused AI course with half of the schools settling on variants of "Ethics in Artificial Intelligence" as the course title. Though some of the courses are offered outside the Computer Science department (CMU, Caltech), it appears that these courses are available to all students. Cornell and Princeton offer courses focused more on the law than on the technology, e.g. Princeton's "Artificial Intelligence: Law, Ethics, and Politics". Only four of the nine equity-focused US institutions offer these courses. None of the liberal arts schools has such offerings, perhaps reflecting constraints due to the size of their curricula. Neither Howard's "Applied Machine Learning, Bias and Ethics" nor Spelman's "The Ethics of Virtual Reality and Artificial Intelligence" were offered within the Computer Science department. Austin stood out in this group with four separate courses on the subject, including "Intro to Ethical AI/Robotics" and "Fair Transparent Machine Learning".

It is worth mentioning that courses offered by other departments may be more difficult to access for CS students for various reasons (e.g. missing pre-requisites, credits not recognized for majors). Although these potential difficulties vary from course to course, we expect in-house courses to be, on average, more advantageous for CS students, and therefore preferable. With this in mind, the US region offers the best scenario for CS students interested in learning more on the topic of AI ethics.

4.6.2 Diverse & Accessible HCI. HCI courses have been highlighted as a receptive environment for CRC and UDL-related curricular interventions [40]. At our target institutions, content related to inclusive or culturally specific design was frequently found to be a small element of general HCI courses. A very limited number of schools offers targeted courses. The course at the University of Adelaide, "Human and Ethical Factors in Computer Science", is the only targeted course found at Australian universities. While many of the HCI courses offered at Canadian universities integrate ethics into a larger curriculum, none of these schools has courses with a specific focus on HCI ethics or inclusive design. At European institutions, only Cambridge's "Theories of Socio-digital Design for Human Centred AI" has a specific ethics focus. In the US, courses offered by Stanford and MIT target designing for accessibility and assistive technologies. UCI's "Ethics, Technology, and Design" is the only ethics-centered HCI course offered at the equity-focused US institutions.

4.6.3 Responsible computing. Students interested in pursuing careers with social impact may not view this as compatible with computing culture [55]. Some institutions have responded to this and other calls within the discipline for a more critical approach to CS education [58], by introducing targeted courses. A few Australian institutions feature such offerings, with UT Sydney proposing a slew of explicitly trans-disciplinary courses–a choice of which is required–such as "Shaping the Technologies That Shape Us" and "Technologies Re-imagined in a Complex World". Six Canadian institutions offer such courses, with some focusing on "Computing for Society" (e.g. McGill, University of Toronto). Half of the European institutions have courses focused on ethics and social justice. The courses at the University of Oxford and of Cambridge have a legislative focus. Nearly all of the US institutions and most of the equity-focused US institutions offer courses with relevant curricula. Caltech's "Data, Algorithms and Society" targets first years and Swarthmore's "Ethics and Technology" targets a mix of Computer Science and Philosophy students. Two of the HBCUs offer many courses in this area including Morehouse's "Data and the African Diaspora" and "Computer Ethics and Human Values", and Howard's "Computational Social Justice" and "Ethical & Social Impacts of Computing".

4.7 Discussion and Concluding Thoughts

4.7.1 CS1 Versions. When considering the four regions of highly ranked universities, the US institutions provide the most flexibility regarding offering multiple streams for CS1, while European institutions provide the least. Australian and Canadian institutions are similar regarding the number of institutions offering advanced or bridging CS1 streams.

Many universities provide additional streams solely to cater to specific non-CS degree types that are closely related to CS (such as Engineering). However, nothing in our literature review indicates that this particular approach to splitting CS1 has any particular EDI benefits. Few universities have reported offering different streams with different programming languages for CS1, except Simon Fraser University and Harvard.

4.7.2 Capstone, Thesis, Research and Internship Options. Different regions vary widely in terms of their capstone, thesis, and research course offerings. The Canadian and US universities offer a good degree of flexibility. On average, they offer more opportunities to engage in projects and research, but without making it a graduation requirement.

In contrast, the curricula of European and Australian institutions are more rigid, almost always requiring all students to complete a thesis (Europe) or a capstone project (Australia) for the major. Most Australian universities also require their honours students to complete a thesis.

All the Canadian institutions on our list provide co-op/internship courses for credit. In contrast, no institutions in Europe offer co-op/internship opportunities for credit.

4.7.3 Interdisciplinary Programs and Options. There were both similarities and differences across institutions from different regions in terms of the available majors/programs, specializations, and minors. Most Australian and European institutions on our list do not offer interdisciplinary programs, unless they require additional years of study. Most of these institutions also do not offer CS minor options. This may be due to the relatively shorter duration of an undergraduate degree in these regions. A key difference between Australian and European institutions is that most Australian institutions offer specialization programs at the Bachelor level, while most European institutions provide specializations primarily at the Master's level.

Most Canadian and US institutions, in contrast, offer interdisciplinary/joint options that do not require additional years of study. However, there are still some differences between Canadian and US institutions, particularly in terms of the fields with which these interdisciplinary/joint options are combined. For example, the majority of interdisciplinary programs in Canadian institutions focused on science majors. Business was one of the few options available outside of science for interdisciplinary/joint programs. In contrast, many US institutions offered a wider variety of interdisciplinary options in diverse fields, including education, music, and linguistics. In addition, most of these institutions also offer specializations, although predominantly in traditional CS areas. Business, bio-informatics, medical health informatics, and multimedia were among the few options available outside traditional CS disciplines. The availability of these interdisciplinary and specialization options, such as CS + Biology, Business, HCI, and Multimedia, is encouraging, as they may attract more diverse demographics into CS [7].

Many equity-focused US institutions, particularly liberal arts colleges and HBCUs, seem to have limitations in offering interdisciplinary or joint programs, specializations, and CS minor programs. This may be due to limited resources or a lack of available CS electives. Although offering new CS courses and specializations may not be technically feasible, institutions like Harvey Mudd College and Spelman College demonstrate more practical ways to broaden participation in computing by combining CS with existing, more gender-balanced majors (e.g., Biology). In general, offering CS minor programs and interdisciplinary majors/programs or specializations in conjunction with more gender-balanced fields beyond Business and Biology could be a strategy to further broaden participation in computing for institutions worldwide.

4.7.4 Curriculum Complexity. Prior literature finds that women and underrepresented groups are more likely to be included in programs that allow interdisciplinary routes to computing, for example CS+X. It is reasonable to think that institutions that allow students to pursue their interests outside of computer science would be more welcoming for these students. Our curriculum analysis indicates that institutions in the US offer CS programs that require the smallest percentage of required courses, allowing students to take more courses outside of their major area.

Additionally, prior literature on curricular complexity finds that providing multiple pathways through the major removes the barriers associated with compulsory courses, which may block a student's progress. Therefore, we might expect that schools with a lower percentage of courses required by the major to be more welcoming of women and underrepresented students. In Canada, Australia and the US, students are on average asked to complete a smaller number of mandatory courses, leaving room for them to decide how to complete their major. Conversely, in Europe students are asked to complete a high number of required courses.

A high proportion of elective courses can help creating alternative pathways, but only if there is actually a good number of options to chose from. At institutions with higher values of elective flexibility, we might expect underrepresented students to find more courses they are interested in taking and more reasons to stay connected to the computer science major.

4.7.5 Content Level Interventions. The survey of the courses in selected institutions confirms that AI and course with a focus on responsible use of technology are sites of curricular interventions. Many of the courses were notably reserved for upper level or even graduate students; a much smaller proportion were available to take early in a degree program. Many offerings across categories were interdisciplinary and cross-listed – often with psychology, philosophy, or information studies – suggesting this as a fruitful path for future interventions.

On the other hand, HCI courses with a focus on inclusive design were less frequently found. It is possible that this is a limitation of our search, and these topics are actually being taught in HCI courses without being mentioned in the title or in the short description of the course. If this is not the case, this could identify a fertile ground for further course development, as it was recently the case for AI courses with an ethical emphasis.

4.8 Threats to Validity

We collected the data for this section from information available on the public-facing websites of each institution. While most universities provide reasonably detailed information regarding their offerings (e.g., CS1), there may be gaps in our information where institutions choose to share additional information internally.

Language barriers, cultural differences, and variances in terminology across the different institutions can, at times, make it challenging to consistently interpret what constitutes a "standard" pathway of entry into a CS program instead of one that would be considered a bridge course.

5 Students: Surveys and Interviews

5.1 Student Survey

To get a better understanding of the impact of curriculum design choices on undergraduate students, we decided to conduct student surveys and interviews. We designed and implemented an anonymous survey to better understand the students' experiences with their CS curriculum. We designed the survey questions for university students registered in a computer science program. We designed and reviewed the questions multiple times collaboratively. We first defined our goals and each person added their suggested questions, then we conducted two rounds of offline written feedback followed by online group discussion to review each question's scope, wording, and possible ambiguity, incorporating the feedback from educators in the working group with experience in each of our target region institutions.

The survey included 44 survey questions and one additional question to express interest in interviews (a total of 45 questions). This included 25 5-point Likert scale questions, three multiple choice questions, five multi-select questions, and 11 open-ended questions categorized into five categories: (1) observations of the participant of CS discipline, (2) participant's course experiences during CS program, (3) participant's curriculum experience during their CS program, (4) additional comments and suggestions, and (5) questions about the participants, including self-identification of gender, socialization, and accessibility needs. The complete list of Student Survey Questions can be found in Table 32. Because of the short study timeline, it was not possible to conduct a full reliability assessment of the survey questions. While we acknowledge that the lack of reliability assessments is a limitation, it was beyond the scope and timeline of this type of study.

Following our ethics approval guidelines, we found CS department contacts at each of the 49 universities included in our study. During the working group period, the survey link was sent to the CS department contacts included in our study, who were asked to disseminate it to students enrolled in the major. We have also disseminated the survey through social media, to increase our reach and include students from other universities.

To encourage more students to respond, answering was optional for all survey questions. Therefore, not all respondents answered all of the survey questions, and the number of respondents per question (*N*) varies. We determined it was preferable to receive partial feedback than having all questions require an answer and possibly discouraging students from answering at all. The student survey has received 613 responses, 345 of which were fully complete. The composition of the respondents can be summarized as follows:

- 302 students were second to fourth year computer science students. 103 students were a senior (about to graduate) computer science student, and 81 were first-year students. 47 respondents were computer science graduates. 38 were a university student majoring in a discipline other than computer science, but taking a computer science course. 6 were students trying to transfer out of a CS major, while 1 was interested in transferring in. 4 respondents were looking into applying to a CS program. 21 marked other (e.g. students in their second degree).
- 170 students (35.49% of those who responded) did not identify as a racialized person/person of color. 134 students (27.97%) did identify as a racialized person/person of color. 31 students (6.47%) marked that they were not sure and 10 (2.9%) marked they preferred not to answer, while 268 did not answer the question.
- When asked about their racial identity, 128 respondents selected White, 77 Chinese, 40 South Asian, 16 Mixed/Bi-Racial/Multi-Racial, 13 West Asian, 12 Southeast Asian, 10 Filipino, 8 Latin American, 4 Black, 4 Japanese, 4 Korean, 2 Arab, and 1 Indigenous/Aboriginal. 11 students indicated they preferred to self-identify. 15 students indicated they preferred not to answer.
- Respondents include 180 men (52.17%), 137 women (39.71%), and 18 self-identifying/non-binary students (SI/NB) (5.22%). 10 students indicated they prefer not to answer (2.9%).
- 274 students (79.42%) indicated they do not have accessibility needs. 21 students said they do have accessibility needs but are not registered with a Disability Centre. 36 students said they do have accessibility needs and are registered with a Disability Centre. 14 students indicated they preferred not to answer and 268 (4.06%) did not answer the question.
- 153 students said they took or are currently taking one or more CS course(s) as a required component for their major or graduation (and that the course was specifically required). 90 students said they took or are taking a CS course as a requirement, but that there are options to choose from. 84 students said some were required, and some where choice/elective. 14 said they are taking CS courses as electives an 7 students marked other (e.g. taking courses with the intention to transfer).

5.2 Student Survey: Quantitative Analysis

This section presents the results of the analysis of the Likert-scale and multiple selection questions in our student survey. During data validation and cleaning, we chose to include in our analysis all the responses entered by the 613 participants, including those who did not complete the entire survey, and those for whom we had to discard some qualitative answers (e.g. because they entered "N/A" or “I do not know”), as we determined this feedback to be valid and independent from the qualitative responses. Again, because students could skip questions, the number of respondents *N* varies.

The distribution of the responses to the Likert-scale questions is shown in Figure 1. The number of responses received in each question is different and is shown as *N* in the figure. Overall, the sentiment of the respondents appears to be positive, with high levels of agreement with statements such as "I believe Computer Science is important for society", or "I am satisfied with my decision to enroll in computer science". Conversely, respondents showed less agreement with statements like "I have difficulties following lectures because of the language and vocabulary used in my course and by my instructors", or "The computer science graduation requirements at my university are overwhelming". Possibly the biggest flag was raised by S1Q30, "I found some elements of my program’s curriculum discouraging", which found about half of the respondents in agreement, and supports the idea that a closer evaluation of CS curricula may be due.

We were particularly interested in examining possible differences in responses between different student demographics. We used the Mann-Whitney U test to evaluate differences in the students’ responses to 24 5-point Likert scale questions: each Likert scale answer was turned into an ordinal value (from 1 - Strongly Disagree/Never/Very Negative, to 5 - Strongly Agree/Always/Very Positive). To determine statistical difference between answers from groups of different genders, responses from each gender were compared to the other two groups (e.g. women with men and self-identifying/non-binary, men with women and self-identifying/non-binary, and self-identifying/non-binary with men and women). We also looked at differences between students who identified as racialized versus students who did not (S1Q41) and students with and without accessibility needs (S1Q40). For the latter, students were included in the group with accessibility needs whether or not they were also registered with their school’s Disability Centre. Responses with missing demographic information were excluded from the related stratified analysis. We also adjusted for multiple hypothesis testing using the Benjamini–Hochberg Procedure with a False Discovery Rate = 10%. Differences that were found statistically significant after correction are marked with an asterisk (*) in the tables included in this section, and will be discussed in the next paragraphs.

5.2.1 Shared concerns. Two statements showed significant differences in agreement between different demographics whether they were stratified by gender, race and accessibility needs: S1Q2, "I can be successful as a Computer Scientist", and S1Q11, "I feel I have things in common with other students in my course". For S1Q2, women respondents expressed lower levels of agreement (average of 3.89) than the other two groups combined, while the opposite

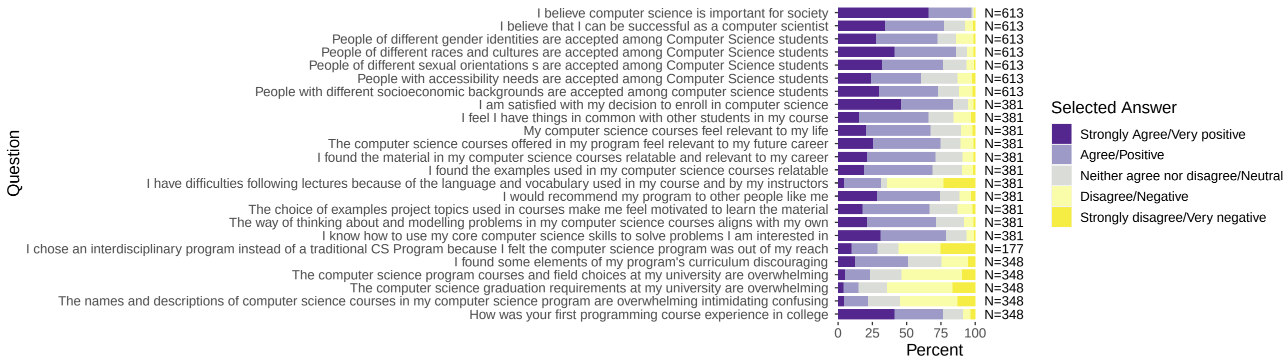

Figure 1: Distribution of responses to Likert scale questions. N indicates the number of nonempty responses per question.

Table 17: Response to Likert scale questions by gender (mean and standard deviation)

	Overall	Man		Woman		SI/NB		*N*
Question	Mean±SD	Mean	p-val	Mean	p-val	Mean	p-val	
S1Q1	4.61±0.62	4.69±0.53	0.017*	4.54±0.69	0.74	4.50±0.51	0.17	335
S1Q2	4.02±0.93	4.16±0.92	0.007*	3.89±0.95	0.007*	4.05±0.87	0.92	335
S1Q3	3.84±1.01	4.11±0.91	0.000*	3.52±1.11	0.000*	3.11±1.08	0.005*	335
S1Q4	4.01±0.90	4.23±0.86	0.000*	3.78±0.94	0.000*	3.78±0.94	0.22	335
S1Q5	4.20±0.86	4.36±0.86	0.000*	3.98±1.00	0.001*	3.78±1.06	0.05	335
S1Q6	3.69±1.03	3.92±0.99	0.000*	3.36±1.10	0.000*	2.89±1.28	0.012	335
S1Q7	3.89±1.01	3.99±1.05	0.016*	3.78±1.10	0.17	3.39±0.92	0.018*	335
S1Q9	4.23±0.89	4.26±0.90	0.37	4.23±0.85	0.60	4.00±1.14	0.41	335
S1Q11	3.62±0.99	3.72±0.94	0.03*	3.49±1.06	0.14	3.33±0.84	0.11	335
S1Q12	3.75±0.94	3.85±0.94	0.05	3.63±0.95	0.06	3.67±1.14	0.75	335
S1Q13	3.87±0.95	3.99±0.94	0.009*	3.75±0.92	0.015*	3.72±1.13	0.63	335
S1Q14	3.81±0.90	3.93±0.82	0.025*	3.72±0.91	0.05	3.61±1.09	0.46	335
S1Q15	3.76±0.90	3.86±0.85	0.07	3.70±0.92	0.16	3.50±1.10	0.33	335
S1Q16	2.25±1.00	2.18±0.97	0.22	2.36±1.00	0.029*	1.78±0.55	0.04	335
S1Q17	3.88±1.02	4.02±0.96	0.020*	3.77±1.04	0.04	3.67±1.19	0.44	335
S1Q18	3.69±0.96	3.75±0.94	0.32	3.61±0.94	0.12	3.89±1.02	0.24	335
S1Q19	3.83±0.88	3.92±0.87	0.016*	3.69±0.92	0.020*	3.72±1.02	0.80	335
S1Q20	4.02±0.88	4.10±0.81	0.09	3.92±0.96	0.18	3.83±0.98	0.38	335
S1Q29	2.57±1.31	2.67±1.35	0.34	2.45±1.23	0.36	2.62±1.77	0.92	170†
S1Q30	3.33±1.08	3.15±1.07	0.001*	3.53±1.06	0.004*	3.51±1.10	0.34	335
S1Q31	2.64±1.04	2.62±1.01	0.81	2.71±1.06	0.45	2.39±1.14	0.26	335
S1Q32	2.38±1.01	2.35±0.99	0.91	2.43±1.02	0.52	2.11±0.96	0.25	335
S1Q33	2.58±1.05	2.47±1.02	0.06	2.73±1.07	0.05	2.56±0.98	0.98	335
S1Q36	4.05±1.04	4.13±0.97	0.26	3.90±1.14	0.041*	4.55±0.51	0.05	335

The * indicates a significant difference between that group and the other two groups combined. †S1Q29 has a smaller number of respondents (N = 170) because 165 respondents selected the option "Not Applicable".

was true for men (4.16). Similarly, racialized and students with accessibility needs agreed less with this statement (average responses of 3.84 and 3.72, respectively) than non-racialized students (4.17) and students without accessibility needs (4.09). This evidence points at varying levels of confidence across different demographics, with underrepresented students being less certain about their chances of success.

Men and students without accessibility needs also agreed more with the statement "I feel I have things in common with other students in my course" than their counterparts did, indicating a higher sense of belonging for these demographics. Interestingly, racialized students agreed more with this statement than non-racialized students. Reasons behind this finding may include a stronger desire from racialized students to connect with their peers, but also a limitation of the survey: since the students were asked to self-identify, some racialized students may not necessarily be in a CS minority.

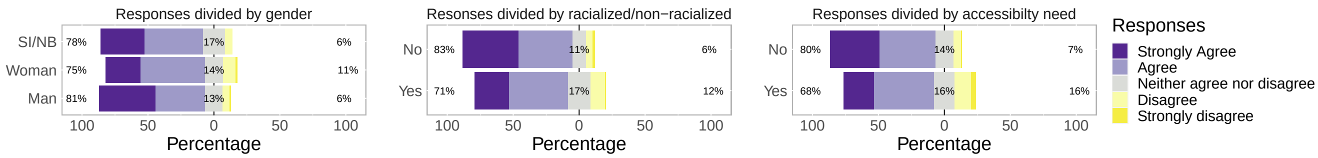

Figure 2: Distribution of responses to S1Q2: I believe that I can be successful as a computer scientist.

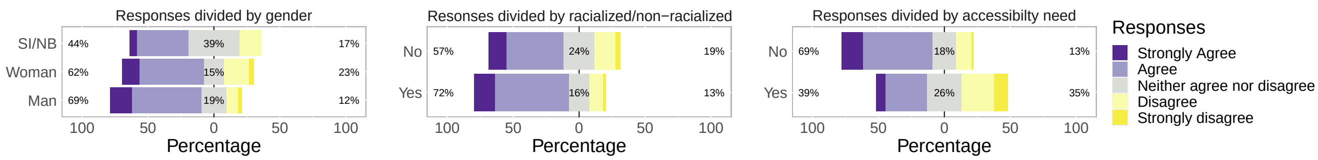

Figure 3: Distribution of responses to S1Q11: I feel I have things in common with other students in my course.

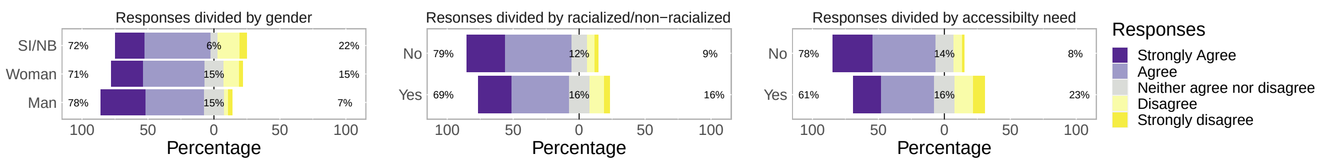

Figure 4: Distribution of responses to S1Q17: I would recommend my program to other people like me.

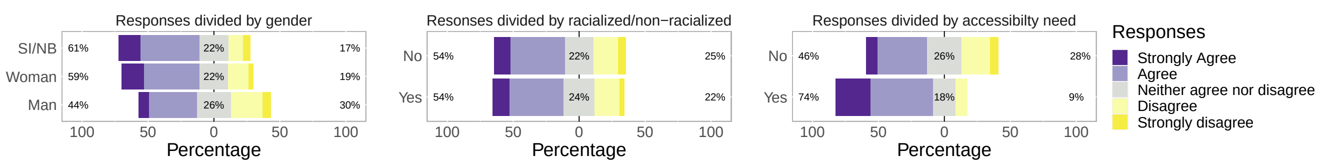

Figure 5: Distribution of responses to S1Q30: I found some elements of my program's curriculum discouraging.

5.2.2 Different sensitivities. When asked about the acceptance of different demographics in the field of computer science, the students' answers varied. As visible in Table 17, women and self-identifying/non-binary respondents expressed significantly less agreement with the statement "People of different gender identities are accepted among computer science students" (S1Q3), and women agreed less with the statement "People of different sexual orientations are accepted among computer science students" (S1Q4). Women respondents also agreed less with S1Q5 and S1Q6, "People of different races and cultures are accepted among computer science students" and "People with accessibility needs are accepted among computer science students", while self-identifying/non-binary students thought that people with different socioeconomic backgrounds may be less welcomed to CS (S1Q7). For all of these statements (S1Q3-7), men respondents expressed significantly higher levels of agreement. This was already partially observed in a previous publication by some members of this working group [5]: when asked about the acceptance in CS of students of different gender, sexual orientation, and race, men expressed more agreement than women and self-identifying/non-binary students.

When comparing students with and without accessibility needs (Table 19), the group of respondents with accessibility needs consistently expressed less agreement with statements concerning the acceptance of different demographics in CS (S1Q3-7). No significant differences were found, instead, in the responses of racialized and non racialized students to these questions.

These responses show some correlation with the answers to question S1Q10, where students were asked to rate the diversity in their departments. Options included "No diversity", "Some Diversity", "Diverse", "Very Diverse", as well as "I am not sure", "Prefer not to answer" or "Other". Because of the different scale, S1Q10 is not shown among other Likert scale questions in the summary tables, but the distribution of responses is visible in Figure 6. Scoring the answers "No diversity", "Some Diversity", "Diverse" and "Very Diverse" on a scale from 1 to 4, we found that men ranked the diversity of their department significantly higher (2.59±0.81) than women (2.40±0.78) and self-identifying/non-binary students (2.33±0.62). Similarly, students without accessibility needs had a more favorable opinion of the diversity of their department than those with accessibility needs (2.26±0.79 and 2.56±0.79, respectively).

These results show that, when asked about whether or not computer science is a welcoming field, students of various identities

Table 18: Response to Likert scale questions by racialized and non-racialized respondents (mean and standard deviation)

	Overall	Racialized	Non-racialized	p-val	*N*
S1Q1	4.61±0.62	4.57±0.68	4.65±0.55	0.41	304
S1Q2	4.02±0.93	3.84±0.96	4.17±0.92	0.001*	304
S1Q3	3.84±1.01	3.72±1.04	3.76±1.12	0.52	304
S1Q4	4.01±0.90	3.89±0.95	4.02±0.93	0.24	304
S1Q5	4.20±0.86	4.04±1.02	4.18±0.93	0.24	304
S1Q6	3.69±1.03	3.54±1.13	3.63±1.09	0.54	304
S1Q7	3.89±1.01	3.90±1.02	3.73±1.16	0.27	304
S1Q9	4.23±0.89	4.04±0.98	4.33±0.84	0.005*	304
S1Q11	3.62±0.99	3.72±0.93	3.48±1.03	0.026*	304
S1Q12	3.75±0.94	3.60±1.03	3.81±0.92	0.13	304
S1Q13	3.87±0.95	3.77±1.00	3.95±0.89	0.14	304
S1Q14	3.81±0.90	3.78±0.92	3.83±0.89	0.75	304
S1Q15	3.76±0.90	3.73±0.87	3.75±0.92	0.80	304
S1Q16	2.25±1.00	2.37±0.99	2.08±0.95	0.009*	304
S1Q17	3.88±1.02	3.74±1.10	3.96±0.95	0.08	304
S1Q18	3.69±0.96	3.66±1.00	3.68±0.94	0.96	304
S1Q19	3.83±0.88	3.77±0.93	3.82±0.89	0.70	304
S1Q20	4.02±0.88	3.87±0.90	4.05±0.90	0.07	304
S1Q29	2.57±1.31	2.81±1.33	2.35±1.36	0.032*	150†
S1Q30	3.33±1.08	3.40±1.05	3.35±1.10	0.78	304
S1Q31	2.64±1.04	2.69±1.09	2.59±1.03	0.49	304
S1Q32	2.38±1.01	2.37±0.99	2.33±1.03	0.69	304
S1Q33	2.58±1.05	2.64±1.04	2.54±1.06	0.52	304
S1Q36	4.05±1.04	3.95±1.07	4.15±0.98	0.10	304

The * indicates a significant difference between the two groups. †S1Q29 has a smaller number of respondents (N = 150) because 154 respondents selected the option "Not Applicable".

Table 19: Response to Likert scale questions by students with accessibility needs and students without (mean and standard deviation)

	Overall	w/ access. needs	w/o access. needs	p-val p-val	*N* *N*
S1Q1	4.61±0.62	4.54±0.63	4.64±0.60	0.22	331
S1Q2	4.02±0.93	3.72±1.06	4.09±0.90	0.012*	331
S1Q3	3.84±1.01	3.32±1.07	3.93±1.03	0.000*	331
S1Q4	4.01±0.90	3.68±0.93	4.08±0.91	0.002*	331
S1Q5	4.20±0.86	3.84±1.05	4.26±0.90	0.002*	331
S1Q6	3.69±1.03	2.91±1.21	3.81±1.00	0.000*	331
S1Q7	3.89±1.01	3.21±1.22	4.02±0.99	0.000*	331
S1Q9	4.23±0.89	3.96±1.08	4.28±0.85	0.05	331
S1Q11	3.62±0.99	3.00±1.13	3.70±0.92	0.000*	331
S1Q12	3.75±0.94	3.58±1.02	3.78±0.94	0.13	331
S1Q13	3.87±0.95	3.61±1.06	3.94±0.90	0.04	331
S1Q14	3.81±0.90	3.68±0.89	3.85±0.89	0.17	331
S1Q15	3.76±0.90	3.53±0.96	3.82±0.88	0.05	331
S1Q16	2.25±1.00	2.32±0.98	2.20±0.97	0.44	331
S1Q17	3.88±1.02	3.51±1.23	3.99±0.93	0.008*	331
S1Q18	3.69±0.96	3.47±0.97	3.75±0.94	0.05	331
S1Q19	3.83±0.88	3.49±0.98	3.89±0.87	0.006*	331
S1Q20	4.02±0.88	3.84±0.96	4.04±0.87	0.18	331
S1Q29	2.57±1.31	2.75±1.48	2.51±1.26	0.49	170†
S1Q30	3.33±1.08	3.91±0.89	3.20±1.07	0.000*	331
S1Q31	2.64±1.04	2.58±1.08	2.66±1.04	0.51	331
S1Q32	2.38±1.01	2.65±1.14	2.31±0.97	0.05	331
S1Q33	2.58±1.05	2.88±1.10	2.52±1.03	0.030*	331
S1Q36	4.05±1.04	4.03±1.12	4.09±1.00	0.92	331

The * indicates a significant difference between the two groups. †S1Q29 has a smaller number of respondents (N = 170) because 161 respondents selected the option "Not Applicable".

will give different answers. They also show that typically non-minoritized respondents (men, students who do not need accessibility accommodations) tend to have more optimistic estimates of how well accepted different demographics are in our field. This may be unsurprising, but it is worth mentioning to remind ourselves about the blind spots that occasionally occur when thinking about people with different backgrounds.

5.2.3 The impact of the curriculum. Significant differences across populations was found in some questions more closely tied to experiences with the curriculum. Women and students with accessibility needs expressed more agreement with the statement "I found some elements of my program's curriculum discouraging (S1Q30)", while men and students without accessibility needs agreed more with S1Q17, "I would recommend my program to other people like me". Additionally, women and racialized students reported having more difficulties following lectures because of the language and vocabulary used in their courses (S1Q16), pointing at the existence of a language barrier, and possibly at hidden assumptions about what vocabulary students should know. The results of the student interviews, discussed in Section 5.4, may help identifying starting points to address these inequalities.

5.2.4 Other differences. Table 17, 18 and 19 show other significant differences among the populations included in our analysis which were only found in particular groups. For example, racialized students were the only group to express significantly higher agreement than their counterparts with the statement "I chose an interdisciplinary program instead of a traditional CS Program because I felt the computer science program was out of my reach" (S1Q29). Students with accessibility needs were more frequently intimidated or confused by names and descriptions of computer science courses (S1Q33). And women were less likely to label their first programming course experience in college as positive (S1Q36). The survey highlighted several differences between the groups of respondents, each of them worthy of further investigation to identify the root causes. We hope our findings lay the foundations for future studies on these topics.

5.2.5 Multiple selection questions. The survey included four multiple selection questions, aimed at identifying students' preferred CS applications (Figure 7), knowledge areas (Figure 8), previous programming experience (Figure 10), and sources of information the help with course selection (Figure 9).

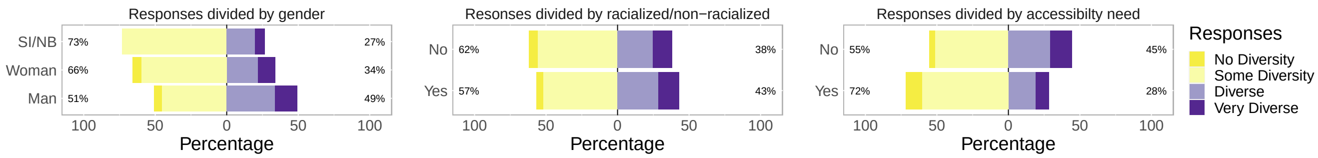

Figure 6: Distribution of responses to S1Q10: I rate the diversity of students in my computer science courses as...

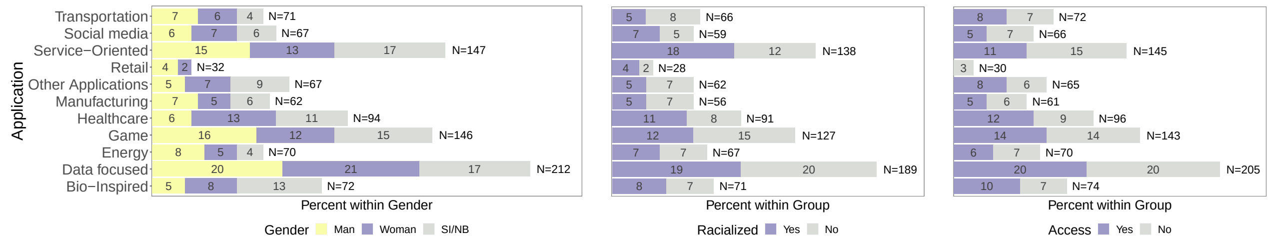

Figure 7: Distribution of responses for S1Q21: What types of computer science applications are you most drawn to?

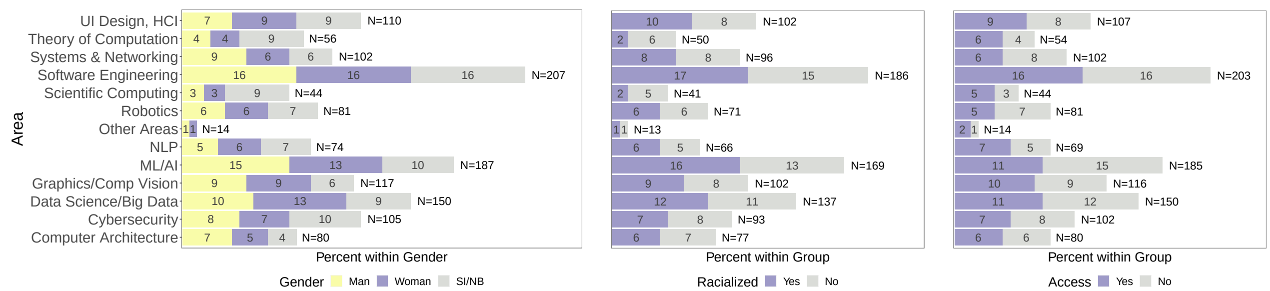

Figure 8: Distribution of responses for S1Q22: What areas of computer science are you most drawn to?

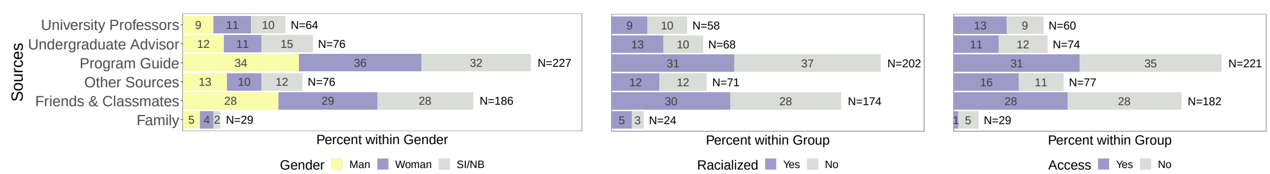

Figure 9: Distribution of responses for S1Q34: Where did you go to get information about course selection?

When looking at CS applications, data-driven applications were the most popular (selected by 186 respondents), followed by service-oriented applications (138) and games (127). The distribution of interest is fairly even across different demographics, although racialized students appear more interested than their counterparts in service-oriented applications, and a larger fraction of women and self-identifying/non-binary students is drawn to healthcare and bio-inspired applications.

With 186 preferences, software engineering emerged as the most popular area in computer science, followed by Machine Learning/AI (169 preferences), and data science/big data (137). Once again, preferences were equally distributed across demographics, with the exception of theory of computation being seemingly less appealing for racialized students.

When asked what sources they used to help them in their course selection, respondents indicated to favor their program guide (221 preferences) and the advice of their friends and classmates (182) above everything else. Only 29 students reported asking their family for help, making it the least popular option, even more so among students with accessibility needs: only 1% of this group reported relying on this source for help with course selection.

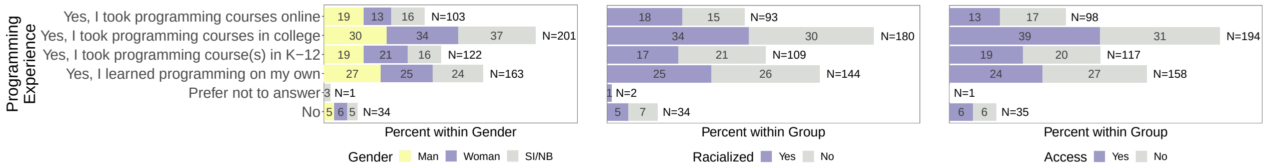

Figure 10: Distribution of responses for S1Q39: Have you had any programming experience?

Table 20: Codebook for S1Q23, S1Q24, S1Q27, and S1Q28, including the definition of each theme, examples, and the overall occurrence frequency.

Theme	Definition	Example Quotes	Frequency
Content	Course context, subjects & materials covered	*"nice to somewhat understand what's going on under the hood", "lecture content", " because I am interested in functional programming, that's all.", "they appeal to my intuition"*	45.7%
Course Design	Course components & strategies	*"nicely structured", "Project-based courses", "allowed me to work in groups", "we had a semester long project*	17.8%
Practicality	Involving experience & implementation	*"...create useful applications", "...connection made between theory and the real world.", "anything hands on", "...make a big project", " generally pretty obviously useful"*	16.9%
Instruction	Impacted by the instructor either by their teaching strategies or by the learning environment they create	*"the instruction was fantastic", "both the professors are best", "the prof did not make the course feel like a safe space.", "it was taught very well."*	11.3%
Belonging	Feeling of affinity & comfort in the course & the field	*"it reaffirmed my decision that this is the field that I want to be in", " felt like I was capable and like I belonged there."*	4.4%
Future	Impact on possibilities in the future	*"it is helpful for the future", "very applicable to many jobs", "...future work"*	3.8%

The percentages report the frequency of each theme among themes appearing in all valid responses across all questions (S1Q23, S1Q24, S1Q27, and S1Q28).

Finally, our survey reports that prior programming experience includes prevalently courses at the college level and independent learning. The distribution of answers for this question was noticeably uniform across demographics.

5.2.6 Threats to Validity — Quantitative Results. Since the link to the survey for students was shared on social media, we cannot be sure exactly who filled it out.

The surveys had language not consistent across regions and languages. Therefore, some questions could be interpreted differently. For example, the term "racialized" may have impacted our results as discussed in 7.

5.3 Student Survey: Qualitative Analysis

To answer **RQ2** on outcomes of curriculum efforts from a student perspective, and learn from student suggestions towards answering **RQ3** on improving curriculum, we performed qualitative analysis on open-ended questions in our student surveys.

5.3.1 Methodology. We performed thematic analysis on two categories of open-ended questions in our student survey: Student Experience Questions, and Suggestions. Student Experience Questions included six questions: S1Q23 - S1Q28. These questions ask about the experience of students in CS courses. Three of these questions, S1Q23, S1Q25, and S1Q28 are associated with positive experiences. They ask about courses the students enjoy the most, perform the best in, and courses the students particularly find engaging and the reasons. Three other questions, S1Q24, S1Q26, and S1Q27 are associated with negative experiences. They ask about courses the students enjoy the least, perform the worst in, and courses the students particularly find frustrating and the reasons. For each of these questions, we separated the analysis of the indicated course names from the analysis of the reasons discussed for the choices made. We performed a thematic analysis of the reasons provided and classified the course names into course categories.

We began inductive coding on S1Q23 in a bottom-up manner. This open-ended question asks students about the computer science courses they enjoy taking the most, and the reason for their enjoyment. You can find the question in Table 32. Two coders open-coded an initial sample of 76 student responses for this question (46 answers after removing blank and inapplicable entries such as “N/A”). The coders converged upon an initial codebook after discussion. The codebook includes six themes for the reason (for enjoyment or being frustrated) and seven categories of courses. The themes, their definitions and sample quotes are depicted in Table 20. The course categories are Core, Systems, Introductory, Theory, HCI, Application Oriented, and Data.

For question S1Q24, due to the similar structure of the question, the two coders coded independently while examining the

Table 21: Course Categories for S1Q23, S1Q24, S1Q27, and S1Q28, including the definition of each theme, examples, and the overall occurrence frequency.

Course Categories	Example Courses Included	Frequency
Core	Data Structures, Algorithms, Advanced Programming, Artificial Intelligence	27.6%
Theory	Advanced Algorithms, Complexity, Machine Learning	15.%
Introductory	Basic Programming, Introduction to Computing	15.9%
Systems	Operating Systems, Networks, Systems Security, Mobile Security	18%
Application-Oriented	Gaming, Computer Vision, Robotics, Bio-Computing, Web Development	3.3%
Data	Database Systems, Data Mining, Big Data	7%
HCI	Human-Centered Computing, Ethics of Computing, Societal Impacts of Computing	9%
None	When the student explicitly indicated they have not had such an experience with any courses.	5.6%

The percentages report the frequency of each course category among course categories in all valid responses across all questions (S1Q23-S1Q28). We have listed all of the course numbers we collected in responses and their mapping to course categories in Appendix C.

re-usability of the themes from the previous question and the possibility of adding new themes. After this process and discussions, based on the student responses, and the purpose of meaningful analysis, the coders decided to reuse the course categories and decided on a combination of reused themes from the previous question and a new added ones. The two coders decided similarly for S1Q25 and S1Q26 on course categories, and S1Q27 and S1Q28 for the course categories and the reason themes.

Finally, the two coders independently coded the question S1Q23 for all set of 613 responses and calculated an Inter-Rater Reliability Cohen's Kappa score of κ = 0.89 across these six themes and κ = 0.94 across seven course categories, indicating near perfect agreement. The coders then split the remaining responses for Q24-Q28 for themes and courses. For course categories, some participants used course codes in their responses (e.g., COMP200). The entries containing course numbers were not categorized by the coders. Instead, after initial coding of all entries performed by coders, for entries including course numbers, we determined the categories for course numbers by checking the referred university websites and agreed on course categories based on the syllabus and programs to automate classifying those items in the course categories. We have listed example courses in each course category in Table 21, and compiled all the courses referred to by course number and the category decided for them in Appendix C.

For question S1Q37, the coders repeated the inductive coding procedure in a bottom-up manner. This open-ended question asks for student suggestions on things that could be done to improve diversity, equity, and inclusion in the CS curriculum. You can find the question in 32. Two coders open-coded an initial sample of 76 student responses for this question (after removing blank and inapplicable entries such as "N/A"). The coders conferred and converged upon an initial codebook after discussion. The codebook includes three sentiment indicators for responder attitudes towards EDI initiatives, four themes for curriculum-related suggestions, and four themes for non-curriculum-related suggestions. The sample set of responses was then independently re-coded using this codebook and the coders calculated an Inter-Rater Reliability Cohen's Kappa score of κ = 0.49 across these eight themes and three attitudes. Codes from one of the coders were used to continue with for all answers. The themes for S1Q37, their definition, example quotes, and frequencies are listed in table 22.

In sections 5.3.2, 5.3.3, and 5.3.4, we present our qualitative analysis results reporting on student experiences, course categories we observe in their answers, and the themes and attitudes in suggestions to improve EDI in undergraduate CS curriculum. Please find the details of our analysis at: *https://osf.io/kdeyj?view_only=48eefdeb0c9d4538a51a251a48b539c9*, including the calculation of Cohen's kappa values for agreement, and Fihser's Exact significance test for all themes, course categories, suggestions, and attitudes for groups stratified by gender, racialization, and accessibility needs.

5.3.2 Results: Student Experiences. Course contents was the most frequently mentioned reason for the enjoyment of a course and engagement with it (56.4% of all valid responses in S1Q23 and 48% of all valid responses in S1Q28). Practicality followed as the second most frequent reason in response themes of both questions (22.9% of all valid responses in S1Q23 and 35.5% of all valid responses in S1Q28). Course design (21%), instruction (15.5%), belonging (5.4%), and future (3.8%) were the following themes for S1Q23 respectively. This order was slightly different for S1Q28, with Future as the third most frequent theme (11.5%), and course design (7.8%), instruction (5.6%), and belonging (2.5%), following as the other themes appearing respectively.

We did not observe significant differences among populations across gender and race in the frequency of reasons mentioned for enjoyment of a course. We observed significant differences between participants with and without accessibility needs in the frequency of mentioning course design as a reason for enjoyment of a course (S1Q23). Students with accessibility needs reported course design more frequently (29.8%) than participants without accessibility needs (17.9%) as a reason to enjoy a course in S1Q23.

Course contents was also the most frequently mentioned reason for not enjoying a course and frustration with it (S1Q24, SQ27). Practicality was not mentioned as frequently in reasons for frustration with a course, but course design and instructions followed contents in frequency. We did not observe differences between participants across races and accessibility needs in the frequency of reasons mentioned for not enjoying and being frustrated with a course. We observed significant differences between women and the other two groups combined, and men and the other two groups

Table 22: Codebook for S1Q37, including the definition of each theme, examples, and the overall occurrence frequency.

Theme	Definition	Example Quotes	Frequency
		Attitudes	
Support EDI	Explicit or implicit support for EDI goals through concrete suggestions to improve EDI	*"more female professors and TAs", "encourage more people of unique backgrounds to take computer science", "get more women interested in the field", "allocate seats for underrepresented groups"*	81.9%
Status Quo	Contentment with the current state and that no additional change should be made to support EDI	*"the current status quo is great for diversity, equity, and inclusion", "its pretty good right now in my opinion", "CS department is exceptionally diverse and inclusive in my opinion"*	11%
Against EDI	Explicit or implicit resistance against EDI goals through suggestions	*"removing grants opportunities specified for specific group/minorities etc.", "nothing should be done.", "I don't think diversity is an issue."*	5.7%
		Curriculum-Related Suggestions	
Topics & Content	Suggestions on topics to include or exclude in the CS curriculum	*"including more consideration to ethical and social topics in computing course", "There should be a course that teaches students about EDI", "A better underdivision ethics course, or potentially splitting the writing and ethics course into two"*	15.3%
Course Design	Suggestions on course design including methods and modes of instruction	*"more flexibility with grading and deadlines", "incorporate more group assignments and in-class activities", "promoting online work would help"*	14.8%
Program	Suggestions on program requirements, structure, flexibility, or revealing the hidden curriculum	*"don't assume people are good at math because they're in cs", "... should not be mandatory", "more courses", "more elective classes on..."*	13.8%
Course Access	Availability of the course for all students already in the program in some way	*"more seats for required CS major courses would help", "course's reserved seats for...", "able to have courses accessible to all students"*	2%
		Other Suggestions	
Cultural	Suggestions impacting social life, environment, or the culture of the program at an institution	*"social/group events for women in CS that are just casual are great", "continue to fund clubs like ... and encourage profs to attend", "start having events..."*	22.7%
Support	Suggestions involving instructor, teaching team, advisor, or equipment or facility support	*"advising!", "longer or more frequent office hours", "more female professors and TAs", "assistance to those with mobility issues"*	13.8%
Outreach	Suggestions on recruitment efforts and exposure and introduction of CS in earlier ages	*"introduce coding at earlier ages to everyone", "encourage more girls to do CS! This needs to start quite early on", "Efforts to 'demystify' the field"*	11.3%
Pathways	Suggestions on creating more equitable pathways for historically marginalized students	*"affirmative action", "allocate seats for underrepresented groups so there is more gender and racial diversity", "a more rounded application to CS and related disciplines that looks beyond good grades"*	6.4%

N=201. The percentages we report are the frequency of each code's usage among all the coded responses belonging to either of the categories of Attitudes (Support EDI, Status Quo, Against EDI) and Suggestions (Curriculum-Related Suggestions, Other Suggestions).

combined in the frequency of mentioning course design as a reason for feeling frustrated with a course (S1Q27). Women (31.4%) reported course design significantly more frequently as a reason for frustration with a course than men (20%) and self-identifying/non-binary participants in S1Q27 (27.8%). Men mentioned it significantly less frequently than the other two groups combined.

You can find the frequency of all themes appearing for reasons in S1Q23, S1Q24, S1Q27, and S1Q28 for all students and across gender, race, and accessibility needs in tables 23, 24, and 25.

We may infer from these results that contents are the main reason for student enjoyment in a course, but attention to design, possible future use cases, instruction, practicality, and sense of belonging can help improve it. Course design is particularly important for the enjoyment of students with accessibility needs.

Table 23: Themes appearing in the answers for Questions S1Q23, S1Q24, S1Q27, and S1Q28 by gender.

	All				**Women**				**Men**				**SI/NB**			
	Q23	Q24	Q27	Q28	Q23	Q24	Q27	Q28	Q23	Q24	Q27	Q28	Q23	Q24	Q27	Q28
Content	56.4%	54.0%	40.7%	48.0%	54.7%	51.1%	39.4%	46.7%	58.9%	49.4%	37.2%	46.7%	50.0%	44.4%	44.4%	50.0%
Practicality	22.9%	10.4%	4.3%	35.5%	26.3%	8.8%	4.4%	33.6%	22.8%	10.6%	3.9%	35.0%	11.1%	5.6%	0.0%	44.4%
Course Design	21.0%	23.4%	25.8%	7.8%	22.6%	24.8%	**31.4%***	8.8%	18.9%	19.4%	**20.0%***	7.2%	27.8%	38.9%	27.8%	11.1%
Instruction	15.5%	14.5%	13.8%	5.6%	19.7%	11.7%	14.6%	7.3%	13.3%	13.3%	10.0%	5.0%	16.7%	22.2%	16.7%	0.0%
Belonging	5.4%	5.8%	5.4%	2.5%	7.3%	5.8%	3.6%	3.6%	3.3%	3.3%	6.1%	1.7%	16.7%	11.1%	5.6%	0.0%
Future	3.8%	0.9%	0.3%	11.5%	4.4%	0.7%	0.0%	13.9%	3.9%	1.1%	0.6%	9.4%	0.0%	0.0%	0.0%	16.7%

N= 367, 346, 349, and 358 answered questions S1Q23, S1Q24, S1Q27, and S1Q28 respectively, out of 613 respondents. The percentages in the **All** column highlight the percentage of valid answers in each question that mention each theme. The percentages in **Women**, **Men**, and **SI/NB** columns indicate the percentage within **all** respondents in each group in our survey. The highlighted row indicates the theme where we observed significant differences among groups. The * indicates a significant difference between that group and the other two groups combined.

Table 24: Themes appearing in the answers for Questions S1Q23, S1Q24, S1Q27, and S1Q28 by Racialization.

	All				**Non-Racialized**				**Racialized**			
	Q23	Q24	Q27	Q28	Q23	Q24	Q27	Q28	Q23	Q24	Q27	Q28
Content	56.4%	54.0%	40.7%	48.0%	59.4%	53.5%	38.2%	45.9%	56.7%	47.8%	41.0%	47.8%
Practicality	22.9%	10.4%	4.3%	35.5%	21.2%	7.6%	2.9%	32.4%	26.9%	12.7%	6.0%	37.3%
Course Design	21.0%	23.4%	25.8%	7.8%	20.0%	24.1%	25.3%	8.2%	23.9%	23.9%	26.1%	9.0%
Instruction	15.5%	14.5%	13.8%	5.6%	17.1%	13.5%	14.7%	6.5%	15.7%	14.9%	11.9%	5.2%
Belonging	5.4%	5.8%	5.4%	2.5%	7.6%	4.1%	5.9%	2.9%	3.7%	6.0%	4.5%	2.2%
Future	3.8%	0.9%	0.3%	11.5%	3.5%	1.2%	0.0%	12.9%	5.2%	0.7%	0.7%	9.7%

N= 367, 346, 349, and 358 answered questions S1Q23, S1Q24, S1Q27, and S1Q28 respectively, out of 613 respondents. The percentages in the **All** column highlight the percentage of valid answers in each question that mention each theme. The percentages in **Non-Racialized** and **Racialized** columns indicate the percentage within **all** respondents in each group in our survey. The highlighted row indicates the theme where we observed significant differences among groups. The * indicates a significant difference between the two groups.

Table 25: Themes appearing in the answers for Questions S1Q23, S1Q24, S1Q27, and S1Q28 by Accessibility Needs.

	All				**No Accessibility Needs**				**With Accessibility Needs**			
	Q23	Q24	Q27	Q28	Q23	Q24	Q27	Q28	Q23	Q24	Q27	Q28
Content	56.4%	54.0%	40.7%	48.0%	57.3%	48.5%	38.3%	45.6%	57.9%	49.1%	36.8%	49.1%
Practicality	22.9%	10.4%	4.3%	35.5%	22.3%	8.4%	3.3%	35.8%	26.3%	12.3%	7.0%	31.6%
Course Design	21.0%	23.4%	25.8%	7.8%	**17.9%***	20.8%	24.8%	8.0%	**29.8%***	28.1%	28.1%	8.8%
Instruction	15.5%	14.5%	13.8%	5.6%	14.6%	13.5%	10.9%	5.5%	21.1%	14.0%	15.8%	5.3%
Belonging	5.4%	5.8%	5.4%	2.5%	5.5%	4.0%	4.4%	1.8%	7.0%	8.8%	8.8%	5.3%
Future	3.8%	0.9%	0.3%	11.5%	4.0%	1.1%	0.4%	10.6%	3.5%	0.0%	0.0%	14.0%

N= 367, 346, 349, and 358 answered questions S1Q23, S1Q24, S1Q27, and S1Q28 respectively, out of 613 respondents. The percentages in the **All** column highlight the percentage of valid answers in each question that mention each theme. The percentages in **No Accessibility Need** and **With Accessibility Needs** columns indicate the percentage within **all** respondents in each group in our survey. The highlighted row indicates the theme where we observed significant differences among groups. The * indicates a significant difference between the two groups.

5.3.3 Results: Course Categories. Core courses were the most frequently mentioned course category for the enjoyment of a course (39.8% of all valid responses in S1Q23 and 29.2% of all valid responses in S1Q28) as well as the courses participants performed the best in (43.2% of all valid responses in S1Q25). An interesting observation is that the core courses were also the most frequently mentioned course category for the courses students enjoyed the least (25.1% of all valid responses in S1Q24 and 21.9% of all valid responses in S1Q27), and the course categories the students performed the worst in (33% of all valid responses in S1Q26). The high frequency of core courses in the responses might be due to fact that these courses are required courses and students in different tracks and specializations all take these courses at some point in their program. You can find the frequency of course categories appearing in S1Q23-S1Q28 for all participant responses and across gender, race, and accessibility needs in tables 26, 27, and 28.

We observed significant differences between participant populations across gender, racialization, and accessibility needs in the frequency of course categories in their responses to the positive questions (S1Q23, S1Q25, and S1Q28). Introductory courses appeared with significantly higher frequency in responses to S1Q28 of self-identifying/non-binary participants (27.8%) compared to men (8.9%) and women (8.8%). For positive questions (S1Q23, S1Q25, and S1Q28) across groups stratified by accessibility needs, we observed significant differences in the frequency of mentioning theory courses. Participants with accessibility needs show significantly higher frequency (21.1%) of mentioning theory courses in response to S1Q25, the courses they performed best in, compared to the frequency of these courses mentioned by students who identified without accessibility needs (10.2%).

We observed significant differences in the frequency of mentioning theory courses between participants who identified as racialized and those who did not identify as racialized. Racialized participants indicated theory less frequently (19.4%) in their responses to S1Q23, the courses they most enjoyed, compared to non-racialized participants (30%). Racialized participants also mentioned theory courses less frequently (9%) in S1Q25, courses they performed best in, compared to non-racialized (17.1%) participants. Another significant difference between participant populations across racialization is in S1Q25 in the frequency of mentioning HCI courses. Racialized participants mentioned HCI courses significantly more frequently (5.2%) compared to non-racialized participants (0.6%) in response to S1Q25, the courses in which they performed the best. We also observed significant differences between participant populations across racialization in the frequency of mentioning core courses in response to S1Q28. Participants who identified as racialized mentioned core courses more frequently (36.6%) in their responses to S1Q28, the courses they particularly enjoyed compared to those who did not identify as racialized (25.3%). All of these results are subject to threats to validity we have discussed in section 7.

In negative questions (S1Q24, S1Q26, and S1Q27) we observed significant differences among genders. Application-oriented courses appeared with significantly higher frequency in responses to S1Q24 of self-identifying/non-binary participants (16.7%) compared to men (2.8%) and women (5.8%). In S1Q27, the courses the participants felt frustrated with, we observed significant differences among groups stratified by gender in introductory courses. Introductory courses appeared with significantly higher frequency in responses of women (19.7%) compare to men (12.2%) and self-identifying/non-binary participants (5.6%). We observed significant differences among genders in the frequency of answering S1Q27 with None, with lower percentages among women (10.2%) and higher percentages from men (21.1%), when compared to the other two groups combined.

We also observed significant differences between participants with and without accessibility needs in their responses to the negative questions (S1Q24, S1Q26, and S1Q27). Participants with accessibility needs reported application-oriented courses significantly more frequently (15.8%) in their responses compared to participants without accessibility needs (2.2%) in response to S1Q24, the courses they least enjoyed. The participants with accessibility needs also reported data-related courses significantly more frequently (12.3%) in their responses to S1Q27, the courses they felt frustrated with, compared to participants without accessibility needs (3.6%). These findings suggest that some of the more applied courses may present more hidden barriers for participants with accessibility needs, and explain why these participants found more discouraging elements in the curriculum than other participants (S1Q30).

We did not observe significant differences between racialized and non-racialized respondents in response to the negative questions (S1Q24, S1Q26, and S1Q27).

5.3.4 Results: Suggestions. While the question asked for specific suggestions on CS curriculum, the considerable number of suggestions included non-curriculum suggestions including exposure in early ages, outreach, admissions and diverse entry pathways to computing, as well as social activities and support groups to change the culture and help the student's feeling of belonging.

We did not observe significant differences among genders or racialization in the categories of solutions offered. We observed significant differences between participants with and without accessibility needs in the frequency of their responses in support and course design categories. Participants with accessibility needs suggested solutions within categories of course design (22.8%), and support (15.8%) significantly more frequently than participants without accessibility needs (5.5% and 6.6% for course design and support respectively).

While general support for EDI was high in our survey, we observed significant differences among genders in their attitudes supporting the status quo, or categorizing as anti-EDI. The support for status quo was significantly higher in responses by men (11.1% comments from all men participants of the survey), in comparison with the other two groups, women (2.2% of comments from all women participants), and self-identifying/non-binary (0% of all SI/NB participants). We identified anti-EDI attitudes significantly less frequently (0%) in responses by women compared to responses of 4.4% of men participants, and 5.6% of responses from self-identifying/non-binary participants, and significant difference among all categories in their comments in support for EDI, with comments from 55.5% of women participants, 35.6% men participants and, 72.2% of self-identifying/non-binary participants.

5.3.5 Threats to Validity — Qualitative Results. Some participants answered the questions only using course numbers and for some courses it was not possible to find what course or which university. Therefore, the analysis misses the analysis of some of entries.

Some participants seemed to be in their first year of the university. This reduces the chance of exposure to courses in later years (and potentially new interests) when answering the questions in this survey, increasing the possibility of listing the introductory (or core) courses in the answers.

Our survey terminology, specifically the term "racialized", may have been interpreted inconsistently. This might have impacted the resluts observed in tabels 24, 27, and 30. We further discuss this with details and exampls in section 7.

5.4 Student Interviews

Our team collaboratively designed the questions for student interviews, listed in Table 34. We designed the questions to be open-ended and provide us with an opportunity to further explore the participant answers towards answering **RQ2** and **RQ3**, the outcomes

Table 26: Course Categories for Questions S1Q23 - S1Q28 by Gender

	All			**Women**			**Men**			**SI/NB**		
Positive Questions	Q23	Q25	Q28	Q23	Q25	Q28	Q23	Q25	Q28	Q23	Q25	Q28
Core	39.8%	43.2%	29.2%	38.0%	38.0%	29.9%	42.2%	42.8%	30.6%	38.9%	27.8%	11.1%
Theory	24.8%	13.9%	11.4%	19.0%	13.9%	13.1%	27.8%	11.7%	11.1%	33.3%	16.7%	11.1%
Introductory	23.2%	28.4%	9.2%	23.4%	26.3%	8.8%	21.1%	25.0%	8.9%	27.8%	44.4%	**27.8%***
Systems	22.9%	16.5%	18.7%	19.7%	13.9%	14.6%	23.9%	18.3%	21.7%	33.3%	11.1%	27.8%
Data	10.6%	13.9%	12.8%	10.9%	14.6%	16.8%	11.1%	13.3%	11.1%	5.6%	16.7%	5.6%
Application-Oriented	13.1%	6.2%	18.9%	16.1%	8.0%	19.7%	10.0%	4.4%	15.0%	5.6%	5.6%	33.3%
HCI	3.5%	2.3%	5.3%	5.1%	2.2%	6.6%	2.2%	2.2%	5.0%	5.6%	5.6%	5.6%
None	0.8%	0.0%	11.7%	0.7%	0.0%	8.8%	1.1%	0.6%	8.3%	0.0%	0.0%	0.0%
Negative Questions	Q24	Q26	Q27	Q24	Q26	Q27	Q24	Q26	Q27	Q24	Q26	Q27
Core	25.1%	33.0%	21.9%	24.1%	26.3%	19.7%	22.8%	31.1%	21.7%	33.3%	38.9%	11.1%
Systems	22.0%	26.8%	19.1%	21.2%	28.5%	21.2%	20.6%	21.7%	17.8%	22.2%	27.8%	22.2%
Introductory	21.4%	12.8%	16.0%	21.2%	12.4%	**19.7%***	18.9%	11.1%	12.2%	22.2%	16.7%	5.6%
Theory	14.7%	14.6%	14.5%	11.7%	13.9%	14.6%	15.0%	13.3%	13.3%	11.1%	11.1%	11.1%
Data	10.4%	8.9%	6.0%	11.7%	8.8%	5.1%	7.8%	6.1%	3.9%	11.1%	11.1%	16.7%
HCI	5.8%	2.1%	3.7%	4.4%	2.9%	4.4%	6.7%	1.7%	2.8%	0.0%	0.0%	5.6%
Application-Oriented	5.8%	2.4%	2.0%	5.8%	3.6%	2.2%	2.8%	1.7%	1.1%	**16.7%***	0.0%	0.0%
None	4.6%	5.7%	16.5%	2.9%	5.1%	**10.2%***	3.9%	5.0%	**21.1%***	5.6%	5.6%	5.6%

N= 367, 346, 352, 336, 349, and 358 answered questions S1Q23- S1Q28 respectively, out of 613 respondents. The percentages in the **All** column highlight the percentage of valid answers in each question that mention each theme. The percentages in **Women**, **Men**, and **SI/NB** columns indicate the percentage within **all** respondents in each group in our survey. The highlighted row indicates the course category where we observed significant differences among groups. The * indicates a significant difference between that group and the other two groups combined.

Table 27: Course Categories for Questions S1Q23 - S1Q28 by Racialization

	All			**Non-Racialized**			**Racialized**		
Positive Questions	Q23	Q25	Q28	Q23	Q25	Q28	Q23	Q25	Q28
Core	39.8%	43.2%	29.2%	37.6%	37.1%	**25.3%***	43.3%	44.0%	**36.6%***
Theory	24.8%	13.9%	11.4%	**30.0%***	**17.1%***	11.2%	**19.4%***	**9.0%***	12.7%
Introductory	23.2%	28.4%	9.2%	23.5%	26.5%	11.2%	18.7%	25.4%	6.7%
Systems	22.9%	16.5%	18.7%	20.6%	15.3%	17.1%	26.9%	17.9%	20.9%
Application-Oriented	13.1%	6.2%	18.9%	11.8%	7.1%	19.4%	14.9%	6.0%	14.2%
Data	10.6%	13.9%	12.8%	10.0%	15.3%	11.8%	11.9%	14.2%	14.2%
HCI	3.5%	2.3%	5.3%	3.5%	**0.6%***	4.7%	3.7%	**5.2%***	8.2%
None	0.8%	0.0%	11.7%	1.2%	0.0%	9.4%	0.7%	0.0%	8.2%
Negative Questions	Q24	Q26	Q27	Q25	Q26	Q27	Q24	Q26	Q27
Core	25.1%	33.0%	21.9%	25.9%	27.1%	19.4%	25.4%	34.3%	25.4%
Systems	22.0%	26.8%	19.1%	20.6%	22.9%	17.6%	22.4%	27.6%	20.1%
Introductory	21.4%	12.8%	16.0%	20.6%	10.6%	17.6%	21.6%	14.2%	13.4%
Theory	14.7%	14.6%	14.5%	12.9%	11.2%	12.9%	14.2%	17.9%	17.9%
Data	10.4%	8.9%	6.0%	11.2%	9.4%	7.1%	9.7%	6.0%	4.5%
HCI	5.8%	2.1%	3.7%	5.9%	2.9%	4.7%	3.7%	1.5%	3.0%
Application-Oriented	5.8%	2.4%	2.0%	4.7%	2.4%	1.8%	5.2%	2.2%	1.5%
None	4.6%	5.7%	16.5%	2.9%	5.9%	14.1%	3.0%	2.2%	11.2%

N= 367, 346, 352, 336, 349, and 358 answered questions S1Q23- S1Q28 respectively, out of 613 respondents. The percentages in the **All** column highlight the percentage of valid answers in each question that mention each theme. The percentages in **Non-Racialized** and **Racialized** columns indicate the percentage within **all** respondents in each group in our survey. The highlighted row indicates the course category where we observed significant differences among groups. The * indicates a significant difference between the two groups.

of EDI-centered curriculum design efforts from the perspectives of students, and suggestions for computer sciences undergraduate programs in adopting EDI-centered curricula.

Table 28: Course Categories for Questions S1Q23 - S1Q28 by Accessibility Needs

	All			No Accessibility Needs			With Accessibility Needs		
Positive Questions	Q23	Q25	Q28	Q23	Q25	Q28	Q23	Q25	Q28
Core	39.8%	43.2%	29.2%	39.1%	40.9%	29.2%	40.4%	33.3%	28.1%
Theory	24.8%	13.9%	11.4%	23.4%	**10.2%***	9.9%	26.3%	**21.1%***	17.5%
Introductory	23.2%	28.4%	9.2%	21.9%	25.9%	10.2%	21.1%	29.8%	3.5%
Systems	22.9%	16.5%	18.7%	24.1%	15.3%	21.2%	21.1%	21.1%	12.3%
Application-Oriented	13.1%	6.2%	18.9%	10.2%	5.5%	17.5%	19.3%	5.3%	17.5%
Data	10.6%	13.9%	12.8%	10.6%	15.0%	12.8%	10.5%	10.5%	12.3%
HCI	3.5%	2.3%	5.3%	3.3%	2.2%	4.7%	3.5%	1.8%	10.5%
None	0.8%	0.0%	11.7%	1.5%	0.4%	9.5%	0.0%	0.0%	7.0%
Negative Questions	Q24	Q26	Q27	Q24	Q26	Q27	Q24	Q26	Q27
Core	25.1%	33.0%	33.0%	23.0%	27.4%	20.4%	29.8%	38.6%	15.8%
Systems	22.0%	26.8%	26.8%	20.4%	24.5%	17.9%	21.1%	24.6%	22.8%
Introductory	21.4%	12.8%	12.8%	20.1%	10.9%	15.7%	19.3%	14.0%	14.0%
Theory	14.7%	14.6%	14.6%	12.4%	13.5%	15.3%	17.5%	14.0%	7.0%
Data	10.4%	8.9%	8.9%	9.5%	6.9%	**3.6%***	10.5%	10.5%	**12.3%***
HCI	5.8%	2.1%	2.1%	5.1%	1.5%	2.9%	5.3%	5.3%	5.3%
Application-Oriented	5.8%	2.4%	2.4%	**2.2%***	2.2%	1.1%	**15.8%***	3.5%	3.5%
None	4.6%	5.7%	5.7%	3.3%	5.8%	17.2%	1.8%	0.0%	8.8%

N= 367, 346, 352, 336, 349, and 358 answered questions S1Q23- S1Q28 respectively, out of 613 respondents. The percentages in the **All** column highlight the percentage of valid answers in each question that mention each theme. The percentages in **No Accessibility Need** and **With Accessibility Needs** columns indicate the percentage within **all** respondents in each group in our survey. The highlighted row indicates the course category where we observed significant differences among groups. The * indicates a significant difference between the two groups.

Table 29: Suggestions and Attitudes in S1Q37 by Gender

Other Suggestions	All	Women	Men	SI/NB
Outreach	11.4%	5.8%	6.7%	11.1%
Pathways	6.5%	3.6%	3.3%	11.1%
Support	13.9%	10.2%	6.1%	16.7%
Cultural	22.9%	18.2%	9.4%	11.1%
Curriculum Suggestions	All	Women	Men	SI/NB
Program	13.9%	10.9%	6.1%	5.6%
Course Access	2.0%	1.5%	1.1%	0.0%
Course Design	14.9%	12.4%	6.1%	11.1%
Topics and Content	15.4%	10.9%	6.1%	27.8%
Attitude	All	Women	Men	SI/NB
Status Quo	11.9%	**2.2%***	**11.1%***	0.0%
Anti-EDI	5.5%	**0.0%***	**4.4%***	5.6%
Pro-EDI	78.6%	**55.5%***	**35.6%***	**72.2%***

N= 201. The percentages in the **All** column highlight the percentage of valid answers in each question that mention each theme. The percentages in **Women**, **Men**, and **SI/NB** columns indicate the percentage within **all** respondents in each group in our survey. The highlighted row indicates the theme where we observed significant differences among groups. The * indicates a significant difference between that group and the other two groups combined.

We recruited students for the interviews based on whether or not they expressed interest in a follow-up in the survey. We recruited 12 students, including three men, six women, two non-binary gender identities, and one person who preferred not to self-identify. Participants were from different racial and ethnic backgrounds from Australia, Canada, and the United States. Eleven interviewees believed in the need for equity, diversity, and inclusion initiatives in computer science, while one interviewee believed in the effectiveness of the status quo.

5.4.1 Results: Observations and Experiences. The majority of interview participants, across all gender identities, pointed to a lack of gender diversity in computer science. They reported as concerning issues such as the lack of women, the lack of acceptance of gender diversity, and the different standards in the professional treatment

Table 30: Suggestions and Attitudes in S1Q37 by Racialization

Other Suggestions	All	Non-Racialized	Racialized
Outreach	11.4%	7.6%	6.7%
Pathways	6.5%	4.7%	3.0%
Support	13.9%	7.6%	9.7%
Cultural	22.9%	14.1%	13.4%
Curriculum Suggestions	All	Non-Racialized	Racialized
Program	13.9%	7.6%	9.7%
Course Access	2.0%	0.6%	1.5%
Course Design	14.9%	8.8%	10.4%
Topics and Content	15.4%	8.2%	11.9%
Attitude	All	Non-Racialized	Racialized
Status Quo	11.9%	7.1%	5.2%
Anti-EDI	5.5%	1.8%	4.5%
Pro-EDI	78.6%	46.5%	51.5%

N= 201. The percentages in the **All** column highlight the percentage of valid answers in each question that mention each theme. The percentages in **Non-Racialized** and **Racialized** columns indicate the percentage of **all** respondents in each group in our survey. The highlighted row indicates the theme where we observed significant differences among groups. The * indicates a significant difference between the two groups.

Table 31: Suggestions and Attitudes in S1Q37 by Accessibility Needs

Other Suggestions	All	No Accessibility Need	With Accessibility Need
Outreach	11.4%	6.9%	3.5%
Pathways	6.5%	3.3%	1.8%
Support	13.9%	**6.6%***	**15.8%***
Cultural	22.9%	13.1%	10.5%
Curriculum Suggestions	All	No Accessibility Need	With Accessibility Need
Program	13.9%	6.6%	14.0%
Course Access	2.0%	1.1%	1.8%
Course Design	14.9%	**5.5%***	**22.8%***
Topics and Content	15.4%	8.4%	12.3%
Attitude	All	No Accessibility Need	With Accessibility Need
Status Quo	11.9%	7.7%	3.5%
Anti-EDI	5.5%	3.6%	0.0%
Pro-EDI	78.6%	**40.9%***	**66.7%***

N= 201. The percentages in the **All** column highlight the percentage of valid answers in each question that mention each theme. The percentages in **No Accessibility Need** and **With Accessibility Needs** columns indicate the percentage of **all** respondents in each group in our survey. The highlighted row indicates the theme where we observed significant differences among groups. The * indicates a significant difference between the two groups.

of women compared to men. Different treatment of women was mentioned by both men and women in different regions of the world. One example of the harmful behaviors mentioned was more people jumping in to help women when facing a problem, assuming their inability to solve it on their own. The observations also included additional requirements placed on women to prove their abilities before building trust, while the same standard does not apply to men in the field. All participants connected the highlighted observations with the low retention of women in the field.

The negative curriculum-related experiences included inappropriate examples causing unintentional harm (e.g., “The Lady and the Tiger”), the need for compulsory math contents not related to the degree, use of multiple programming languages in a course or sequence of courses, and lack of practicality or explanation of usage and context (e.g., Unified Modeling Language (UML) diagrams in software engineering) of the course materials.

Non-curriculum-related negative experiences included bullying incidents during COVID (which could be prevented through better platform settings), lack of sense of belonging, discouragement in asking questions, and a non-welcoming environment. An example of a non-welcoming environment was a first-year course instructor suggesting to students that their course may not be for everyone (due to its challenging nature), and that dropping out was an option. Such an announcement could result in students from diverse backgrounds or without prior programming experience feeling unwelcome.

Positive curriculum-related experiences included the practicality of the topics and offering of the courses and the inclusion of open-ended projects enabling personalizing and autonomy and applying skills in implementing projects of interest. Programming and systems courses were given as examples of courses with practicality. Data science and web application development courses were mentioned as examples of courses containing the favorably viewed open-ended projects. Positive experiences were also mentioned in the context of ethics and social implications courses due to the importance and relevance of the topics.

Non-curriculum-related good experiences (mentioned by women and non-binary participants) highlighted experiences with smart, well-respected, highly competent women whose presence in the program cultivated the feeling of belonging for people who otherwise did not see themselves represented. It also included examples of instructors sharing their experience (e.g., industry or startup experience). Students reported this helped them to feel more comfortable sharing their own experiences, put things in perspective, and make the program and courses seem more relatable.

5.4.2 Results: Suggestions to improve EDI. The most frequent recurring theme in the suggestions to improve EDI in CS curriculum was the inclusion of courses or topics on ethics and social implications of a computerized society. Participants also believed the importance of including this topic across a range of different courses should be explicitly emphasized in the curriculum. Multiple participants emphasized how vital this topic is in light of the fact that the training students receive in a CS program will empower them to make decisions that depend on critical thinking and ethical decision-making.

Another curriculum-related suggestion was spacing out CS courses. This will allow students more time to include variety in their program and when provided at the beginning of the program enables them to make an informed decision about their specialization. Altering the curriculum to allow people to branch out to their specialization earlier in the program was another suggestion for improving EDI in the curriculum.

At the course level, practicality and contextualization of course contents and introducing practical examples to otherwise abstract core courses were content-related suggestions. Participants also suggested addition of group projects, and smaller course components or flavors from different specializations in core courses earlier in the program. These together with use of paper-based exams or better design of online exams to prevent academic dishonesty and improve fairness were course-design level curriculum-related suggestions to improve EDI.

Non-curriculum-related suggestions included hiring more women and people from diverse backgrounds. Participants reasoned that providing more diverse representation would help in cultivating a sense of belonging, as well as increasing the likelihood of unintentional harmful course content being identified and removed. Other suggestions included providing communication guidelines by instructors for creating safe spaces in the classroom for women and minoritized populations, and outreach.

6 Educators: Surveys and Interviews

6.1 Educator Survey

To include the voices of educators who are at the forefront of implementing changes and observing their impact, we also designed

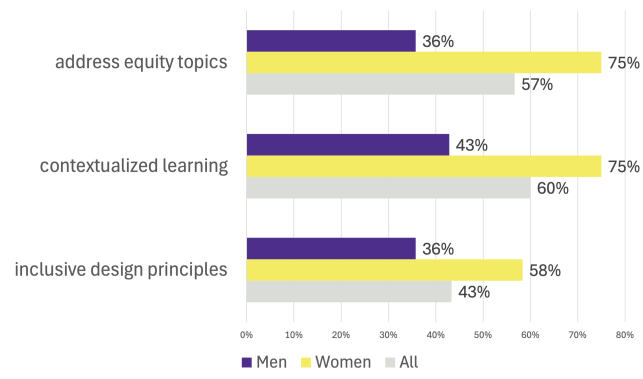

Figure 11: Percentage of educators, by gender, utilizing three types of EDI curriculum (addressing equity needs, contextualized learning, and inclusive design principles).

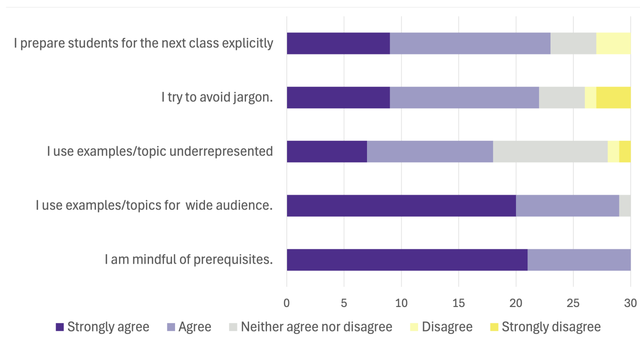

Figure 12: How educators use EDI-inspired practices.

an Educator Survey and follow-up interviews. We designed the survey questions for university educators teaching in a computer science program. We designed and reviewed the questions multiple times collaboratively, in a process similar to our student survey and interview design process. The survey included 20 questions and one for expressing interest in participating in an interview (a total of 21 questions). It included one 5-point Likert Scale, one matrix, four multiple choice, and 14 open-ended questions. Survey questions are listed in Table 33.

The educator survey received 30 responses and all responses are fully complete. The data is collected through advertisement during ITiCSE 2024 and also extended to our educators within our studied institutions and within our networks. The composition of the respondents can be summarized as follows:

- 57% of the educators have taught lower division courses. 50% upper division courses, 27% graduate courses, 17% service courses, and 10% other types of courses including Teacher Training.
- 70% of the educators identified as not racialized, 13% identified as racialized, 7% were not sure, and 10% preferred not to answer. 63% identified as White, 17% preferred not to answer. We do not announce other racial minorities due to

the possibility of identification due to the small number of participants in each category.

- 40% of respondents were women, 47% were men, and 13% preferred not to answer.

Our survey has several questions themed around types of EDI-inspired curriculum interventions educators have tried in their classes or programs. We see that 33% of educators responded that their institution has gone through significant changes in the undergraduate curriculum to increase diversity in the student population (i.e. changes in core courses, new paths into the degree, etc.) There were two main ways that educators reported making these significant changes. The first is through the addition of interdisciplinary programs like CS+X and Data Science. The second is through incorporating a modified CS1 course or the addition of a bridge program. Both of these themes were found in our literature review (Section 3), as well. We also asked educators to tell us about EDI-inspired curriculum and/or practices that they personally have tried out in their courses. Figure 11 shows what percentage of faculty, broken down by gender, have addressed equity topics, used contextualized learning, and used inclusive design principles. In all cases, women were more likely than men to use this type of curriculum. In Figure 12, we show how faculty responded to EDI inspired practices. We did not break this down by gender because the differences were not great. However, women reported higher agreement with avoiding jargon in courses compared to men, and men reported more agreement with preparing students for the next course compared to women. Lastly, we asked educators to share how they designed their courses and assessments to address the needs of a diverse set of students. The most popular response is to design relatable, real-world, or diverse examples, which we also found as a theme for EDI inspired interventions in our literature review. Other repeated ideas shared by educators in the survey include flexibility (which showed up as a theme from our student survey), multiple modalities (like lecture, textbook, recordings, etc.), group work or pair programming, and universal design for learning.

Our survey also had several questions to understand ideas educators have, as well as resources they need and challenges they face. When asked how to improve CS curriculum in general and for purposes of EDI, and the most popular emerging theme is related to admissions or pathways in the program. The second most popular theme is to incorporate EDI into the curriculum (in general, no specifics). Other ideas (in order of popularity) include more breadth courses for students to explore CS, more modern or industry-related curriculum, more ethics infused in the curriculum, better consistency/coordination across faculty/sections/years of the curriculum, reduce the number of required courses, and a more diverse faculty. The biggest challenge educators face in the area of EDI is faculty buy-in and skepticism, which was mentioned by nearly half of the respondents. Lastly, educators reported the lack of resources, mainly that faculty teaching loads are too high or the need for time/money. Many respondents also reported that they do not know how to fix the problem of the lack of diversity in CS and many mentioned they would appreciate having an EDI expert in their department to help guide the changes.

Lastly, 75% of the educators surveyed reported that their institution has tried to address inequities in computer science participation with students (at any point in time). The two most popular examples provided are outreach efforts (e.g. for “disadvantaged” high school students, women, minorities, etc.) or student-run groups (e.g. women in computer science, minority-focused CS student organizations, etc.). The two most popular examples provided were not curriculum-focused. Other examples mentioned (with much less frequency) include: admissions-specific changes, split CS1 courses or bridge programs, interdisciplinary programs like CS + X programs, EDI faculty committees, and providing more role models (such as diverse TA staff).

6.2 Educator Interviews

To further explore educators' perspectives towards answering **RQ2** and **RQ3**, we designed educator interview questions. The interview questions for educator interviews are listed in Table 35. We designed these questions to be used as a follow-up to educator surveys or independently. The recruitment process for educators included survey follow-up as well as a direct response to our email requests sent to our investigated universities. We only recruited interviewees through follow-up from our Educators survey. We recruited less than five interviewees, all white men from Canada and Europe. We do not state the exact number of participants to protect our participants against identifiablity.

All of our participants indicated their institution's awareness and efforts towards addressing EDI issues in their CS programs. These efforts were in different dimensions of diversity and inclusion, with a higher emphasis on gender diversity.

When asked about contextualized learning, all respondents provided examples of practical real-world examples in the content for their courses. However, while some participants invited their students to share about themselves, others indicated they refrain from this mostly because of practical issues pertaining to larger classes which may not allow the time and make the students uncomfortable when sharing about themselves.

All educators were observant of their students. Responses from our Canadian participants indicated numerous examples of observing and being aware of experiences of sexism by women in the field as well as racialized, and socio-economic experiences. Our European respondents were also observant of the experiences of women, and experiences because of social cliques and isolation of students disabling them from receiving peer support.

Our participants suggested the following solutions to improve EDI through CS Curriculum:

(1) Offering "pure" and "applied" versions of computer science programs to cater to a wider audience, making it interesting for a wider audience and creating additional pathways.
(2) Adding Undergraduate research opportunities. This suggestion was based on an observation about the higher interest of women in being involved in undergraduate research courses and co-curricular activities.
(3) Further flexibility in electives to foster interests in specialized directions.

Other recommendations included an emphasis on the importance of community for supporting EDI, with concrete suggestions such as integrating labs in course components and fostering a sense of community through shared activities in a shared safe space to learn and collaborate as a group.

7 Threats to Validity

Our analysis only considered three regions: Australia, Canada, Europe, and the US. Additionally, only ten renowned universities and a total of 9 equity-focused universities are considered per region, and these universities may not be representative of all universities within the region. There are variations between regions, and our criteria for selecting the universities (CS program ranking Times Higher Education) and establishing methods of reporting differences in their admission criteria and websites may not result in the generalizability of our results for the regions.

Our selected universities published their data in English and other languages, such as French. Our working group, however, included people who were fluent in all languages needed for analysis. Despite the resourcefulness, the differences in reporting were evident in institutions whose data was not available in English.

Since the link to the surveys for students was shared on social media, we can not guarantee that only the intended audiences responded. We also have higher response rates from some regions (Canada and Australia) than others (Europe). Finally, it is possible for the data collected to be influenced by voluntary response bias.

Our choice of terminology may have impacted the survey results. We specifically observe this for the term "racialized". To analyze this, we compared the answer to S1Q41, self-identification as a racialized person, with the answer to S1Q42, racial and/or ethnic identity. We found results that may have impacted our analysis. For example, 53.85% of 13 participants who identified in S1Q42 as West Asian, in S1Q41 identified as non-racialized, 30.77% identified as racialized, and 15.38% answered "Not Sure". 20.78% of 77 participants who identified as Chinese in S1Q42 identified in S1Q41 as non-racialized, 59.74% identified as racialized, and 18.18% answered "Not Sure". We observed this probable impact of terminology for participants who identified Latin American, Mixed/Bi-Racial, and South Asian as well. Nevertheless, the answer to S1Q41 could help in understanding students' experiences in self-association with racialized identity.

The educator survey was advertised to peers in ITiCSE Working Groups. The majority of our responses were collected from this recruitment method, including the recruitment of some authors of this work. While this includes our valid target populations with a diversity of opinions on the subject, it introduces selection bias by including educators working on the subject or otherwise active in an informed community.

8 Conclusion and Recommendations

To address the problem of underrepresentation in computer science, we have to provide a positive experience for historically marginalized students throughout their education and career. One of the important elements of student experience throughout their CS education is the CS curriculum. While the CS curriculum should be rigorous in order to ensure that the education received is worthwhile, it also needs to be exciting, inviting, inclusive, and relevant. This can contribute to fixing the leaky pipeline of retention and help raise interest in prospective students.

In this work, we explored addressing equity, diversity and inclusion in CS undergraduate curricula. To address our **RQ1**, looking for examples of EDI-centered curriculum design efforts in undergraduate computer science programs, we performed an extensive literature review and inductively classified interventions to improve EDI into six main categories:

(1) Levelling the playing field
(2) Integration of capstones, research, and industry projects
(3) Simplifying curricular complexity
(4) Creating interdisciplinary routes into computing
(5) Challenging stereotypes
(6) Cultural and social justice competency

We explored examples for each category in the literature and examined their adoption in forty-nine institutions across three regions around the world. We explored examples of such curriculum interventions through public data of the examined institutions and explored student experiences with the curriculum by including the voices of 613 students from these regions through surveys and interviews.

Through the study of public data from the chosen institutions and the feedback from students, we gathered information to answer **RQ2** — what are the outcomes of these EDI-centered curriculum design efforts — as well as suggestions to improve CS curriculum. Our study highlights some signs of frustration with course contents and design, with students of different demographics recounting that they have dealt with discouraging curriculum elements at some time or another. Our student survey results indicate differences in experiences within the program, impacting the sense of attainability of success across gender, race, and accessibility needs. The results also acknowledge the prevalence of cultural problems impacting historically marginalized students. Minoritized students (across all dimensions of diversity) were found to be less confident in their potential success as computer scientists. Other questions, inquiring about perceived acceptability among fellow students, show significant differences from respondents of different genders. We also found that underrepresented students reported with higher frequency finding discouraging elements in the curriculum, suggesting that more could be done to address the needs of these populations. Our results also highlighted student interest in topics and courses on bio-inspired applications, ethical computing, and the social implications of computing. Increasing the presence and availability of courses on these topics could contribute to increasing the appeal of computer science to a more diverse audience.

We complemented our study by examining institutional cultures and the state of the current efforts from the lens of CS educators. We also explored educator ideas to improve EDI in CS, and the resources needed to support the implementation of those ideas, using surveys and interviews. Educators pointed to increasing pathways into CS programs as a way to improve diversity. They also mentioned challenges on their path to promoting EDI, including securing other faculty buy-in and the lack of resources to dedicate to these efforts.

The combined results from our literature review, study of public data, student surveys and interviews, and educator surveys and interviews highlight the need for EDI considerations in the CS curriculum, and helped us formulate an answer to our **RQ3** on how undergraduate computer science programs can adopt EDI-centered curricula. In the remaining paragraphs, we present changes to the CS curricula which we believe may lead to better equity, diversity, and inclusion in computing science programs.

At the course level, contents have a major impact toward a variety of potential outcomes. These outcomes range from unintentional harm to students' sense of belonging, to increasing their interest through thoughtful, accessible, practical, and engaging content. Course design can also impact students. It can impair or support their feeling of fairness. It can also help foster an environment for enriching additional social and soft skills. Therefore:

(1) Be mindful while creating or adopting content. Examine and remove potentially harmful language or examples that may alienate students of different gender identities, racial backgrounds, and accessibility needs. This may include historically used terms, well-known but inappropriate examples, or gendered examples with assumed roles.
(2) Be aware that not all students in your courses share the same vocabulary. Avoid jargon and consider adding a summary of definitions when appropriate.
(3) Pay attention to the usage of pronouns and gender in your content and examples.
(4) Make interesting, practical, real-world connections for abstract content to make it interesting and comprehensible.
(5) Consider students with diverse accessibility needs in your course design. This includes presenting and encouraging different approaches to problem-solving, and being mindful of barriers included in hands-on course components.

At the program level, entry pathways, program requirements, required or elective subjects, and the availability of topics of interest and research or industrial experiences can engage or deter students with diverse backgrounds or interests. Therefore:

(1) Consider leveling the playing field by using appropriate language and technology choices in introductory computing science and programming courses.
(2) Introduce topics of interest to the curriculum. Examples include bio-related applications of computing in entry courses, or ethics and social implications of computing throughout the curriculum.
(3) Provide group projects, hands-on experiences, and practical industry, or research opportunities when possible.
(4) Re-evaluate and update degree requirements regularly to ensure an alignment between program goals and structure. For example, examine and upgrade the chain of once-relevant now-obsolete math or physics pre-requisites.
(5) Increase flexibility (e.g., electives, research experiences, exploratory topics), and create specialized paths (e.g., bio-inspired) to the program when possible.

The recommendations outlined above may produce positive effects beyond the targeted group of students, and facilitate a cultural shift to foster a more inclusive environment for all.

9 Future Work

As a first step in exploring the impacts of curriculum on equity, diversity, and inclusion in computing science curricula, our work provides valuable insights into the categories of interventions experimented with, ongoing efforts in committed institutions, and student and educator perspectives on the current status. In the future, we would like to collect more data and expand our current work in the following directions:

(1) We explored curriculum related factors relating to CS participation in Western post-secondary education. However, the assumptions which underlie our work—in particular those relating to cultural perceptions of CS careers—have been shown to not necessarily hold in different global contexts [117]. We would like investigate the specificity of other regions and consider the extent to which the barriers and assets we identified may apply, using them as a point of departure and comparison in identifying others.
(2) We found indicators of disparity of interest among different groups in certain topics, therefore we would like to explore the reason behind such interests or lack thereof, beyond industry influence or job prospects.
(3) Our results on discouraging elements of the curriculum, together with examples discussed by students, suggest the interdependence of culture and curriculum. We would like to examine the impact of cultural and support elements, such as role models and support systems, on raising interest in core and systems topics in the CS curriculum.
(4) Some of the discussed approaches such as the use of instructional programming languages are in some ways in contrast with the practicality of the contents, but improve EDI through levelling the playing field for students without prior chance of exposure. We would like to further examine student experiences through programs focusing on interest and practicality with those trying to level the playing field to further examine the impact and recommended use cases. This may help to create pathways tailored more precisely to each scenario.

Acknowledgments

Ouldooz Baghban Karimi's work on this project was supported by a Rajan Family Scholarship and a Google exploreCSR 2023 award. The authors also acknowledge the support from Dr. Bianca Brigidi, Center for Educational Excellence, Simon Fraser University.

References

[1] Andrea Aler Tubella, Marçal Mora-Cantallops, and Juan Carlos Nieves. 2024. How to teach responsible AI in Higher Education: challenges and opportunities. *Ethics and Information Technology* 26 (2024). Issue 3. https://doi.org/10.1007/s10676-023-09733-7
[2] Meghan Allen, Celina Berg, Jessica Q. Dawson, and Neil Leveridge. 2018. Insights from the Application of Universal Design Principles to Support English Language Learners. In *Proceedings of the 23rd Western Canadian Conference on Computing Education* (Victoria, BC, Canada) *(WCCCE '18)*. Association for Computing Machinery, New York, NY, USA, Article 10, 6 pages. https://doi.org/10.1145/3209635.3209646
[3] Christine Alvarado, Zachary Dodds, and Ran Libeskind-Hadas. 2012. Increasing women's participation in computing at Harvey Mudd College. , 55–64 pages. https://doi.org/10.1145/2381083.2381100
[4] Isis Artze-Vega, Flower Darby, Bryan Dewsbury, and Mays Imad. 2023. *The Norton Guide to Equity-Minded Teaching.* New York: WW Norton.
[5] Ouldooz Baghban Karimi, Giulia Toti, Mirela Gutica, Rebecca Robinson, Lisa Zhang, James Paterson, Peggy Lindner, and Michael O'Dea. 2023. Enhancing Diversity and Inclusion in Computer Science Undergraduate Programs: The Role of Admissions. In *Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education* (Turku, Finland) *(ITiCSE-WGR '23)*. Association for Computing Machinery, New York, NY, USA, 1–29. https://doi.org/10.1145/3623762.3633496

[6] Ouldooz Baghban Karimi, Giulia Toti, Mirela Gutica, Rebecca Robinson, Lisa Zhang, James H. Paterson, Peggy Lindner, and Michael O'Dea. 2023. Exploring Computing Science Programs' Admission Procedures with a Diversity and Inclusion Lens. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 2* (Turku, Finland) *(ITiCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 563–564. https://doi.org/10.1145/3587103.3594208

[7] 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

[8] 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. https://doi.org/10.1145/1539024.1508923

[9] Valerie Barr. 2018. Different denominators, different results: reanalyzing CS degrees by gender, race, and ethnicity. *ACM Inroads* 9, 3 (2018), 40–47. https://doi.org/10.1145/3239261

[10] Daphne Barretto, Julienne LaChance, Emanuelle Burton, and Soohyun Nam Liao. 2021. Exploring Why Underrepresented Students Are Less Likely to Study Machine Learning and Artificial Intelligence. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1* (Virtual Event, Germany) *(ITiCSE '21)*. Association for Computing Machinery, New York, NY, USA, 457–463. https://doi.org/10.1145/3430665.3456332

[11] Maria Cecilia Bastarrica and Jocelyn Simmonds. 2019. Gender Differences in Self and Peer Assessment in a Software Engineering Capstone Course. In *2019 IEEE/ACM 2nd International Workshop on Gender Equality in Software Engineering (GE)*. 29–32. https://doi.org/10.1109/GE.2019.00014

[12] Rebecca Bates, Jonathan Hardwick, Guarionex Salivia, and Lin Chase. 2022. A Project-Based Curriculum for Computer Science Situated to Serve Underrepresented Populations. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education - Volume 1* (Providence, RI, USA) *(SIGCSE 2022)*. Association for Computing Machinery, New York, NY, USA, 585–591. https://doi.org/10.1145/3478431.3499312

[13] Adam SZ Belloum, Spiros Koulouzis, Tomasz Wiktorski, and Andrea Manieri. 2019. Bridging the demand and the offer in data science. *Concurrency and Computation: Practice and Experience* 31, 17 (2019), e5200. https://doi.org/10.1002/cpe.5200

[14] Sylvia Beyer. 2014. Why are women underrepresented in Computer Science? Gender differences in stereotypes, self-efficacy, values, and interests and predictors of future CS course-taking and grades. *Computer Science Education* 24, 2-3 (2014), 153–192. https://doi.org/10.1080/08993408.2014.963363

[15] Charles C Bonwell and James A Eison. 1991. *Active learning: Creating excitement in the classroom. 1991 ASHE-ERIC higher education reports.* ERIC.

[16] Steven Bradley, Miranda C. Parker, Rukiye Altin, Lecia Barker, Sara Hooshangi, Thom Kunkeler, Ruth G. Lennon, Fiona McNeill, Julià Minguillón, Jack Parkinson, Svetlana Peltsverger, and Naaz Sibia. 2023. Modeling Women's Elective Choices in Computing. In *Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education* (Turku, Finland) *(ITiCSE-WGR '23)*. Association for Computing Machinery, New York, NY, USA, 196–226. https://doi.org/10.1145/3623762.3633497

[17] Rex P. Bringula, John Benedic R. Enriquez, Rodany A. Merida, Gil Francis S. Abinal, and Jose Mari S. Ramos. 2014. Attitude towards Capstone Project: Looking from gender and stage perspectives. In *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*. 1–7. https://doi.org/10.1109/FIE.2014.7044315

[18] Carla E. Brodley, Benjamin J. Hescott, Jessica Biron, Ali Ressing, Melissa Peiken, Sarah Maravetz, and Alan Mislove. 2022. Broadening Participation in Computing via Ubiquitous Combined Majors (CS+X). *SIGCSE 2022 - Proceedings of the 53rd ACM Technical Symposium on Computer Science Education* 1, 544–550. https://doi.org/10.1145/3478431.3499352

[19] Carla E. Brodley, Benjamin J. Hescott, Jessica Biron, Ali Ressing, Melissa Peiken, Sarah Maravetz, and Alan Mislove. 2022. Broadening Participation in Computing via Ubiquitous Combined Majors (CS+X). In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education - Volume 1* (Providence, RI, USA) *(SIGCSE 2022)*. Association for Computing Machinery, New York, NY, USA, 544–550. https://doi.org/10.1145/3478431.3499352

[20] Giang Bui, Naaz Sibia, Angela Zavaleta Bernuy, Michael Liut, and Andrew Petersen. 2023. Prior Programming Experience: A Persistent Performance Gap in CS1 and CS2. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1* (Toronto ON, Canada) *(SIGCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 889–895. https://doi.org/10.1145/3545945.3569752

[21] Sheryl Burgstahler. 2011. Universal Design. *ACM Transactions on Computing Education* 11 (10 2011), 1–17. Issue 3. https://doi.org/10.1145/2037276.2037283

[22] 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. https://doi.org/10.1093/iwc/iwv046

[23] Burçin Campbell. 2022. *Computing Bachelor's Degrees Awarded vs US College Age Population by Race/Ethnicity and Gender/Sex (2022)*. Retrieved July 2, 2024 from https://cra.org/crn/2024/02/computing-bachelors-degrees-awarded-vs-us-college-age-population-by-race-ethnicity-and-gender-sex-2022/

[24] Valerie Carr, Morris Jones, and Belle Wei. 2020. Interdisciplinary Computing: Applied Computing for Behavioral and Social Sciences. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20)*. Association for Computing Machinery, New York, NY, USA, 400–406. https://doi.org/10.1145/3328778.3366799

[25] Lori Carter. 2014. Interdisciplinary computing classes: worth the effort. In *Proceedings of the 45th ACM technical symposium on Computer science education (SIGCSE '14)*. Association for Computing Machinery, New York, NY, USA, 445–450. https://doi.org/10.1145/2538862.2538882

[26] Sapna Cheryan, Ella J Lombard, Lauren Hudson, Kengthsagn Louis, Victoria C Plaut, and Mary C Murphy. 2019. Double isolation: Identity expression threat predicts greater gender disparities in computer science. *Self and Identity* 19, 4 (2019), 412–434. https://doi.org/10.1080/15298868.2019.1609576

[27] Sapna Cheryan, Victoria C. Plaut, Caitlin Handron, and Lauren Hudson. 2013. The Stereotypical Computer Scientist: Gendered Media Representations as a Barrier to Inclusion for Women. *Sex Roles* 69, 1 (01 Jul 2013), 58–71. https://doi.org/10.1007/s11199-013-0296-x

[28] Joel Coffman, Adrian A. de Freitas, Justin M. Hill, and Troy Weingart. 2023. Visual vs. Textual Programming Languages in CS0.5, In Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1 (New York, NY, USA). *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, 32–38. https://doi.org/10.1145/3545945.3569722

[29] James P. Cohoon and Luther A. Tychonievich. 2011. Analysis of a CS1 approach for attracting diverse and inexperienced students to computing majors. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education* (Dallas, TX, USA) *(SIGCSE '11)*. Association for Computing Machinery, New York, NY, USA, 165–170. https://doi.org/10.1145/1953163.1953217

[30] Tammie Cumming, M David Miller, and Isana Leshchinskaya. 2023. DEI institutionalization: Measuring diversity, equity, and inclusion in postsecondary education. *Change: The Magazine of Higher Learning* 55, 1 (2023), 31–38. https://doi.org/10.1080/00091383.2023.2151802

[31] James Davis, Michael Lachney, Zoe Zatz, William Babbitt, and Ron Eglash. 2019. A cultural computing curriculum. *SIGCSE 2019 - Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 1171–1175. https://doi.org/10.1145/3287324.3287439

[32] Jessica Q. Dawson, Meghan Allen, Alice Campbell, and Anasazi Valair. 2018. Designing an Introductory Programming Course to Improve Non-Majors' Experiences. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (Baltimore, Maryland, USA) *(SIGCSE '18)*. Association for Computing Machinery, New York, NY, USA, 26–31. https://doi.org/10.1145/3159450.3159548

[33] Zachary Dodds, Malia Morgan, Lindsay Popowski, Henry Coxe, Caroline Coxe, Kewei Zhou, Eliot Bush, and Ran Libeskind-Hadas. 2021. A Biology-based CS1: Results and Reflections, Ten Years In. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education* (Virtual Event, USA) *(SIGCSE '21)*. Association for Computing Machinery, New York, NY, USA, 796–801. https://doi.org/10.1145/3408877.3432469

[34] Ron Eglash, Juan E. Gilbert, and Ellen Foster. 2013. Toward culturally responsive computing education. *Commun. ACM* 56, 7 (July 2013), 33–36. https://doi.org/10.1145/2483852.2483864

[35] Christopher Emdin, Edmund Adjapong, and Ian Levy. 2016. Hip-hop based interventions as pedagogy/therapy in STEM: A model from urban science education. *Journal for Multicultural Education* 10 (2016), 307–321. Issue 3. https://doi.org/10.1108/JME-03-2016-0023

[36] National Science Foundation. April 29, 2021. *Women, Minorities, and Persons with Disabilities in Science and Engineering, Table 5-1 :Bachelor's degrees awarded, by sex and field: 2008–18.* Retrieved July 2, 2024 from https://ncses.nsf.gov/pubs/nsf21321/table/5-1

[37] Paulo Freire. 2018. *Pedagogy of the Oppressed: 50th Anniversary Edition.* Bloomsbury Academic.

[38] Carol Frieze and Jeria L. Quesenberry. 2019. How computer science at CMU is attracting and retaining women. *Commun. ACM* 62, 2 (jan 2019), 23–26. https://doi.org/10.1145/3300226

[39] Daniela Gachago, Maha Bali, and Nicola Pallitt. 2024. Equity-Oriented Learning Design: An Entangled Future. *Postdigital Science and Education* 6 (3 2024), 173–193. Issue 1. https://doi.org/10.1007/s42438-023-00420-w

[40] 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 Trans. Comput. Educ.* 23, 3, Article 34 (jul 2023), 35 pages. https://doi.org/10.1145/3603535

[41] Rosalinda Garcia, Patricia Morreale, Gail Verdi, Heather Garcia, Geraldine Jimena Noa, Spencer P. 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 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 657, 22 pages. https://doi.org/10.1145/3613904.3642475

[42] Geneva Gay. 2018. *Culturally responsive teaching: Theory, research, and practice*. teachers college press.

[43] Michail N Giannakos, Ilias O Pappas, Letizia Jaccheri, and Demetrios G Sampson. 2017. Understanding student retention in computer science education: The role of environment, gains, barriers and usefulness. *Education and Information Technologies* 22 (2017), 2365–2382. https://doi.org/10.1007/s10639-016-9538-1

[44] Pawel Grabarczyk, Alma Freiesleben, Amanda Bastrup, and Claus Brabrand. 2022. Computing Educational Programmes with more Women are more about People & less about Things. *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE* 1, 172–178. https://doi.org/10.1145/3502718.3524784

[45] Mark Guzdial and August Evrard. 2024. Identifying the Computing Education Needs of Liberal Arts and Sciences Students (Discussion Paper). In *Proceedings of the 23rd Koli Calling International Conference on Computing Education Research* (Koli, Finland) *(Koli Calling '23)*. Association for Computing Machinery, New York, NY, USA, Article 19, 7 pages. https://doi.org/10.1145/3631802.3631805

[46] Mark Guzdial and August Evrard. 2024. Identifying the Computing Education Needs of Liberal Arts and Sciences Students (Discussion Paper). In *Proceedings of the 23rd Koli Calling International Conference on Computing Education Research* (Koli, Finland) *(Koli Calling '23)*. Association for Computing Machinery, New York, NY, USA, Article 19, 7 pages. https://doi.org/10.1145/3631802.3631805

[47] Shirin Haji Amin Shirazi, Mariam Salloum, Annika Speer, and Neftali Watkinson. 2024. An Experience Report: Integrating Oral Communication and Public Speaking Training in a CS Capstone Course. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1* (Portland, OR, USA) *(SIGCSE 2024)*. Association for Computing Machinery, New York, NY, USA, 450–455. https://doi.org/10.1145/3626252.3630776

[48] Andrew Hanson, Carlo Salerno, Matt Sigelman, Mels de Zeeuw, and Stephen Moret. 2024. *Talent Disrupted: Underemployment, College Graduates, and the Way Forward*. Technical Report. The Burning Glass Institute, Strada Institute. Retrieved Dec 5, 2024 from https://stradaeducation.org/wp-content/uploads/2024/02/Talent-Disrupted.pdf

[49] Michael Haungs, Christopher Clark, John Clements, and David Janzen. 2012. Improving first-year success and retention through interest-based CS0 courses. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education* (Raleigh, North Carolina, USA) *(SIGCSE '12)*. Association for Computing Machinery, New York, NY, USA, 589–594. https://doi.org/10.1145/2157136.2157307

[50] Reiko Hishiyama and Tehgfei Shao. 2022. Educational Effects of the Case Method in Teaching AI Ethics. In *World Conference on Information Systems and Technologies*. Springer, 226–236. https://doi.org/10.1007/978-3-031-04826-5_22

[51] A. C. Hlas, K. L. Schuh, and S. M. Alessi. 2008. Native and non-native speakers in online and face-to-face discussions: Leveling the playing field. *Journal of Educational Technology Systems* 36, 4 (2008), 337–373. https://doi.org/10.2190/ET.36.4.b

[52] Amanda Holland-Minkley, Jakob Barnard, Valerie Barr, Grant Braught, Janet Davis, David Reed, Karl Schmitt, Andrea Tartaro, and James D. Teresco. 2023. Computer Science Curriculum Guidelines: A New Liberal Arts Perspective. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*. ACM, 617–623. https://doi.org/10.1145/3545945.3569793

[53] Yeonju Jang, Seongyune Choi, and Hyeoncheol Kim. 2022. Development and validation of an instrument to measure undergraduate students' attitudes toward the ethics of artificial intelligence (AT-EAI) and analysis of its difference by gender and experience of AI education. *Education and Information Technologies* 27, 8 (Sept. 2022), 11635–11667. https://doi.org/10.1007/s10639-022-11086-5

[54] Yasmin Kafai, Kristin Searle, Cristobal Martinez, and Bryan Brayboy. 2014. Ethnocomputing with electronic textiles: Culturally responsive open design to broaden participation in computing in American Indian youth and communities. *SIGCSE 2014 - Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, 241–246. https://doi.org/10.1145/2538862.2538903

[55] Maria Kallia and Quintin Cutts. 2021. Re-examining inequalities in computer science participation from a Bourdieusian sociological perspective. In *Proceedings of the 17th ACM Conference on International Computing Education Research*. 379–392. https://doi.org/10.1145/3446871.3469763

[56] Linda J. Sax Kathleen J. Lehman and Hilary B. Zimmerman. 2016. Women planning to major in computer science: Who are they and what makes them unique? *Computer Science Education* 26, 4 (2016), 277–298. https://doi.org/10.1080/08993408.2016.1271536 arXiv:https://doi.org/10.1080/08993408.2016.1271536

[57] Michael S. Kirkpatrick and Chris Mayfield. 2017. Evaluating an alternative CS1 for students with prior programming experience. *Proceedings of the Conference on Integrating Technology into Computer Science Education, ITiCSE*, 333–338. https://doi.org/10.1145/3017680.3017759

[58] Amy J Ko, Alannah Oleson, Neil Ryan, Yim Register, Benjamin Xie, Mina Tari, Matthew Davidson, Stefania Druga, and Dastyni Loksa. 2020. It is time for more critical CS education. *Commun. ACM* 63, 11 (2020), 31–33. https://doi.org/10.1145/3424000

[59] Minji Kong and Lori Pollock. 2023. Experiences Piloting a Diversity and Inclusion in Computing Innovations Course. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1* (Toronto ON, Canada) *(SIGCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 242–248. https://doi.org/10.1145/3545945.3569773

[60] Anagha Kulkarni, Ilmi Yoon, Pleuni S. Pennings, Kazunori Okada, and Carmen Domingo. 2018. Promoting diversity in computing. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE 2018)*. Association for Computing Machinery, New York, NY, USA, 236–241. https://doi.org/10.1145/3197091.3197145

[61] Amruth N Kumar. 2012. A study of stereotype threat in computer science. In *Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education*. 273–278. https://doi.org/10.1145/2325296.2325361

[62] Michael Lachney, Jean Ryoo, and Rafi Santo. 2021. Introduction to the Special Section on Justice-Centered Computing Education, Part 1. *ACM Transactions on Computing Education* 21 (12 2021). Issue 4. https://doi.org/10.1145/3477981

[63] Jennifer J Lee and Janice M Mccabe. 2021. Who speaks and who listens: Revisiting the chilly climate in college classrooms. *Gender & society* 35, 1 (2021), 32–60. https://doi.org/10.1177/0891243220977141

[64] Kathleen J Lehman, Kaitlin N.S Newhouse, Linda J Sax, Marcus B Weaver-Hightower, and Nancy S Niemi. 2020. Global Participation in Undergraduate Computing. In *The Wiley Handbook of Gender Equity in Higher Education*. John Wiley Sons, Inc, Hoboken, NJ, USA. https://doi.org/10.1002/9781119257639.ch15

[65] Albert Lionelle, McKenna Quam, Carla Brodley, and Catherine Gill. 2024. Does Curricular Complexity in Computer Science Influence the Representation of Women CS Graduates? *SIGCSE 2024 - Proceedings of the 55th ACM Technical Symposium on Computer Science Education* 1, 729–735. https://doi.org/10.1145/3626252.3630835

[66] Brian Magerko, Jason Freeman, Tom Mcklin, Mike Reilly, Elise Livingston, Scott McCoid, and Andrea Crews-Brown. 2016. EarSketch: A STEAM-Based Approach for Underrepresented Populations in High School Computer Science Education. *ACM Transactions on Computing Education* 16 (09 2016), 1–25. https://doi.org/10.1145/2886418

[67] 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). *ICER 2021 - Proceedings of the 17th ACM Conference on International Computing Education Research*, 145–156. https://doi.org/10.1145/3446871.3469761

[68] Jane Margolis, Rachel Estrella, Joanna Goode, Jennifer Jellison Holme, and Kim Nao. 2017. *Stuck in the Shallow End: Education, Race, and Computing*. The MIT Press.

[69] Cindy Marling and David Juedes. 2016. CS0 for Computer Science Majors at Ohio University. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (Memphis, Tennessee, USA) *(SIGCSE '16)*. Association for Computing Machinery, New York, NY, USA, 138–143. https://doi.org/10.1145/2839509.2844624

[70] A. Master and A. Meltzoff. 2020. Cultural Stereotypes and Sense of Belonging Contribute to Gender Gaps in STEM. *International Journal of Gender, Science and Technology* 12 (2020), 152–198. Issue 1.

[71] A Masters, A Meltzoff, and S. Cheryan. 2021. Gender stereotypes about interests start early and cause gender disparities in computer science and engineering. *PSYCHOLOGICAL AND COGNITIVE SCIENCES* 118 (2021).

[72] Jenna E. McChesney, Tara S. Behrend, and Alexander Glosenberg. 2022. Stereotypical descriptions of computer science career interests are not representative of many computer scientists. , 5990 pages. https://doi.org/10.1038/s41598-022-09522-0

[73] Nora McDonald and Shimei Pan. 2020. Intersectional AI: A Study of How Information Science Students Think about Ethics and Their Impact. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW2 (Oct. 2020), 1–19. https://doi.org/10.1145/3415218

[74] Susan Mckeever and Deirdre Lillis. 2021. Addressing the Recruitment and Retention of Female Students in Computer Science at Third Level. *ArXiv* abs/2110.06090. https://api.semanticscholar.org/CorpusID:238634175

[75] Mehrube Mehrubeoglu, Lifford McLauchlan, David Hicks, and Adetoun Yeaman. 2023. Engaged Student Learning through IoT-based Capstone Projects: Particular Look at Student Engagement in Historically Underrepresented Groups. In *2023 IEEE Frontiers in Education Conference (FIE)*. 01–07. https://doi.org/10.1109/FIE58773.2023.10343194 ISSN: 2377-634X.

[76] Briana B. Morrison, Beth A. Quinn, Steven Bradley, Kevin Buffardi, Brian Harrington, Helen H. Hu, Maria Kallia, Fiona McNeill, Oluwakemi Ola, Miranda Parker, Jennifer Rosato, and Jane Waite. 2022. Evidence for Teaching Practices that Broaden Participation for Women in Computing *(ITiCSE-WGR '21)*. Association for Computing Machinery, New York, NY, USA, 57–131. https://doi.org/10.1145/3502870.3506568

[77] Sukanya Kannan Moudgalya and Amanda Zeller. 2024. The Need for More Justice-Oriented Courses in Undergraduate Computer Science Curricula. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1* (Portland, OR, USA) *(SIGCSE 2024)*. Association for Computing Machinery, New York, NY, USA, 888–894. https://doi.org/10.1145/3626252.3630949

[78] Felix Muzny, Megan Giordano, Emma Sommers, and Carla E. Brodley. 2024. Collecting, Analyzing, and Acting on Intersectional, Longitudinal Data and Pass/Fail/Withdraw Rates in Computing Courses. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1* (Portland, OR, USA) *(SIGCSE 2024)*. Association for Computing Machinery, New York, NY, USA, 909–915. https://doi.org/10.1145/3626252.3630806

[79] Felix Muzny and Michael D. Shah. 2023. Teaching Assistant Training: An Adjustable Curriculum for Computing Disciplines. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1* (Toronto ON, Canada) *(SIGCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 430–436. https://doi.org/10.1145/3545945.3569866

[80] Kendall Nakai and Philip J. Guo. 2023. Uncovering the Hidden Curriculum of University Computing Majors via Undergraduate-Written Mentoring Guides: A Learner-Centered Design Workflow. In *Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 1* (Chicago, IL, USA) *(ICER '23)*. Association for Computing Machinery, New York, NY, USA, 63–77. https://doi.org/10.1145/3568813.3600113

[81] Break Through Tech National. 2022. *Achieving Greater Gender Equity in Tech; The Sprintership™ Program.* Retrieved September 2nd, 2024 from https://www.breakthroughtech.org/about-us/thought-leadership/achieving-greater-gender-equity-in-tech/

[82] Oxford English Dictionary (OED). 1921. *Racialized, Adj.* Retrieved November 23, 2024 from https://www.oed.com/dictionary/racialized_adj

[83] National Association of Colleges and Employers (NACE). 2020. *Open the Door: Disparities in Paid Internships.* Retrieved Dec 5, 2024 from https://www.naceweb.org/diversity-equity-and-inclusion/trends-and-predictions/open-the-door-disparities-in-paid-internships/

[84] National Association of Colleges and Employers (NACE). 2020. *Women are Underrepresented Among Paid Interns.* Retrieved Dec 5, 2024 from https://www.naceweb.org/diversity-equity-and-inclusion/trends-and-predictions/women-are-underrepresented-among-paid-interns

[85] Alannah Oleson, Meron Solomon, and Amy J Ko. 2020. Computing students' learning difficulties in HCI education. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–14. https://doi.org/10.1145/3313831.3376149

[86] ACM/IEEE-CS Joint Task Force on Computing Curricula. 2023. *CS2023: ACM/IEEE-CS/AAAI Computer Science Curricula.* Retrieved Dec 5, 2024 from https://csed.acm.org/

[87] Pankati Patel, Patricia Morreale, Yulia Kumar, Daehan Kwak, Jean Chu, Rose Garcia, and Margaret Burnett. 2023. Embedding Equitable Design in the CS Computing Curricula. *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 2*, 1332. https://doi.org/10.1145/3545947.3576278

[88] Elizabeth Patitsas, Michelle Craig, and Steve Easterbrook. 2014. A historical examination of the social factors affecting female participation in computing. In *Proceedings of the 2014 conference on Innovation & technology in computer science education*. 111–116. https://doi.org/10.1145/2591708.2591731

[89] Joan Peckham, Lisa L Harlow, David A Stuart, Barbara Silver, Helen Mederer, and Peter D Stephenson. 2007. Broadening participation in computing: issues and challenges. *ACM SIGCSE Bulletin* 39, 3 (2007), 9–13. https://doi.org/10.1145/1269900.1268790

[90] Michela Pedroni, Manuel Oriol, and Bertrand Meyer. 2011. The prior experience of entering CS students. *Technical report* 712 (2011).

[91] Piumi Perera, Geethya Tennakoon, Supunmali Ahangama, Rangana Panditharathna, and Buddhika Chathuranga. 2021. A Systematic Mapping of Introductory Programming Languages for Novice Learners. *IEEE Access* 9 (2021), 88121–88136. https://doi.org/10.1109/ACCESS.2021.3089560

[92] Marta Peña, Noelia Olmedo-Torre, Elisabet Mas de les Valls, and Amaia Lusa. 2021. Introducing and Evaluating the Effective Inclusion of Gender Dimension in STEM Higher Education. *Sustainability* 13 (4 2021), 4994. Issue 9. https://doi.org/10.3390/su13094994

[93] Jeria Quesenberry and Carol Frieze. 2016. *Change culture, not curriculum, to get more women into computer science.* Retrieved January 17, 2024 from https://blog.google/outreach-initiatives/education/change-culture-not-curriculum-to-ge/

[94] Yolanda A Rankin, Jakita O Thomas, and Sheena Erete. 2021. Black women speak: Examining power, privilege, and identity in CS education. *ACM Transactions on Computing Education (TOCE)* 21, 4 (2021), 1–31. https://doi.org/10.1145/3451344

[95] Lauren Rich, Heather Perry, and Mark Guzdial. 2004. A CS1 Course Designed to Address Interests of Women. http://coweb.cc.gatech.edu/cs1315

[96] Audrey Smith Rorrer, Joseph Allen, and Huifang Zuo. 2018. A National Study of Undergraduate Research Experiences in Computing: Implications for Culturally Relevant Pedagogy. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (Baltimore, Maryland, USA) *(SIGCSE '18)*. Association for Computing Machinery, New York, NY, USA, 604–609. https://doi.org/10.1145/3159450.3159510

[97] Audrey S. Rorrer, Sepi Hejazi Moghadam, Breauana Spencer, and Ting Sun. 2020. eCSR: Creating Intensive Research Experiences that Cultivate Community for Undergraduate Women & Women of Color. In *2020 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*, Vol. 1. 1–4. https://doi.org/10.1109/RESPECT49803.2020.9272423

[98] David Rose. 1999. Universal Design for Learning. *Journal of Special Education Technology* 15 (1999), 67–70. Issue 1. https://doi.org/10.1177/016264340001500108

[99] Monique Ross, Mark A. Weiss, Lilia Minaya, Andrew Laginess, Disha Patel, and Kathleen Quardokus Fisher. 2022. Removing a Barrier: Analysis of the Impact of Removing Calculus and Physics from CS on Employability, Salary, and Broadening Participation. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education - Volume 1 (SIGCSE 2022, Vol. 1)*. Association for Computing Machinery, New York, NY, USA, 460–466. https://doi.org/10.1145/3478431.3499356

[100] Isabel Ruthotto, Quintin Kreth, Jillian Stevens, Clare Trively, and Julia Melkers. 2020. Lurking and participation in the virtual classroom: The effects of gender, race, and age among graduate students in computer science. *Computers & Education* 151 (2020), 103854. https://doi.org/10.1016/j.compedu.2020.103854

[101] Pooja Sankar, Jessica Gilmartin, and Melissa Sobel. 2015. An examination of belongingness and confidence among female computer science students. *Acm Sigcas Computers and Society* 45, 2 (2015), 7–10. https://doi.org/10.1145/2809957.2809960

[102] Linda J. Sax, Kathleen J. Lehman, and Christina Zavala. 2017. Examining the Enrollment Growth: Non-CS Majors in CS1 Courses. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (Seattle, Washington, USA) *(SIGCSE '17)*. Association for Computing Machinery, New York, NY, USA, 513–518. https://doi.org/10.1145/3017680.3017781

[103] Amber Settle and Theresa Steinbach. 2016. Improving retention and reducing isolation via a linked-courses learning community. In *Proceedings of the 17th Annual Conference on Information Technology Education*. 34–39. https://doi.org/10.1145/2978192.2978212

[104] E. Seymour, G. Melton, H. Thiry, L. Laursen, and Hunter A.-B. 2010. *Undergraduate Research in the Sciences: Engaging Students in Real Science.* Wiley.

[105] Robert M. Siegfried, Katherine G. Herbert-Berger, Kees Leune, and Jason P. Siegfried. 2021. Trends of commonly used programming languages in CS1 and CS2 learning. *ICCSE 2021 - IEEE 16th International Conference on Computer Science and Education*, 407–412. https://doi.org/10.1109/ICCSE51940.2021.9569444

[106] Simon, Andrew Luxton-Reilly, Vangel V Ajanovski, Eric Fouh, Christabel Gonsalvez, Juho Leinonen, Jack Parkinson, Matthew Poole, and Neena Thota. 2019. Pass rates in introductory programming and in other stem disciplines. In *Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education*. Association for Computing Machinery, New York, NY, USA, 53–71. https://doi.org/10.1145/3344429.3372502

[107] Ahmad Slim, Jarred Kozlick, Gregory L. Heileman, and Chaouki T. Abdallah. 2014. The Complexity of University Curricula According to Course Cruciality. In *2014 Eighth International Conference on Complex, Intelligent and Software Intensive Systems*. 242–248. https://doi.org/10.1109/CISIS.2014.34

[108] Robert H. Sloan and Patrick Troy. 2008. CS 0.5: a better approach to introductory computer science for majors. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education* (Portland, OR, USA) *(SIGCSE '08)*. Association for Computing Machinery, New York, NY, USA, 271–275. https://doi.org/10.1145/1352135.1352230

[109] Kathleen N Smith and Joy Gaston Gayles. 2018. "Girl power": Gendered academic and workplace experiences of college women in engineering. *Social sciences* 7, 1 (2018), 11. https://doi.org/10.3390/socsci7010011

[110] Chris Stephenson, Alison Derbenwick Miller, Christine Alvarado, Lecia Barker, Valerie Barr, Tracy Camp, Carol Frieze, Colleen Lewis, Erin Cannon Mindell, Lee Limbird, et al. 2018. *Retention in computer science undergraduate programs in the us: Data challenges and promising interventions.* ACM. Retrieved Dec 5, 2024 from https://www.acm.org/binaries/content/assets/education/retention-in-cs-undergrad-programs-in-the-us.pdf

[111] Gijsbert Stoet and David C. Geary. 2018. The Gender-Equality Paradox in Science, Technology, Engineering, and Mathematics Education. *Psychological Science* 29, 4 (2018), 581–593. https://doi.org/10.1177/0956797617741719 PMID: 29442575.

[112] Anya Tafliovich, Jennifer Campbell, and Andrew Petersen. 2013. A student perspective on prior experience in CS1. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education* (Denver, Colorado, USA) *(SIGCSE '13)*. Association for Computing Machinery, New York, NY, USA, 239–244. https://doi.org/10.1145/2445196.2445270

[113] Charles C Tappert, Andreea Cotoranu, and John V Monaco. 2015. A Real-World-Projects Capstone Course in Computing: A 15-year Experience. In *Proceedings of the EDSIG Conference*. n3476.

[114] Diversity The Interdepartmental Terminology Committee on Equity and Inclusion. 2022. *Guide on Equity, Diversity and Inclusion Terminology.* Retrieved November 23, 2024 from https://www.noslangues-ourlanguages.gc.ca/

en/publications/equite-diversite-inclusion-equity-diversity-inclusion-eng

[115] Luther Tychonievich and Mark Sherriff. 2022. Engineering a Complete Curriculum Overhaul. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education - Volume 1* (Providence, RI, USA) *(SIGCSE 2022)*. Association for Computing Machinery, New York, NY, USA, 453–459. https://doi.org/10.1145/3478431.3499287

[116] Phil Ventura and Bina Ramamurthy. 2004. Wanted: CS1 students. No experience required. *ACM SIGCSE Bulletin* 36, 1 (2004), 240–244. https://doi.org/10.1145/1028174.971387

[117] Anna Vitores and Adriana Gil-Juárez. 2015. The trouble with 'women in computing': a critical examination of the deployment of research on the gender gap in computer science. *Journal of Gender Studies* 25, 6 (2015), 666–680. https://doi.org/10.1080/09589236.2015.1087309

[118] Raechel Wallker. 2023. Systemic Justice Capstone Project: Enabling Students to Mitigate Systemic Oppression Through Data Activism. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 2* (Toronto ON, Canada) *(SIGCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 1237. https://doi.org/10.1145/3545947.3573293

[119] Alicia Nicki Washington. 2020. When twice as good isn't enough the case for cultural competence in computing. *SIGCSE 2020 - Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 213–219. https://doi.org/10.1145/3328778.3366792

[120] Alicia Nicki Washington, Shaundra Daily, and Cecilé Sadler. 2022. Identity-Inclusive Computing: Learning from the Past; Preparing for the Future. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 2* (Providence, RI, USA) *(SIGCSE 2022)*. Association for Computing Machinery, New York, NY, USA, 1182. https://doi.org/10.1145/3478432.3499172

[121] Kelly Widdicks, Alice Ashcroft, Emily Winter, and Lynne Blair. 2021. Women's Sense of Belonging in Computer Science Education: The Need for a Collective Response. In *Proceedings of the 2021 Conference on United Kingdom & Ireland Computing Education Research* (Glasgow, United Kingdom) *(UKICER '21)*. Association for Computing Machinery, New York, NY, USA, Article 4, 7 pages. https://doi.org/10.1145/3481282.3481288

[122] Adriana Wilde. 2023. Equality, Diversity, and Inclusion in the CS Curriculum, In Proceedings of the 2023 Conference on United Kingdom & Ireland Computing Education Research (Swansea, Wales Uk). *ACM International Conference Proceeding Series*, Article 12, 1 pages. https://doi.org/10.1145/3610969.3611125

[123] Bill Manaris William Bares and Renée McCauley. 2018. Gender equity in computer science through computing in the arts – a six-year longitudinal study. *Computer Science Education* 28, 3 (2018), 191–210. https://doi.org/10.1080/08993408.2018.1519322 arXiv:https://doi.org/10.1080/08993408.2018.1519322

[124] Marcelo Worsley and David Bar-El. 2020. Inclusive Making: designing tools and experiences to promote accessibility and redefine making. *Computer Science Education* 32, 2 (2020), 155–187. https://doi.org/10.1080/08993408.2020.1863705

[125] Evelyn Yarzebinski. 2021. CRN At-A-Glance. Retrieved July 2, 2024 from https://cra.org/crn/wp-content/uploads/sites/7/2021/10/OCT-21-CRN.pdf

[126] Kimberly Michelle Ying, Alexia Charis Martin, Fernando J. Rodríguez, and Kristy Elizabeth Boyer. 2021. CS1 Students' Perspectives on the Computer Science Gender Gap: Achieving Equity Requires Awareness. In *2021 Conference on Research in Equitable and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*. 1–9. https://doi.org/10.1109/RESPECT51740.2021.9620659

URLs for Australian universities

[AUS127] University of Adelaide. 2024. Bachelor of Computer Science (Advanced) Domestic. https://www.adelaide.edu.au/degree-finder/bcomp_bcmpsci.html (Accessed: 2024-07-05).

[AUS128] University of Adelaide. 2024. Bachelor of Computer Science (Advanced) International. https://www.adelaide.edu.au/degree-finder/hcomp_hbcompsc.html (Accessed: 2024-07-05).

[AUS129] University of Melbourne. 2024. Bachelor of Science (major in Computing and Software Systems). https://study.unimelb.edu.au/find/courses/major/computing-and-software-systems/structure/#nav (Accessed: 2024-07-05).

[AUS130] University of New South Wales (UNSW Syndey). 2024. Bachelor of Computer Science. https://www.unsw.edu.au/study/undergraduate/bachelor-of-computer-science (Accessed: 2024-07-05).

[AUS131] University of New South Wales (UNSW Syndey). 2024. Bachelor of Computer Science Honous. https://www.unsw.edu.au/study/undergraduate/bachelor-of-computer-science-honours (Accessed: 2024-07-05).

[AUS132] University of Queensland. 2024. Bachelor of Computer Science. https://study.uq.edu.au/study-options/programs/bachelor-computer-science-2451 (Accessed: 2024-07-05).

[AUS133] University of Queensland. 2024. Bachelor of Computer Science Honours. https://study.uq.edu.au/study-options/programs/bachelor-computer-science-honours-2452 (Accessed: 2024-07-05).

[AUS134] University of Queensland. 2024. Bachelor of Computer Science Requirements. https://my.uq.edu.au/programs-courses/requirements/program/2451/2025 (Accessed: 2024-07-05).

[AUS135] University of Queensland. 2024. Bachelor of Computer Science Semesters. https://www.eait.uq.edu.au/employability/industry-partnerships/student-and-graduate-engagement#1 (Accessed: 2024-07-05).

[AUS136] University of Sydney. 2024. Bachelor of (Advanced) Computing. https://www.sydney.edu.au/courses/courses/uc/bachelor-of-advanced-computing.html (Accessed: 2024-07-05).

[AUS137] University of Technology Sydney. 2024. Bachelor of Computing Science. https://www.uts.edu.au/study/find-a-course/bachelor-computing-science (Accessed: 2024-07-05).

[AUS138] University of Technology Sydney. 2024. Bachelor of Computing Science Honours. https://www.uts.edu.au/study/find-a-course/bachelor-computing-science-honours (Accessed: 2024-07-05).

[AUS139] Australian National University. 2024. Programs and Courses Bachelor of (Advanced) Computing. https://programsandcourses.anu.edu.au/2024/program/aacom (Accessed: 2024-07-05).

[AUS140] Monash University. 2024. Courses Bachelor of Computer Science Year 4. https://handbook.monash.edu/2024/courses/C3702 (Accessed: 2024-06-30).

[AUS141] Monash University. 2024. Courses Bachelor of Computer Science Years 1–3. https://handbook.monash.edu/2024/courses/C2001?year=2024 (Accessed: 2024-06-30).

[AUS142] Macquarie University. 2024. Courses Bachelor of Information Technology. https://www.mq.edu.au/study/find-a-course/courses/bachelor-of-information-technology (Accessed: 2024-06-30).

[AUS143] Monash University. 2024. Entry Requirements. https://www.monash.edu/study/courses/find-a-course/computer-science-c2001#entry-requirements-2 (Accessed: 2024-06-30).

[AUS144] Swinburne University. 2024. Bachelor of Computer Science. https://www.swinburne.edu.au/course/undergraduate/bachelor-of-computer-science/ (Accessed: 2024-07-05).

[AUS145] Swinburne University. 2024. Bachelor of Computer Science Honours. https://www.swinburne.edu.au/course/undergraduate/bachelor-of-computer-science-honours/ (Accessed: 2024-07-05).

URLs for Canadian universities

[CA146] Université de Montréal. 2024. Baccalauréat en informatique Courses. https://diro.umontreal.ca/programmes-cours/premier-cycle/ (Accessed: 2024-07-05).

[CA147] Université de Montréal. 2024. Requirements. https://admission.umontreal.ca/programmes/baccalaureat-en-informatique/ (Accessed: 2024-07-05).

[CA148] Université du Québec (à Montréal). 2024. Baccalauréat en informatique et génie logiciel. https://info.uqam.ca/baccalaur%C3%A9at_en_informatique_et_g%C3%A9nie_logiciel/ (Accessed: 2024-07-05).

[CA149] Université du Québec (à Montréal). 2024. Courses. https://etudier.uqam.ca/recherche-cours?disc_code=030&disc_nom=Informatique (Accessed: 2024-07-05).

[CA150] University of Alberta. 2024. Bachelor of Science Computing Science. https://calendar.ualberta.ca/preview_program.php?catoid=44&poid=55570#MajorInComputingScienceRequirements (Accessed: 2024-07-05).

[CA151] University of British Columbia. 2024. Bachelor of Science in Computer Science. https://vancouver.calendar.ubc.ca/faculties-colleges-and-schools/faculty-science/bachelor-science/computer-science (Accessed: 2024-06-30).

[CA152] University of Toronto. 2024. Bachelor of Science, Computer Science Specialist. https://artsci.calendar.utoronto.ca/section/Computer-Science (Accessed: 2024-07-05).

[CA153] University of Waterloo. 2024. Bachelor of Computer Science. https://uwaterloo.ca/academic-calendar/undergraduate-studies/catalog#/programs/view/65ea084cf40b273115278dbf (Accessed: 2024-07-05).

[CA154] McGill University. 2024. Bachelor of Science, Computer Science. https://www.mcgill.ca/study/2024-2025/faculties/science/undergraduate/programs/bachelor-science-bsc-major-computer-science (Accessed: 2024-06-05).

[CA155] McGill University. 2024. Co-op Course. https://academiccalendars.romcmaster.ca/preview_program.php?catoid=41&poid=21886&hl=%22computer+science%22&returnto=search (Accessed: 2024-06-05).

[CA156] McGill University. 2024. Courses. https://www.cs.mcgill.ca/academic/courses/"McMasterUniversityBachelorofAppliedScienceinComputerScience (Accessed: 2024-06-05).

[CA157] McGill University. 2024. Credit Time Limits. https://www.mcgill.ca/study/2024-2025/faculties/science/undergraduate/ug_sci_time_credit_limit_degree (Accessed: 2024-06-05).

[CA158] McGill University. 2024. Honours Co-op Course. https://academiccalendars.romcmaster.ca/preview_program.php?catoid=56&poid=28104 (Accessed:

2024-06-05).
[CA159] McGill University. 2024. Interships. https://www.cs.mcgill.ca/undergrad/future/internship/ (Accessed: 2024-06-05).
[CA160] McGill University. 2024. Requirements. https://www.mcgill.ca/undergraduate-admissions/apply/requirements/university (Accessed: 2024-06-05).
[CA161] McGill University. 2024. Transfer Information. https://www.cs.mcgill.ca/undergrad/future/transfer/ (Accessed: 2024-06-05).
[CA162] McGill University. 2024. Undergraduate Programs. https://www.cs.mcgill.ca/academic/undergrad/program/ (Accessed: 2024-06-05).
[CA163] Simon Fraser University. 2024. Bachelor of Science, Computing Science. https://www.sfu.ca/students/calendar/2024/fall/programs/computing-science/major/bachelor-of-science-or-bachelor-of-arts.html (Accessed: 2024-07-05).
[CA164] Simon Fraser University. 2024. Future Students. https://www.sfu.ca/computing/prospective-students/undergraduate-students/admissions.html (Accessed: 2024-07-05).
[CA165] Western University. 2024. Bachelor of Science, Specialization in Computer Science. https://www.westerncalendar.uwo.ca/Modules.cfm?ModuleID=21125&SelectedCalendar=Live&ArchiveID= (Accessed: 2024-07-05).

URLs for European universities

[EU166] KU Leuven. 2024. Bachelor in Computer Science. https://www.kuleuven.be/opleidingen/bachelor-informatica (Accessed: 2024-07-05).
[EU167] Imperial College London. 2024. Bachelor of Engineering in Computing. https://www.imperial.ac.uk/computing/prospective-students/ug/beng-meng-computing/beng-comp/ (Accessed: 2024-07-01).
[EU168] Imperial College London. 2024. Requirements. https://www.imperial.ac.uk/study/courses/undergraduate/computing-beng/ (Accessed: 2024-07-01).
[EU169] University College London. 2024. Bachelor of Science in Computer Science. https://www.ucl.ac.uk/prospective-students/undergraduate/degrees/computer-science-bsc#tab1-year1 (Accessed: 2024-07-01).
[EU170] University of Cambridge. 2024. Bachelor of Arts Computer Science. https://www.undergraduate.study.cam.ac.uk/courses/computer-science-ba-hons-meng (Accessed: 2024-07-01).
[EU171] University of Edinburgh. 2024. Bachelor of Science Computer Science. http://www.drps.ed.ac.uk/23-24/dpt/utcmpsibe.htm (Accessed: 2024-06-18).
[EU172] University of Edinburgh. 2024. Requirements. https://www.ed.ac.uk/studying/undergraduate/degrees/index.php?action=view&code=G400 (Accessed: 2024-06-18).
[EU173] Technical University of Munich. 2024. Bachelor of Science in Informatics. https://www.cit.tum.de/cit/studium/studiengaenge/bachelor-informatik/studienplan/#c3270 (Accessed: 2024-07-01).
[EU174] Technical University of Munich. 2024. Courses Bachelor of Science in Informatics. https://campus.tum.de/tumonline/ee/ui/ca2/app/desktop/#/pl/ui/$ctx/wbstpcs.showSpoTree?$ctx=design=ca2;header=max;lang=de&pStpStpNr=4998 (Accessed: 2024-07-01).
[EU175] University of Oxford. 2024. Bachelor of Arts Computer Science. https://www.cs.ox.ac.uk/admissions/undergraduate/courses/cs.html (Accessed: 2024-07-01).
[EU176] Paris Sciences Lettres (PSL). 2024. First Year Licence d'Informatique (L3). https://dauphine.psl.eu/formations/licences/l1-mathematiques-informatique/programme#label_modal_labtd65ii9 (Accessed: 2024-07-05).
[EU177] Paris Sciences Lettres (PSL). 2024. Licence d'Informatique (L3). https://dauphine.psl.eu/formations/licences/l1-mathematiques-informatique/programme (Accessed: 2024-07-05).
[EU178] Paris Sciences Lettres (PSL). 2024. Second Year Licence d'Informatique (L3). https://dauphine.psl.eu/en/training/bachelors-degrees/bachelors-degree-in-organizational-computer-science/mathematics-and-computer-science-bachelors-year-2/program (Accessed: 2024-07-05).
[EU179] Paris Sciences Lettres (PSL). 2024. Third Year Licence d'Informatique (L3). https://dauphine.psl.eu/formations/licences/informatique-des-organisations/l3-methodes-informatiques-appliquees-pour-la-gestion-des-entreprises/programme (Accessed: 2024-07-05).
[EU180] ETH Zurich. 2024. Bachelor's in Computer Science. https://inf.ethz.ch/studies/bachelor.html (Accessed: 2024-07-01).
[EU181] École Polytechnique Fédérale de Lausanne (EPFL). 2024. Bachelor's in Computer Science. https://docs.google.com/spreadsheets/d/e/2PACX-1vTxK3SpOgjVl_v2ZOucQDeqNt2x1hCF0YeL4_uXurJvcINpb_AHxPLSHtnL9ddYxLtDbxNSC55fhyX3/pubhtml (Accessed: 2024-07-05).

URLs for US universities

[US182] UC Berkeley. 2024. Bachelor of Arts in Computer Science. https://eecs.berkeley.edu/resources/undergrads/cs/ (Accessed: 2024-06-25).
[US183] University of Illinois Urbana-Champaign. 2024. Bachelor of Science in Computer Science. http://catalog.illinois.edu/undergraduate/engineering/computer-science-bs/#degreerequirementstext (Accessed: 2024-06-25).
[US184] California Institute of Technology (Caltech). 2024. Bachelor of Science in Computer Science. https://catalog.caltech.edu/current/information-for-undergraduate-students/graduation-requirements-all-options/computer-science-option-and-minor-cs/ (Accessed: 2024-07-05).
[US185] Massachusetts Institute of Technology (MIT). 2024. Bachelor of Science in Computer Science and Engineering. https://catalog.mit.edu/degree-charts/computer-science-engineering-course-6-3/ (Accessed: 2024-06-25).
[US186] Massachusetts Institute of Technology (MIT). 2024. Courses. https://catalog.mit.edu/schools/engineering/electrical-engineering-computer-science/#subjectstext (Accessed: 2024-06-25).
[US187] Massachusetts Institute of Technology (MIT). 2024. Degree Requirements. https://eecsis.mit.edu/degree_requirements.html#track (Accessed: 2024-06-25).
[US188] University of Washington. 2024. Bachelor of Science in Computer Science. https://www.washington.edu/students/gencat/program/S/ComputerScience-210.html (Accessed: 2024-07-05).
[US189] Cornell University. 2024. Bachelor of Science in Computer Science. https://www.cs.cornell.edu/undergrad/csmajor (Accessed: 2024-06-25).
[US190] Carnegie Mellon University. 2024. Bachelor of Science in Computer Science. https://csd.cmu.edu/academics/bachelors/overview (Accessed: 2024-06-25).
[US191] Carnegie Mellon University. 2024. Courses. http://coursecatalog.web.cmu.edu/schools-colleges/schoolofcomputerscience/undergraduatecomputerscience/#bscurriculumtextcontainer (Accessed: 2024-06-25).
[US192] Harvard University. 2024. Bachelor of Arts (A.B.) in Computer Science. https://seas.harvard.edu/computer-science/bachelors-degree-computer-science (Accessed: 2024-07-05).
[US193] Princeton University. 2024. Bachelor of Arts in Computer Science. https://www.cs.princeton.edu/ugrad/class-of-2025-departmental-requirements (Accessed: 2024-07-05).
[US194] Stanford University. 2024. Bachelor of Science in Computer Science. https://www.cs.stanford.edu/academics/academics-bachelors-program (Accessed: 2024-06-25).

URLs for equity-focused universities

[USEF195] Carleton College. 2024. Computer Science, BA. https://www.carleton.edu/computer-science/major/requirements/ (Accessed: 2024-07-05).
[USEF196] Harvey Mudd College. 2024. Computer Science, B.S. https://catalog.hmc.edu/content.php?catoid=22&navoid=1149 (Accessed: 2024-07-05).
[USEF197] Morehouse College. 2024. Computer Science, B.S. https://morehouse.edu/academics/majors/computer-science#major (Accessed: 2024-07-05).
[USEF198] Swarthmore College. 2024. Computer Science, BA. https://www.swarthmore.edu/computer-science/honors-majors-and-minors (Accessed: 2024-07-05).
[USEF199] Spelman College. 2024. Computer Science, B.S. https://www.spelman.edu/_1_Docs-and-Files/spelman-course-sequence-booklet-2024.pdf (Accessed: 2024-07-05).
[USEF200] Swarthmore College. 2024. Computer Science Courses. https://catalog.swarthmore.edu/preview_program.php?catoid=29&poid=2211#computersciencecourses (Accessed: 2024-07-05).
[USEF201] Spelman College. 2024. Courses. https://www.spelman.edu/academics/computer-and-information-sciences/index.html,https://www.spelman.edu/_1_Docs-and-Files/academics/college-bulletin/college-bulletin-2020-22.pdf (Accessed: 2024-07-05).
[USEF202] Wellesley College. 2024. Computer Science, BA. https://catalog.wellesley.edu/display.php?pos=25&doc_type=computer+science+major (Accessed: 2024-07-05).
[USEF203] University of California Irvine (UCI). 2024. Computer Science, B.S. https://catalogue.uci.edu/donaldbrenschoolofinformationandcomputersciences/departmentofcomputerscience/computerscience_bs/#requirementstext (Accessed: 2024-07-05).
[USEF204] University of California Irvine (UCI). 2024. Computer Science Courses. https://catalogue.uci.edu/donaldbrenschoolofinformationandcomputersciences/departmentofcomputerscience/#majorsminorandgraduateprogramstext (Accessed: 2024-07-05).
[USEF205] University of Texas Austin. 2024. Computer Science, B.S. https://www.cs.utexas.edu/undergraduate-program/academics (Accessed: 2024-07-05).
[USEF206] University of Texas Austin. 2024. Courses. https://catalog.utexas.edu/general-information/coursesatoz/c-s/ (Accessed: 2024-07-05).
[USEF207] Howard University. 2024. Computer Science, B.S. https://catalogue.howard.edu/computer-science/computer-science-bs,https://cea.howard.edu/sites/cea.howard.edu/files/2021-01/CS_Undergraduate_Handbook_

2016_v1.doc%20-%20CS_Undergraduate_Handbook_2018.pdf (Accessed: 2024-07-05).

URLs for university demographic data

[Stat208] EPFL. 2020. School of Computer & Communication Sciences(IC) Gender Monitoring. https://www.epfl.ch/about/equality/wp-content/uploads/2020/05/IC_GenderMonitoring2020_En.pdf (Accessed: 2024-12-05).
[Stat209] HESA. 2022. Higher Education Student Data. https://www.hesa.ac.uk/data-and-analysis/students (Accessed: 2024-12-05).
[Stat210] ICL. 2019. Women in Computing, Faculty of Engineering, Imperial College London. https://www.imperial.ac.uk/computing/about/equality-and-diversity/women-in-computing/ (Accessed: 2024-12-05).
[Stat211] mediaTUM. 2022. Technical University of Munich. https://mediatum.ub.tum.de/doc/1731943/1731943.pdf (Accessed: 2024-12-05).
[Stat212] University of Cambridge. 2022. Undergraduate Admissions Statistics – 2022 cycle. https://www.undergraduate.study.cam.ac.uk/files/publications/undergrad_admissions_statistics_2022_cycle.pdf (Accessed: 2024-12-05).
[Stat213] Council of Ontario Universities. 2020–2021. Common University Data Ontario (CUDO). https://cudo.ouac.on.ca/ (Accessed: 2024-12-05).
[Stat214] Oxford. 2023. Admissions process summary for the 2023–24 cycle. https://www.cs.ox.ac.uk/admissions/undergraduate/admissions_statistics/public_report2023.html (Accessed: 2024-12-05).
[Stat215] Institutional Research and Simon Fraser University Planning. 2023–2024. Academic Information Report - School of Computing Science. https://www.sfu.ca/content/dam/sfu/irp/departments/CMPT_tables.pdf (Accessed: 2024-12-05).
[Stat216] UBC Senate. 2022–2023. University of British Columbia Annual Enrolment Report. https://scs-senate-2021.sites.olt.ubc.ca/files/UBC-Annual-Enrolment-Report-2022-23.pdf (Accessed: 2024-12-05).
[Stat217] McGill University. 2020–2021. McGill Enrolment Services Reports 2020-2021. https://public.tableau.com/app/profile/apb.mcgill/viz/McGillEnrolmentServicesReports2020-2021-GraduationStatistics/SummarybyLevel (Accessed: 2024-12-05).
[Stat218] ETH Zurich. 2020. Equality Monitoring. https://ethz.ch/content/dam/ethz/associates/services/Anstellung-Arbeiten/diversity/Strategie_und_Zahlen/monitoring-und-studien/2019-2020/Equality_Monitoring_general_report19-20_engl.pdf (Accessed: 2024-12-05).

Appendix A Survey Questions

Please see tables 32 for our student survey questions, and table 33 for our educator survey questions.

Appendix B Interview Questions

Please see table 34 for student interview questions. These questions are designed for student interviews that follow the survey. Please see table 35 for our educators' (and advisors') interview questions.

Appendix C Course Code Categories

Please see table 36 for a list of the course code categories.

Note: This is not a complete list of all the courses mentioned in the survey respondents' entries; therefore, it is not a complete list of all the courses that our analysis is based on. In answers to questions S1Q23-S1Q28 we also observed the following types of responses: (1) course categories (e.g., "Theory courses"), (2) elements of the course that the answer is based on (e.g., "functional programming as a paradigm"), and (3) full course names (e.g., "Analysis of algorithms"). We did not need to find courses on university websites to categorize the answer in course categories in any of the cases above because the responses provided enough information to map the answer to the course categories. In addition, some respondents answered the questions S1Q23-S1Q28 by mentioning only a number (course number without a code such as "COMP"). In these cases, we were unable to determine the course since we could not find enough information about the university or any other information to help us find the course mentioned to categorize it.

Table 32: Survey Questions: Students

No.	Question	Question Type
	Your Observations of Computer Science	
S1Q1	Computer Science is important for society.	5-Point Likert Scale
S1Q2	I can be successful as a Computer Scientist.	5-Point Likert Scale
S1Q3	People of different gender identities are accepted among computer science students.	5-Point Likert Scale
S1Q4	People of different sexual orientations are accepted among computer science students.	5-Point Likert Scale
S1Q5	People of different races and cultures are accepted among computer science students.	5-Point Likert Scale
S1Q6	People with accessibility needs are accepted among computer science students.	5-Point Likert Scale
S1Q7	People with different socioeconomic backgrounds are accepted among computer science students.	5-Point Likert Scale
S1Q8	I currently am ... (a) a university student who is interested in applying for a computer science program. (b) a university student who is interested in transferring into a computer science program. (c) a first-year computer science student. (d) a second to fourth-year computer science student. (e)a senior (about to graduate) computer science student. (f) a computer science graduate. (g) a university student trying to transfer out of computer science. (h) a university student majoring in a discipline other than computer science, taking a computer science course. (i) other [please specify]	Multiple-Choice
	Your Computing Science Course Experience	
S1Q9	I am satisfied with my decision to enroll in computer science.	5-Point Likert Scale
S1Q10	I rate the diversity of students in my computer science courses as...	5-Point Likert Scale
S1Q11	I feel I have things in common with other students in my course.	5-Point Likert Scale
S1Q12	My computer science courses feel relevant to my life.	5-Point Likert Scale
S1Q13	The computer science courses offered in my program feel relevant to my future career.	5-Point Likert Scale
S1Q14	I found the material in my computer science courses relatable and relevant to my career.	5-Point Likert Scale
S1Q15	I found the examples used in my computer science courses relatable.	5-Point Likert Scale
S1Q16	I have difficulties following lectures because of the language and vocabulary used in my course and by my instructors.	5-Point Likert Scale
S1Q17	I would recommend my program to other people like me.	5-Point Likert Scale
S1Q18	The choice of examples/project topics used in courses makes me feel motivated to learn the material.	5-Point Likert Scale
S1Q19	The way of thinking about and modelling problems in my computer science courses aligns with my own.	5-Point Likert Scale
S1Q20	I know how to use my core computer science skills to solve problems I am interested in.	5-Point Likert Scale
S1Q21	What types of computer science applications are you most drawn to?	Multi-Select
S1Q22	What areas of computer science are you most drawn to?	Multi-Select
S1Q23	Which computer science courses did you enjoy taking the most? Why?	Open-ended
S1Q24	Which computer science courses did you enjoy taking the least? Why?	Open-ended
S1Q25	Which computer science units did you perform best in?	Open-ended
S1Q26	Which computer science units did you perform worst in?	Open-ended
S1Q27	Did you at any time in your computing science studies, feel frustrated with a specific course content? if yes, When? Which course? and why?	Open-ended
S1Q28	Can you think of a time in your studies where a particular class, topic, or example was particularly engaging or relevant to your interests or goals? What and why?	Open-ended
	Your Curriculum Experience	
S1Q29	I chose an interdisciplinary program instead of a traditional CS Program because I felt the computer science program was out of my reach.	5-Point Likert Scale
S1Q30	I found some elements of my program's curriculum discouraging.	5-Point Likert Scale
S1Q31	The computer science program courses and field choices at my university are overwhelming.	5-Point Likert Scale
S1Q32	The computer science graduation requirements at my university are overwhelming.	5-Point Likert Scale
S1Q33	The names and descriptions of computer science courses in my computer science program are overwhelming/intimidating/confusing.	5-Point Likert Scale
S1Q34	Where did you go to get information about course selection? Please select all that apply.	Multi-Select
S1Q35	Did you take or are you currently taking one or more CS course(s) as a required component for your major or graduation?	Multi-Select
S1Q36	How was your first programming course experience in college?	5-Point Likert Scale

Additional Comments		
S1Q37	What could be done to improve diversity, equity, and inclusion in the Computing Science Curriculum?	Open-ended
S1Q38	Do you have any other comments, questions, or concerns?	Open-ended
About You		
S1Q39	Have you had any programming experience? (You may select multiple answers)	Multi-Select
S1Q40	Do you have any accessibility needs?	Multiple Choice
S1Q41	Do you identify as a racialized person/person of colour?	Multiple-Choice
S1Q42	Please indicate which of the following terms best describe your racial and/or ethnic identity.	Multiple-Choice
S1Q43	Please indicate which of the following terms best describes your gender identity.	Multiple-Choice
S1Q44	To assist us in our review of this survey, please share any comments about the questions or process of this survey with us. We appreciate your feedback as we work to collect accurate information.	Open-ended
S1Q45	If you are interested in providing further feedback through an in-person interview, please provide your preferred contact information here.	Open-ended

Table 33: Survey Questions: Educators

No.	Question	Question Type
	Computer Science Curriculum	
S2Q1	My institution's computer science admission is very competitive.	5-Point Likert Scale
S2Q2	Has your institution, at any point in time, tried to address inequities in computer science participation with students? If Yes, How?	Open-ended
S2Q3	Does your institution promote equity, diversity, inclusion, and justice (EDIJ) efforts (i.e., targeted funding, workshops, etc.)? If so, how?	Open-ended
S2Q4	My institution has gone through significant changes in the undergraduate curriculum to increase diversity in the students population (i.e. changes in core courses, new paths into the degree, etc.).	Open-ended
S2Q5	I believe our undergraduate curriculum could be improved by...	Open-ended
S2Q6	What is the one thing that if done in curriculum design can improve diversity, equity, and inclusion?	Open-ended
S2Q7	What type of support would help you incorporate EDIJ more effectively into your institution's computer science curriculum?	Open-ended
S2Q8	What challenges, if any, have you encountered in incorporating EDIJ principles into your curriculum?	Open-ended
	Your Course Design	
S2Q9	Please state your level of agreement with the following statements: * When designing material for my courses, I am mindful of the students' expected prerequisites * When designing material for my courses, I purposefully choose examples/topics that can appeal to a wide audience * When designing material for my courses, I purposefully include examples/topics that can appeal to underrepresented groups in computer science * In class and in my course material, I try to avoid jargon. * In my class, I prepare students for the next class explicitly (e.g. introduce next course options, discuss what will be covered, define terms, explain how the course might run, etc.)	Matrix
S2Q10	What challenges, if any, have you encountered in incorporating EDIJ principles into your curriculum?	Multiple-Choice
S2Q11	Do you, at any point, explicitly address equity topics in computer science with your students?	Multiple-Choioce
S2Q12	How did you design your course and assessments to address the needs of a diverse set of students?	Open-ended
S2Q13	What type of support would help you incorporate EDI more effectively into your Course?	Open-ended
S2Q14	Do you have any other comments, questions, or concerns?	Open-ended
	About You	
S2Q15	What type of courses do you usually teach?	Open-ended
S2Q16	Do you have any accessibility needs?	Multiple-Choice
S1Q17	Do you identify as a racialized person/person of colour?	Multiple-Choice
S2Q18	Please indicate which of the following terms best describe your racial and/or ethnic identity.	Multiple-Choice
S2Q19	Please indicate which of the following terms best describes your gender identity.	Multiple-Choice
S2Q20	To assist us in our review of this survey, please share any comments about the questions or process of this survey with us. We appreciate your feedback as we work to collect accurate information.	Open-ended
S2Q21	If you are interested in providing further feedback through an in-person interview, please provide your preferred contact information here.	Open-ended

Table 34: Student Interview Questions: Follow up from Survey

No.	Question
I1Q1	In Question X of the survey "Exploring Equity, Diversity, and Inclusion in Computer Science Undergraduate Curricula", you identified Y. Can you please provide more details? For example: How did you realize Y? Why did you find Y? Why do you feel Y? Can you provide more details about Y?
I1Q2	Do you want to share any additional information or experiences about the computer science curriculum with us?
I1Q3	Do you have any additional comments?

Table 35: Interview For Educators & Advisors

No.	Question
I2Q1	How do you feel about the relationship of EDI and the Computer Science Undergraduate Curriculum?
I2Q2	How diverse is the student population in your courses/institution?
I2Q3	Do you incorporate contextual learning into your courses/institution? Why or why not? (if yes, what contexts do you use/how do you select them?)
I2Q4	Do you invite your students to share about themselves? Why or why not?
I2Q5	Have you noticed any situations in your teaching/advising where a student has encountered challenges or obstacles because of being part of a minoritized group? Please elaborate.
I2Q6	Do you have any recommendations about making the Computer Science Undergraduate Curriculum more inclusive?
I2Q7	For multiple Questions (X), follow-up might happen: Can you please provide more details? For example: How did you realize Y? Why did you find Y? Why do you feel Y? Can you provide more details about Y?

Table 36: Course Codes mentioned in Student Survey entries

Course Code	Course Name	University	Course Category
COSC 121	Computer Programming II	UBC	Introductory
DSCI 310	Reproducible and Trustworthy Workflows for Data Science	UBC	Data
CPSC 110	Computation programs and programming	UBC	Introductory
CPSC 121	Models of Computation	UBC	Introductory
CPSC 210	Software Construction	UBC	Core
CPSC 213	Introduction to Computer Systems	UBC	Systems
CPSC 221	Basic Algorithms and Data Structures	UBC	Core
CPSC 303	Numerical Approximation and Discretization	UBC	Theory
CPSC 304	Introduction to Relational Databases	UBC	Data
CPSC 310	Introduction to Software Engineering	UBC	Core
CPSC 312	Functional and Logic Programming	UBC	Core
CPSC 313	Computer Hardware and Operating Systems	UBC	Systems
CPSC 314	Computer Graphics	UBC	Application Oriented
CPSC 317	Internet Computing	UBC	Systems
CPSC 320	Intermediate Algorithm Design and Analysis	UBC	Theory
CPSC 322	Introduction to Artificial Intelligence	UBC	Core
CPSC 330	Applied Machine Learning	UBC	Core
CPSC 340	Machine Learning and Data Mining	UBC	Data
CPSC 344	Introduction to HCI Methods	UBC	HCI
CPSC 406	Computational Optimization	UBC	Theory
CPSC 410	Advanced Software Engineering	UBC	Application Oriented
CPSC 420	Advanced Algorithm Design and Analysis	UBC	Theory
CPSC 421	Introduction to Theory of Computing	UBC	Theory
CPSC 422	Intelligent Systems	UBC	Core
CPSC 427	Video Game Programming	UBC	Application Oriented
CPSC 430	Computers and Society	UBC	HCI
CPSC 436	Topics in Computer Science	UBC	–
CPSC 440	Advanced Machine Learning	UBC	Theory
CPSC 442	Introduction to Cybersecurity	UBC	Systems
CPSC 444	Advanced Methods for Human-Computer Interaction	UBC	HCI
CPSC 447	Introduction to Visualization	UBC	Application Oriented
CPSC 448	Directed Studies in Computer Science	UBC	–
CPSC 455	Applied Industry Practices	UBC	Application Oriented
CMPT 105W	Social Issues and Communication Strategies in Computing Science	SFU	HCI
CMPT 120	Introduction to Computing Science and Programming 1	SFU	Introductory
CMPT 125	Introduction to Computing Science and Programming 2	SFU	Introductory
CMPT 210	Probability and Computing	SFU	Theory
CMPT 213	Object Oriented Programming in Java	SFU	Core
CMPT 225	Data Structures and Programming	SFU	Core
CMPT 272	Client-side Development	SFU	Application Oriented
CMPT 276	Introduction to Software Engineering	SFU	Core
CMPT 295	Introduction to Computer Systems	SFU	Systems
CMPT 300	Operating Systems 1	SFU	Systems
CMPT 305	Computer Simulation and Modeling	SFU	Theory
CMPT 307	Data Structures and Algorithms	SFU	Core
CMPT 310	Introduction to Artificial Intelligence	SFU	Core
CMPT 318	Special Topics in Computing Science	SFU	–
CMPT 320	Social Implications - Computerized Society	SFU	HCI
CMPT 353	Computational Data Science	SFU	Data
CMPT 363	User Interface Design	SFU	HCI
CMPT 365	Multimedia Systems	SFU	Application Oriented
CMPT 371	Data Communications and Networking	SFU	Systems
CMPT 376	Professional Responsibility and Technical Writing	SFU	HCI
CMPT 383	Comparative Programming Languages	SFU	Theory
CMPT 404	Cryptography and Cryptographic Protocols	SFU	Systems

Course Code	Course Name	University	Course Category
CMPT 410	Machine Learning	SFU	Theory
CMPT 419	Special Topics in Artificial Intelligence (Human and Data-Centric AI)	SFU	HCI
CMPT 431	Distributed Systems	SFU	Systems
CMPT 433	Embedded Systems	SFU	Systems
CMPT 454	Database Systems II	SFU	Data + Systems
CMPT 473	Software Testing and Reliability and Security	SFU	Systems
CMPT 475	Requirements Engineering	SFU	Core
CMPT 479	Special Topics in Computing Systems	SFU	Systems
COMP 202	Foundations of Programming (Intro to Python)	McGill	Introductory
COMP 206	Introduction to Software Systems	McGill	Introductory
COMP 208	Computer Programming for Physical Sciences and Engineering	McGill	Introductory
COMP 250	Introduction to Computer Science	McGill	Introductory
COMP 251	Algorithms and Data Structures	McGill	Core
COMP 273	Introduction to Computer Systems	McGill	Systems
COMP 302	Programming Languages and Paradigms	McGill	Core
COMP 303	Software Design	McGill	Core
COMP 330	Theory of Computation	McGill	Theory
COMP 345	From Natural Language to Data Science	McGill	Data
COMP 350	Numeral Computing	McGill	Data
COMP 370	Introduction to Data Science	McGill	Data
COMP 417	Introduction to Robotics and Intelligent Systems	McGill	Application Oriented
COMP 421	Database Systems	McGill	Data
COMP 445	Computational Linguistics	McGill	Theory
COMP 451	Fundamentals of Machine Learning	McGill	Theory
COMP 545	Natural Language Understanding with Deep Learning	McGill	Application Oriented
COMP 550	Natural Language Processing	McGill	Application Oriented
COMP 690	Probabilistic Analysis of Algorithms	McGill	Theory
FIT1006	Business information analysis	Monash	Data
FIT1008	Introduction to computer science	Monash	Introductory
FIT1041	Research Project	Monash	–
FIT1043	Introduction to data science	Monash	Data
FIT1045	Algorithms and programming fundamentals in python	Monash	Core
FIT1047	Introduction to computer systems and networks and security	Monash	Systems
FIT1048	Fundamentals of C++	Monash	Core
FIT1049	IT professional practice	Monash	HCI
FIT1050	Web fundamentals	Monash	Introductory
FIT1053	Algorithms and programming in Python (Advanced)	Monash	Core
FIT1054	Computer science (Advanced)	Monash	Core
FIT1055	IT professional practice and ethics	Monash	HCI
FIT1093	Cybersecurity tools and techniques	Monash	Systems
FIT1095	Cybersecurity tools and techniques	Monash	Systems
FIT2004	Algorithms and data structures	Monash	Core
FIT2014	Theory of computation	Monash	Theory
FIT2099	Object-oriented design and implementation	Monash	Core
FIT2081	Mobile application development	Monash	Application Oriented
FIT2083	Innovation and research in computer science	Monash	Core
FIT2094	Databases	Monash	Data
FIT2102	Programming paradigms	Monash	Core
FIT3139	Computational modeling and simulation	Monash	Theory + Systems
FIT3155	Advanced data structures and algorithms	Monash	Core
FIT3171	Databases	Monash	Data
FIT3179	Data Visualization	Monash	Data
ENG1005	Engineering Mathematics	Monash	Theory
MCD4490	Advanced Mathematics	Monash	Theory

Course Code	Course Name	University	Course Category
COMP1131	Computer Programming 1	UNSW	Introductory
COMP1511	Introduction to Programming	UNSW	Introductory
COMP1521	Computer Systems Fundamentals	UNSW	Systems
COMP1531	Software Engineering Fundamentals	UNSW	Core
COMP1911	Computing 1A	UNSW	Introductory
COMP2041	Software Construction: Techniques and Tools	UNSW	Core
COMP2511	Object-Oriented Design and Programming	UNSW	Core
COMP2521	Data Structures and Algorithms	UNSW	Core
COMP3121	Algorithms and Programming Techniques	UNSW	Core
COMP3311	Database Systems	UNSW	Data
COMP3331	Computer Networks and Applications	UNSW	Systems
COMP3411	Artificial Intelligence	UNSW	Core
COMP3821	Extended Algorithm Design and Analysis	UNSW	Theory
COMP3900	Computer Science Project	UNSW	—
COMP4121	Advanced Algorithms	UNSW	Theory
COMP4141	Theory of Computation	UNSW	Theory
COMP4920	Management and Ethics	UNSW	Introductory + HCI
COMP6080	Web Front End Programming	UNSW	Application Oriented
COMP6841	Extended Security Engineering and Cyber Security	UNSW	Systems
COMP6741	Algorithms for Intractable Problems	UNSW	Theory
COMP6771	Advanced C++ Programming	UNSW	Core
COMP6991	Solving programming problems with Rust	UNSW	Core
COMP9313	Big Data Management	UNSW	Data
COMP9417	Machine Learning and Data Mining	UNSW	Data + Theory
ENGG1811	Computing for Engineers	UNSW	Introductory
BINF2010	Introduction to Bioinformatics	UNSW	Introductory + Application Oriented
INFO1110	Introduction to Programming	University of Sydney	Introductory
INFO1111	Computing 1A Professionalism	University of Sydney	Introductory + HCI
CPSC021	Introduction to Computer Science	Swarthmore College	Introductory
CPSC035	Data Structures and Algorithms	Swarthmore College	Core
CS111	Introduction to Computer Science (Indicated by respondent)	UIC	Introductory
CS 141	Program Design II	UIC	Introductory + Core
CS 151	Mathematical Foundations of Computing	UIC	Theory
CS 31	Introduction to Computer Science I	UCLA	Introductory
COMP 1131	Computer Programming 1	Thompson Rivers	Introductory