The Landscape of Computer Science Education Courses: A Syllabi Analysis

Kathryn Cunningham
Computer Science
University of Illinois
Urbana-Champaign
Urbana, Illinois, USA
katcun@illinois.edu

Miranda C. Parker
Computer Science
San Diego State University
San Diego, California, USA
mcparker@sdsu.edu

Jonathan Zhang
Computer Science + Anthropology
University of Illinois
Urbana-Champaign
Urbana, Illinois, USA
jszhang4@illinois.edu

ABSTRACT

Computer science education, which spans topics related to how people teach and learn computer science, has grown from a niche subject to a core area of focus for many educators, computer scientists, and designers of learning technologies. University-level courses about computer science education are a key method for passing on knowledge about the field to the next generation. However, we know very little about how common such courses are, what topics they emphasize, who teaches them, and who learns from them. To better understand computer science education courses, we evaluated the content and goals of 44 university-level courses that teach topics in computer science education, as well as metadata about where and when these courses are offered and the backgrounds of the 36 instructors that teach them. By understanding the current landscape of computer science education courses, we hope to provide insight into not only how the field is training its next generation, but also where computer science education has space to grow.

CCS CONCEPTS

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

KEYWORDS

computer science education courses, syllabi analysis, course types

ACM Reference Format:
Kathryn Cunningham, Miranda C. Parker, and Jonathan Zhang. 2023. The Landscape of Computer Science Education Courses: A Syllabi Analysis. In *23rd Koli Calling International Conference on Computing Education Research (Koli Calling '23), November 13–18, 2023, Koli, Finland*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3631802.3631831

1 INTRODUCTION

Improving access to and quality of computer science education is a priority in countries around the world [15, 16, 22, 38]. The field of computer science education (CSEd), which covers topics related to the teaching and learning of computer science (CS) [23], has a body of knowledge and practices critical to improving CSEd opportunities. To power growth in the number of people who teach and learn CS, a larger variety of people from diverse contexts need to gain knowledge about CSEd.

However, we have limited understanding about the ways in which knowledge about CSEd is propagated. Specifically, where do people learn topics from the field of CSEd, which content do they learn, and who plays the role of sharing this knowledge with others? CSEd is an innately multidisciplinary subject, influenced not only by CS and education, but also fields like cognitive science, program comprehension, and human-computer interaction [18]. CSEd brings together not only academic traditions with diverse values, but also a variety of professionals with distinct goals for their practice, from instructors to researchers to designers. How are such varied perspectives and objectives passed on to new community members?

In this study, we investigate higher education courses about CSEd as a way to understand the dissemination of CSEd knowledge and practices. Courses have distinct advantages for imparting knowledge about CSEd. As CSEd is a relatively niche and interdisciplinary field, learners are unlikely to have the strong background in both computing and education that would support their ability to self-study CSEd topics. Instead, taking a CSEd course offers a more accessible entry point for learners who may have an interest in the field but little prior knowledge. Additionally, due to the one-to-many structure of courses, where one instructor teaches many students, courses reach more individuals as compared to one-on-one mentorship scenarios like graduate advising, co-teaching, policy discussions, or research collaborations. On top of that, a university course spans several weeks and, as a result, can communicate a relatively large amount of content compared to a conference or workshop. As such, CSEd courses can be used to examine the propagation of knowledge about the field. However, we know little about them. To understand the current landscape of CSEd courses, we ask the following research questions:

- **RQ1:** What are the types of computer science education courses?
- **RQ2:** Who teaches computer science education courses?
- **RQ3:** Who takes computer science education courses?
- **RQ4:** What topics are currently taught in computer science education courses?
- **RQ5:** What learning goals do computer science education courses set for students?

Koli Calling '23, November 13–18, 2023, Koli, Finland

ACM ISBN 979-8-4007-1653-9/23/11…$15.00
https://doi.org/10.1145/3631802.3631831

By surveying instructors, we collected metadata and syllabi for 44 CSEd courses from around the world[1]. We compiled course metadata to understand who teaches CSEd courses, in what departments and institutions, for which student populations, and how often. To understand how courses are taught, we extracted 547 topics and 165 learning objectives from the materials the instructors provided and categorized them. Throughout our analyses, we noticed that the courses in our sample approached the topic of CSEd through distinct lenses. We describe how courses teach about CSEd with the goals of preparing students for research, instruction, or design — or combinations of these.

We contribute an understanding of the landscape of CSEd courses as they are currently taught. We find that this multifaceted field has substantial diversity in styles of courses, course audience, and topics covered. At the same time, there are opportunities to expand CSEd course offerings in new directions that are relatively unsupported in our sample.

We believe our findings will help future instructors to understand the variety of approaches they can take to design a new CSEd course. More broadly, we believe that this snapshot of how CSEd courses are taught provides insight into the current values in our area. By describing the audiences of CSEd courses, we hope to shed light on who may be influential in the field's future.

2 BACKGROUND

2.1 Historical influences on the field of CSEd

Guzdial and du Boulay claim that today, CSEd researchers typically hail from a CS background, resulting in limited influence of other fields like psychology or the learning sciences on CSEd [18]. While CSEd has primarily grown from CS roots, several other fields have impacted and participated in CSEd work. Early CSEd research is often associated with the psychology of programming, when experimental studies investigating programming expertise were first performed by computer scientists in the late 1960s and 1970s [18]. At the same time, early innovators in CSEd were developing new languages like LOGO, with the goal of enabling children to build with code [34]. By the 1980s, CSEd became a topic of interest for cognitive scientists, who brought their own methods of cognitive modeling to the topic of understanding programming, and for educational researchers, who focused on evaluating young students' learning [18].

The growth of CSEd was accelerated by the Bootstrapping workshops that created a community of CSEd researchers from among CS faculty in the US and Europe in the early 2000s [6, 13]. These workshops focused on educational theory and research design. The impact of this community is so strong that Guzdial and du Boulay write "most computing education researchers in the USA were part of one of the capacity-building projects...or are a student of someone who was."

2.2 Barriers to growth of the CSEd community

With the growing importance of CSEd research, the CSEdGrad project was formed to understand where CSEd research takes place and how new CSEd researchers are trained at the PhD level [7]. Among CSEd papers published from 2015-2020, the project found over 500 unique institutions represented from around the world and a growing number of both first-time and unique authors [25]. However, the growth of CSEd research is not without challenges. In interviews with 15 CSEd research faculty, Lunn et al. noted concerns related to identifying the appropriate balance between computing and education in the skill set for a CSEd researcher and finding suitable coursework, often across departments [26]. According to the faculty, CSEd PhD students often preferred to be associated with CS departments due to their prior experience teaching in such a department [26].

In a survey of CSEd researchers, McGill et al. found that the most commonly stated community barriers to participating in CSEd research involved the lack of recognition of and respect for CSEd as a sub-discipline within CS, as well as the lack of knowledge about and adoption of theory and practices from education fields [28]. CSEd courses have the potential to address both barriers. CSEd courses are a place where relevant and contextualized educational theory can be shared with future community members. Establishment of CSEd courses within CS departments may also serve as a signal that CSEd is accepted as an area of study.

As far as the needs of teachers, DeLyser noted in 2016 that short-term professional development has powered substantial growth in K-12 offerings of CS content in the United States, but the approach can result in shallow content depth and a lack of sustainability for building a strong workforce of CS teachers [9]. Similarly, Yadav et al. found that in-service teachers in the US struggle with building CS content knowledge, feeling isolated, and finding professional development opportunities. While Cooper et al. noted that it was difficult for CSEd to find a home within Education departments due to the lack of requirements for CS courses in K-12, such mandates have increased. There is recognition of a growing need for schools of education at colleges and universities to teach CSEd content [8, 10]. However, DeLyser et al. claim that in the context of the United States, few CSEd programs for pre-service teachers exist, and therefore "there are few opportunities for faculty to teach courses in CS education."

2.3 Existing descriptions of CSEd courses

The field of CSEd covers topics related to the teaching and learning of CS [23]. This broad definition includes a variety of sub-areas, including (but not limited to) CSEd research and instructor training.

In a review of publications about undergraduate teaching assistants (TAs) in computer science, Mirza et al. found 20 papers that mentioned a TA training course [30]. These courses included content about learning and teaching styles and techniques; practical responsibilities, like grading and working with difficult students; and communication and professional skills. Graduate students have also been offered coursework in CSEd topics. Minnes describes a course with the goals of guiding Masters and PhD students to apply CSEd research and Scholarship of Teaching and Learning to their work, reflect on their professional development, and effectively carry out the basic roles and responsibilities of a CS TA [29].

In their handbook for those who want to offer CSEd courses for pre-service teachers, Mouza et al. describe skills and attitudes

[1]Course data from consenting instructors is publicly shared at https://go.cs.illinois.edu/CSEd-courses-repository

that CS teachers should obtain, suggest curricular models for pre-service education, and outline some existing courses about CSEd as case studies [31]. Oliveira et al. report on a course for pre-service teachers in Brazil that covers computational thinking content and related instructional approaches [33]. Hazzan et al. have written a textbook designed to support a Methods of Teaching Computer Science course that covers content about how to teach introductory CS topics [19].

Published work on CSEd research courses has focused on topics of research methodology. Berglund et al. describe the challenges of teaching CSEd methodology to CS doctoral students [3], who are used to the "conceptual analysis" approach typical of CS and software engineering rather than the "evaluative" approach common in CSEd [5]. Holz et al. listed relevant research methodologies for the CSEd community, including quantitative, qualitative, and literature review methods, as well as 22 learning objectives that cover understanding and analyzing literature, choosing a methodology, and collecting and processing data. They describe potential approaches to teach this content in courses and other avenues [20].

Only a small number of courses about CSEd have been described so far in the literature, and we do not find any work that contrasts different types of CSEd courses or summarizes the metadata, topics and learning objectives of such courses. In this paper, we extend prior work by describing more CSEd courses, as well as investigating what that data tells us about our field.

2.4 Syllabi Analyses of Other CS Topics

Syllabi analyses are a common strategy to understand the way that computing topics are taught in practice, including methods of instruction, topics covered, and learning goals. Syllabi analyses are often targeted to a particular subject area or set of skills. For example, Becker and Fitzpatrick collected 234 CS1 syllabi from the top 900 universities around the world through online searches [2]. They analyzed these syllabi to find the programming language distribution among CS1 courses and the most common learning objectives in CS1. Abad et al. analyzed 51 distributed systems syllabi, also from an online search for syllabi from the top 100 global universities [1]. They identified the breadth of topics covered in these courses, as well as the most common readings. Fiesler et al. studied 115 syllabi for ethics courses that were posted to a crowdsourced online spreadsheet [12]. They analyzed not only the topics and the learning objectives stated in the syllabi, but also the variety of departmental homes for ethics courses and academic backgrounds of ethics course instructors. To our knowledge, courses about CSEd have never undergone a syllabi analysis. Our goals are similar to these other syllabi analyses in that we hope to describe how CSEd topics are taught currently, and what possibilities this suggests for future CSEd instructors.

3 METHODS

3.1 Survey Creation

Data collection occurred via an online survey, where instructors could submit their syllabus as well as metadata about their course. We asked participants to submit information from courses that met our definition of CSEd courses: *"courses that cover topics related to how people teach and learn computer science."* This definition was adapted from Ko's Computing Education Research FAQ [23].

Similar to Fiesler et al., we collected data about how often each course was taught, the intended audience, who taught the course, and at what department and institution the course was taught. We also created space for participants to submit links or documents of the syllabus, reading list, and additional course materials, if relevant. We also asked participants to share names of other instructors of CSEd courses, which we used for snowball sampling.

After drafting the survey in Qualtrics, we piloted the survey with two faculty who have taught a CSEd course. These experts provided their course information and syllabi, as well as feedback about the survey's usability, use cases, wording, and question formats. We revised the survey according to their feedback.

3.2 Data Collection

We distributed the survey broadly by emailing listservs that focus on CSEd: SIGCSE-MEMBERS@listserv.acm.org and csed-research@u.washington.edu. We sent initial invitation emails to these lists and reminder emails three weeks later. Additionally, the first two authors distributed information about the study and a survey link via social media, with reminders two and four weeks later.

For targeted outreach, we used a snowball sampling method. From our survey respondents, we received 44 recommendations for potential participants. Each individual was contacted and reminded, if they did not fill out the survey within a week. Through this process, we received 13 additional participants (30% of individuals we reached out to at this stage), as well as 8 individuals (18% of snowball suggestions) who directly declined to participate because they believed that their courses did not qualify under our definition.

Building on prior work, we reached out directly to 13 instructors who submitted a CSEd course syllabus to the CSEdGrad project [7]. Six of the instructors had already submitted to our survey, and we received permission to use data from an additional two instructors.

3.3 Data Cleaning

Each of the first two authors individually evaluated the 56 submitted syllabi to determine if they met our definition of a CSEd course (see Section 3.1). For example, if a course only focused on computer science content, but did not cover topics about the teaching the learning of computer science, it was eliminated. If a syllabus did not provide enough information, then the reading list and/or other course materials provided were used to determine suitability of the course. If a syllabus was in another language (4 syllabi), we used Google Translate to translate the syllabus into English for analysis. Google Translate was deemed acceptable as the exact wording of the syllabi was not critical, but rather informed the next step of data analysis; no words or phrases were encountered in the translated versions that seemed incorrectly translated. Both authors met to compare the suitability of each syllabus for inclusion into the analysis, discussing and resolving conflicts.

In three cases, the same course was submitted multiple times by different instructors or by the same instructor across different offerings. We kept the more comprehensive response or, when course information was equivalent, the most recent offering. An additional two courses were similar to other courses submitted (i.e.,

same instructor, institution, and syllabus) but were intended for different audiences. In these cases, we kept the courses separate for the summative statistics portion of our analysis but removed the duplicates for the topic and learning objective analysis portion.

3.4 Data Analysis

We extracted the topics and learning objectives from 44 syllabi that met our criteria. Topics were extracted from headers of the course schedule or an explicit listing of course topics. Learning objectives were extracted from explicitly stated lists of learning objectives. 39 syllabi had course topics, and 27 syllabi had learning objectives.

After all topics were extracted, the first two authors collaboratively used a constant comparative method, which allows researchers to code and analyze qualitative data simultaneously [17]. First, the authors applied categories to each topic. Then, each category was discussed and considered to ensure consistency among topics and to formally define the boundaries of each category. When the meaning of a topic was unclear, we looked at the topic in the context of the rest of the syllabus to help us understand the intent and purpose of the topic. The same method was used for the learning objectives. In both analyses, we found that several extracted pieces of data actually contained multiple topics or objectives; we separated out each distinct unit for our analysis. Any outliers were discussed and the authors came to an agreement on all topics. After these were separated, there were 547 total topics, and 165 total learning objectives.

During our analyses, we observed that the CSEd courses often focused on a specific area: instruction, research, or design. After discussing the definitions and applicability of these categories, the first two authors coded each course individually for its primary type, resulting in a Cohen's kappa of 0.89, indicating strong agreement. We discussed discrepancies to reach final categorization.

4 LENSES ON COMPUTER SCIENCE EDUCATION

After an initial analysis of our collected syllabi, it became clear that our broad definition of CSEd courses (*"courses that cover topics related to how people teach and learn computer science"*) resulted in course submissions that prioritized differing aspects of the wide-ranging field of CSEd. Based on the variety of course types we observed, and the fact that most courses tended to take one of a few forms, we categorized CSEd courses into broad types. This allows us to present our analysis both across all CSEd courses and in terms of particular course styles.

To categorize CSEd courses, we present a set of focii that CSEd courses have in our data. These three categories describe much of the variation in the CSEd courses that we collected:

- **Instruction-focused courses**: focus on techniques for implementing teaching of computer science in the classroom, particularly practical strategies. Build skills in enacting classroom practice. Emphasize adaptation and evaluation of existing resources. Often intended for TAs or K-12 teachers.
- **Research-focused courses**: focus on understanding past CSEd research or building skills to create new CSEd research. Emphasize understanding or increasing the evidence base for models of CS learning or strategies for teaching.

Table 1: Primary lenses of CSEd courses in our sample

Primary Lens	# Courses	# Institutions
Instruction	24	15
Teacher audience	*18*	*11*
TA audience	*6*	*5*
Research	18	15
Design	2	1
Total	44	29

- **Design-focused courses**: focus on creating new tools that support learning of computer science. Foregrounds the development of arguments for why certain designs meet key goals.

These three categories of instruction, research, and design provide distinct lenses to view knowledge about CSEd topics. Consider an instructional strategy like Parsons problems [11], for example. Students in a CSEd course can learn how to implement it in the classroom (*instructional lens*), study its effectiveness for learning or motivation (*research lens*), or design new features to support learning outcomes (*design lens*).

Below, we describe course exemplars (using course name pseudonyms) that show the breadth of courses in our sample and further describe how the CSEd lenses are enacted in course structures.

4.1 CSEd courses with an instructional lens

4.1.1 For teachers. CSED36 is a course where students "investigate topics central to computer science for the pk-12 learner", offered by an Education department with an intended audience of masters students. Course goals include not only building "understanding and skills" of CS concepts and developing problem-based lessons, but also describing relevant CSEd research findings and engaging with professional CSEd organizations. Students create and analyze CS lessons for the majority of their assignments. Reading materials consist of research articles about math and data science education as well as descriptions of constructionist learning approaches. This course has a primary goal of preparing students for instruction, but it utilizes educational theory and research findings to broaden future teachers' perspectives.

4.1.2 For teaching assistants. CSED10 is a required course for undergraduate students who are interested in becoming a paid tutor for lower-level CS courses and is housed in a CS department. The course includes weekly lectures about best practices and current research in CS pedagogy, and expects students to apply what they have learned during hands-on, guided tutoring experiences. Students are tasked with reading scholarly articles on CSEd and education generally, writing reflection papers, shadowing other CS tutors, and participating in mock tutoring scenarios. The objective of this course is instruction-focused: namely, to teach students how to teach CS. The largest single component of the grading scheme is their tutoring performance in the course they TA for. However, the course also supports some growth in introductory research skills. In one assignment, students receive advice about how to read research papers and are expected to analyze two CSEd research papers.

4.2 CSEd course with a research lens

CSED34 is aimed at masters and doctoral students. The course is offered by the CS department, and students' objectives are to "evaluate pedagogical interventions/learning technologies" and either "conduct learning research" or "develop learning technologies." Course activities include writing reflections on research articles and leading class discussions, as well as authentic research tasks like undergoing human subjects training and creating an annotated bibliography. In their final project, students have the choice to either propose learning research, design a learning technology, or develop an educational technology prototype. While students can pursue personal design interests and some course topics touch on instructional strategies (e.g., "effect of first programming language choice"), this course prioritizes research knowledge and skills in its activities, as well as learning theory in its course topics.

4.3 CSEd course with a design lens

CSED25 is offered to undergraduate, masters, and doctoral students who are interested in examining the roles of computational tools in designing teaching and learning environments. The course is cross-listed in both education and computer science departments. The course covers topics of equity and inclusion, pedagogy, creative applications of computing, and the integration of computational ideas across multiple disciplines. Students are tasked with weekly readings, discussions, journaling, and reaction papers. Ultimately, the course culminates with the creation of a "learning design", where students incorporate an idea from computing into the design of a learning experience. Although the course is primarily focused on the design of learning environments using computational tools, research is also a key component. In fact, students are expected to interview a computing researcher and integrate their findings into their learning design proposal.

5 WHO TEACHES AND TAKES CSED COURSES?

To answer our second and third research questions, we used the metadata collected via our survey to understand who teaches CSEd courses, what departments these courses and instructors belong to, what level of student the course is designed for, and how often these courses are taught.

5.1 Who teaches CSEd courses?

5.1.1 What are the home departments of CSEd course instructors? CSEd courses in our sample were primarily taught by professors affiliated with CS departments (61%). The next most common department affiliation was Education (17%). Instructors sometimes had a joint affiliation in both a CS and Education department, or were affiliated with an interdisciplinary department like Information or Engineering Education, but this was much less common in our sample (see Table 2).

5.1.2 What is the educational background of CSEd course instructors? Nearly all the course instructors surveyed had a doctoral degree, with only 2 of 36 having a Master's degree. Two-thirds of CSEd course instructors in our sample attained a CS degree as their terminal degree (see Table 2). The next most common educational background was in Education (19%). Two instructors had backgrounds in Human-Computer Interaction, and one instructor had a PhD in Math.

CSEd is a new field, so one might assume most instructors of CSEd courses were never formally trained in CSEd themselves. However, in our data we found that among the 34 course instructors with a PhD, 47% wrote their dissertation on the topic of CSEd (Cohen's kappa of 0.88, indicating very high agreement between coders). Among those 16 instructors with a PhD thesis about CSEd, 9 (56%) had a PhD in Computer Science, 5 (31%) had a PhD in Education, one had a joint PhD in Computer Science and Education, and one had a PhD in HCI (Human-Centered Computing).

Table 2: Departments that offer CSEd courses, house CSEd course instructors, and provide the terminal degree of CSEd course instructors. Some instructors have offered multiple CSEd courses.

Department	Course home	Instructor home	Instructor degree
Computer Science	23	22	24
Education	13	6	7
Joint CS and Education	2	3	1
Information / HCI	1	3	2
Engineering Education	1	1	0
Science	1	1	0
Mathematics	0	0	1
N/A	3	0	0
Total	44	36	36

Table 3: Carnegie Classifications of U.S. Institutions Offering CSEd Courses

Carnegie Classification	#
R1	19
M1	3
R2	1
Baccalaureate Colleges: Arts & Sciences Focus	1

5.2 Where are CSEd courses taught?

5.2.1 Which countries teach CSEd courses? The majority of courses in our sample are taught at institutions located in the US (82%). The rest are located in Canada, the United Kingdom, New Zealand, Sweden, Germany, and Brazil. As expected for a sample largely from the English-speaking world, the most common language of instruction is English (93%), followed by two courses taught in Portuguese, and one course taught in German. The large variability in location of the CSEd courses in our sample suggest that CS Ed courses are not restricted to a single geographical location.

5.2.2 What types of institutions teach CSEd courses? Among courses taught in the United States, CSEd courses in our sample are overwhelmingly taught at institutions with very high research activity (see Table 3). Almost 80% of the American institutions in our data set have an R1 Carnegie classification [32]. The rest of the institutions were M1 Carnegie classification, meaning that they offer Masters degrees but not PhDs, as well as one R2 institution and one

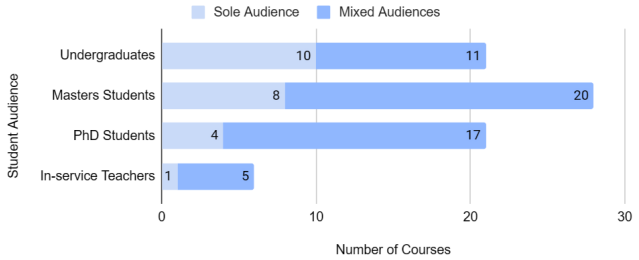

Figure 1: The intended audiences for the CSEd courses.

Table 4: How many times the CSEd course has been taught and how frequently is it offered.

Times taught	#	Frequency	#
1	14	Continuously	4
2-3	13	2-3 times a year	8
4-6	5	Once a year	19
7 or more	8	Once every other year	5
		Sporadically or one-off	8

Baccalaureate Arts and Sciences institution. It appears that CSEd courses can be taught at a variety of institution types, although they seem to have the strongest presence at large research-focused universities.

5.2.3 What departments teach CSEd courses? CSEd courses in our sample were primarily housed in CS departments (52%). The next most common department affiliation was Education (30%). Courses were also sometimes jointly offered between CS and Education. Interdisciplinary departments like Information and Engineering Education only housed a couple of the courses in our sample (see Table 2). Note that some instructors taught multiple courses (we saw as many as six per one (busy) instructor). This leads to the higher number of courses in our data than instructors.

5.3 Who takes CSEd courses?

The anticipated student audience for a course can influence its design. We found that CSEd courses were intended for a variety of degree levels, and frequently there were multiple anticipated audiences for the same course (see Figure 1). CSEd courses in our sample were most often intended for masters students (64%), and were also frequently intended for undergraduates (47%) and PhD students (47%). In-service teachers represented a smaller slice of the audience for CSEd courses (14%).

Courses were likely to have multiple audiences. 23 courses had a single audience (most often undergraduates), but the rest of the courses had more than one intended audience. Many courses in our sample listed PhD students as an intended audience, however, only 9% of courses designed only for PhD students. Only one course in our sample was solely designed for in-service teachers.

5.4 When are CSEd courses taught?

5.4.1 How often are CSEd courses offered? As CSEd is still a relatively niche topic, we might expect CSEd courses to be offered infrequently. However, the majority of courses in our sample are offered at least once per year (70%), and about a quarter are offered 2-3 times a year or continuously (see Table 4). This suggests that many CSEd courses are established in their departments, and a part of the regular teaching schedule. At the same time, 27 (61%) have only been taught once or two to three times, indicating that they are relatively new courses. Thirteen courses have been offered more than four times.

5.4.2 When were the courses in our sample offered? We found that over half of the syllabi we collected were from courses offered in the most recent 1.5 years before data collection (17 courses in Spring 2022, 12 in Spring or Fall 2021). We also looked online to check whether any of these courses were offered again in the Fall semester after our data collection. With this data, we found that 29 of the 44 courses in our sample were offered in Spring or Fall 2022. The fact that the majority of these courses were taught in the most recent year shows that there is a healthy demand for CSEd course content.

6 WHAT IS TAUGHT IN CSED COURSES? (COURSE TOPICS)

From nearly 550 topics across 39 courses, seven themes were identified. These are shown in Table 5 and described below.

Out of the 39 courses that included topics on their syllabi, 29 of the courses included topics on **instructional methods**. Approaches and techniques for teaching were the most common topics among all CSEd courses. This included general statements, such as "pedagogy" or "inclusive teaching methods", as well as techniques specifically for teaching CS, such as "live coding" and "pair programming", and approaches not specific to a subject, such as "active learning" and "peer instruction". A number of syllabi mentioned integrating computing into other subjects.

Also quite common were topics about **classroom and teacher practices**, which 26 courses included in their syllabi. Rather than being focused on specific teaching techniques, this category focused more on topics about being a teacher and the day-to-day activities that align with that reality. The most common subtopics were assessment (e.g., "formative and summative assessment"), as well as specific curricular elements, such as "what makes a good lesson" or "programming projects". This category also included topics related to other aspects of the classroom, like "classroom management", "creating reusable resources", and "communication and leadership".

Most of the CSEd course syllabi mentioned **theories of thinking and learning**. These topics include general theories from learning sciences, cognitive science, and educational psychology, such as "cognitive load", "threshold concepts", "transfer of learning", and "funds of knowledge". There were CS-specific learning theories as well. These included topics like "notional machines", "student understanding: recursion", and "programmer cognition". In this category, we also included topics focused around computational thinking and computational literacy, which attempt to define *how* people think when they think about computing and CS. This category also includes topics on affective factors, such as self-efficacy and motivation, which often impact cognition and learning.

Just over half of the courses (54%) included topics we categorized as **diversity, equity, and inclusion (DEI)**. This is an undercount

Table 5: Themes of topics referenced in CSEd course syllabi.

	# Courses	# Topics
Instructional Methods	**29**	**95**
General Approaches	19	32
Computing-Specific Approaches	17	33
Pedagogy	10	13
Integrating Computing	7	8
Inclusive Teaching Methods	5	6
Gamification	3	3
Classroom and Teacher Practices	**26**	**103**
Assessment	16	19
Curricular Elements	12	20
Professional Development	9	20
Classroom Operation	8	13
Grading	6	10
Resources	4	7
Standards	4	4
Plagiarism	4	4
Current K-12 Overview	3	3
Office Hours	3	3
Theories of Thinking and Learning	**26**	**87**
Learning Science and Cognition	18	47
CS-Specific Cognition	13	19
Computational Thinking and Literacy	11	15
Affective Factors	6	6
Diversity, Equity, and Inclusion	**21**	**38**
Equity Writ Large	11	11
Access, Recruitment, and Retention	5	5
Justice and Power	4	5
Race and Gender	4	8
Definitions and Theories	4	4
CS for All	2	3
Why Teach or Learn CS?	2	2
CSEd Research	**18**	**112**
CSEd Research Overview	14	24
Methods for Data Collection and Analysis	11	30
Research Skills	11	17
History of CSEd Research	8	8
K-12 Research Overview	7	10
Learning Analytics	6	6
CS1 Research Overview	6	7
Research Communities in CSEd Research	3	5
Informal CS Research Overview	3	3
The Future of CSEd Research	2	3
Technological Applications	**15**	**33**
Tools	9	13
Programming Languages	8	10
Learning Environments	3	5
Design	1	5
Computing Content	**9**	**79**
Fundamental Programming Topics	7	41
Other Computing Topics	6	24
Impacts of CS on society	2	14
Total	**39**	**547**

of all DEI-related topics: when a topic was described as an equitable approach to another topic, we clustered it with that topic (e.g. "equitable grading" is counted in Classroom and Teacher Practices). However, several topics did not tie to another category. Many were

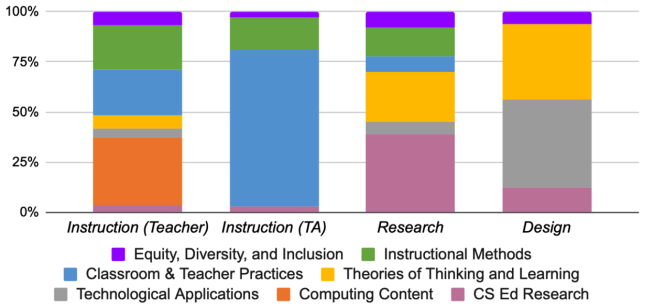

Figure 2: Percentage of topics in each category by course type.

general terms of "equity" or "inclusion." Others included a focus on representation of students and the field, and trying to reach, teach, and recruit diverse populations. There were also four courses with topics that specifically related concepts of justice and power to DEI.

Eighteen courses included topics related to varying aspects of **CSEd Research**. While this category did not have the most courses represented, it did include the most topics within it (n=112). The CSEd research category included an overview of the field of CSEd research, as well as topics on the history of the field and future directions. Overviews of existing research were the most popular sub-category, including topics about CSEd Research overall ("research in learning computer science", "What is computer science education research?"), as well as a number of sub-areas of CSEd research, such as overviews of research on introductory CS, K-12 CS, and informal (outside of school) CS learning. The remaining subtopics in this category include practical aspects of performing CSEd research, including methodological skills and familiarity with research communities.

Some topics focused on **technological applications** for teaching and learning of CS. This category included topics on tools (e.g., "computational tools for creativity", "coding tools"), as well as learning environments (e.g., "learning with online simulators") and programming languages (e.g., "benefits of block-based languages", "programming paradigms"). One course also included explicit topics on design, moving from understanding tools to building them.

Although only mentioned in nine courses, there were 79 topics related to **computing content**. These included fundamental programming topics such as "sequences", "loops", "algorithms", and "debugging", as well as more advanced topics in cybersecurity, web design, artificial intelligence, and beyond. Two courses included topics specifically focused on the societal impact of CS.

6.1 Topics by CSEd course type

The distribution of topics among each type of course reveals that certain course types tend to favor certain course topics (see Figure 2). For **design-focused courses**, the majority of topics focus on Technological Applications. However, they also cover topics related to Theories of Thinking and Learning, as well as CSEd Research, indicating a tight relationship between design and research. However, this relationship does not seem to go in the other direction. In **research-focused courses**, CSEd Research and Theories of Thinking and Learning make up the majority of topics. Technological Applications topics are present but not well-represented.

Table 6: Themes of learning objectives (LOs) in CSEd course syllabi.

	# Courses	# LOs
Teaching	**18**	**75**
Design a lesson / curriculum / assignment	11	22
Teach / Apply knowledge to teaching practice	9	11
Understand teaching practices / pedagogy	8	14
Use technological tools for instruction	8	8
Assess CS learning	5	5
Engage with the CSEd community	3	3
Understand and/or utilize standards	3	3
Know the job of being a teacher	2	4
Explore and evaluate outside resources	2	3
Integrate CS into other disciplines	1	1
Change their own affect or values	1	1
CSEd Research	**11**	**39**
Understand current state of the field	10	15
Critique and evaluate CSEd research	7	8
Perform research tasks	7	12
Understand research processes	4	4
Computing Content	**11**	**19**
Understand CS content knowledge	9	11
Understand the impact of CS on society	4	8
Professional Skills	**7**	**23**
Communicate / Present	7	11
Collaborate / Build interpersonal skills	3	5
Prioritize / Organize	2	2
Reflect on working process	2	2
Reflect on career goals	1	2
Maintain student privacy	1	1
Equity and Justice	**3**	**4**
Examine CSEd for biases	2	3
Plan equitable recruitment	1	1
Learning Technologies	**1**	**5**
Critique / Justify a technology for learning	1	3
Design / Develop a learning technology	1	2
Total	**27**	**165**

In **instruction-focused courses** aimed at **TAs**, Classroom and Teacher Practices and Instructional Methods dominate, indicating a very practical focus. In **instruction-focused courses** aimed at **teachers**, CS Content makes up a large percentage of topics, followed by Instructional Methods and Classroom and Teacher Practices. Both types of instruction-focused courses have few topics related to CSEd Research or Technological Applications. For all course types, Equity, Diversity, and Inclusion topics are fairly equally represented, albeit at a small percentage of overall topics.

7 WHY ARE CSED COURSES TAUGHT? (LEARNING OBJECTIVES)

The result of our analysis of 165 learning objectives from 27 courses can be seen in Table 6, and is described below.

Learning objectives about **teaching** CS were wide-ranging, from preparing materials and assessments to performing teaching to interacting with the CSEd community, resources, and standards. Within this category, learning objectives focused on the practice of teaching, such as designing a lesson or assignment, assessing learning, or using tools for instruction, were most common. However, learning objectives about the theory of teaching were also present; for example, one syllabus included: "describe fundamental concepts concerning how students learn computer science".

CSEd research learning objectives ranged from understanding the state of research, to evaluating others' research, to understanding the research process in general, to performing research activities oneself. This progression aligns with Bloom's Taxonomy [24], presenting a roadmap for CSEd researchers in-training.

Although the courses we analyzed all qualified as CSEd courses, 11 of them also included learning outcomes of understanding **computing content**. Most often, this content was traditional CS concepts, like "read and write computer programs". However, other courses explicitly listed a goal of having their students understand the effect of computing on society. Such CSEd courses present a jumping-off point for ethics discussions.

One category focused on **professional skills**, including collaboration, time management, reflection on ones' own ability and goals, and ability to manage data privacy. These objectives imply that some course instructors feel that CSEd is fertile ground for more general personal development.

Some courses included learning objectives around **equity and justice**, intending to counteract inequitable outcomes outside of typical teaching practices. This included the ability to identify potential biases in curricular materials or classroom policy, as well as the ability to run an equitable recruitment strategy for a CS class. These are not the only learning objectives related to equity, diversity, or inclusion. Many other learning objectives included mention of equitable strategies in the context of other outcomes, e.g., "Apply and evaluate the usage of equitable, evidence-based approaches to teaching CS" (counted in Teaching).

Only one course included learning objectives explicitly about **learning technologies**, but this course had five such objectives. These included critiquing a technology for learning, but also designing the learning technology. Although this presence is small in our sample of syllabi, we believe it represents an important intersection of CSEd with design and development of technology.

8 DISCUSSION

With the above understanding of the current landscape of CSEd courses, we can make some inferences about trends in our field and opportunities for further growth of CSEd training.

8.1 CSEd courses aren't rare, but they have room to grow

There is a narrative in our community that courses focused on CSEd are scarce. However, our survey found many CSEd courses offered at a variety of institutions. Our findings suggest that CSEd courses are not uncommon, and in fact have been offered in at least 29 colleges and universities. CSEd courses in our sample were much more likely to be offered at institutions with high research activity. As CSEd is a specialized topic, it's understandable that it may be most well-supported at well-funded research universities with many faculty. However, our results also show that CSEd courses can be

supported at non-PhD granting institutions, given that we saw four such examples.

CSEd courses were also offered with more frequency than we expected. About half of the CSEd courses in our sample were offered once a year or more, and two-thirds had been offered more than once at the time of data collection. This frequency of offerings shows that CSEd courses have made inroads as standard courses, but also indicates that CSEd courses are not always core or foundational part of departmental course offerings. This data shows promising signs for the establishment of CSEd courses in higher education, but still indicates room for growth.

8.2 CSEd courses are most well-established in CS and Education departments

While the field of CSEd has diverse influences (Section 2.1), our data suggests that CS is the dominant environment that houses, offers, and trains the instructors of CSEd courses. Education departments also have a significant presence, offering about a third of the courses in our sample. Our results show there is a healthy CS to CSEd pipeline that is generating CSEd instructors, alongside a smaller Education to CSEd pipeline.

While CS and Education are highly represented, other departments have a small showing in our data. iSchools, HCI departments, and Engineering Education departments are arguably even more natural homes than CS for a CSEd course, as these units are already interdisciplinary across computing and social sciences [10]. However, despite over 125 iSchools [21] and over 50 Engineering/STEM Education graduate programs [4] globally, these departments housed only two courses and four instructors in our sample.

Our analysis finds signs that CSEd may be increasingly "homegrown." The fact that nearly half of CSEd course instructors wrote a thesis about CSEd suggests that obtaining a doctoral degree focused on CSEd is predictive of teaching a CSEd course in the future. Looking ahead, as we have more community members that receive training in CSEd early in their career and who go on to teach CSEd courses, the influence of outside fields on CSEd may decrease.

8.3 CSEd courses take diverse forms with diverse audiences

Our analysis finds that there are a variety of potential entry points for students at institutions of higher education to learn CSEd content, across all student levels. Undergraduates may access CSEd content as TAs for CS courses, in a service-learning course, or in a course about CSEd research or design (often alongside graduate students). Masters students were among the intended audience for the largest number of courses in our sample (Figure 1). These students can access CSEd content in teacher preparation programs (alongside in-service teachers), TA training in CS departments, as well as graduate-level CSEd research or design courses. PhD students also may see CSEd content in TA training courses and graduate-level CSEd research or design courses.

The variety of CSEd course objectives and topics we found show that CSEd courses can play "double duty" with non-CSEd material. For example, TA training courses may teach CSEd content along with professional development goals like improved communication; or, a teacher preparation course may teach computing content as well as methods for teaching that content. A CSEd research course can teach general research methods alongside CSEd-specific findings and theories, while a design-focused course can teach general prototyping and design argumentation strategies in the context of building CSEd tools. These varied course models provide multiple avenues for instructors to make the case for a CSEd course in less supportive environments.

8.4 CSEd is addressing barriers to its growth

We believe our findings speak to recent concerns about the status of the field of CSEd (Section 2.2). Our analysis contributes evidence for the legitimacy of the field of CSEd within CS by showing that at least 24 CS departments have offered a CSEd course, including 19 at R1 institutions, reaching both undergraduate and graduate audiences. Despite concerns that CSEd courses could not be supported in Education departments [6, 10], our survey found 13 such courses, as well as two courses jointly offered by CS and Education.

While many CSEd community members are concerned about the lack of theory and practices from other education research fields in CSEd [28], we found 18 CSEd courses that cover topics about Learning Science and Cognition (Table 5). As far as educational research methodologies, we found that qualitative research methods were listed as topics in four courses. Critical theory made a showing in our sample, as we found topics about power imbalances and objectives about critical analysis in multiple courses, including both research-focused and instruction-focused courses.

8.5 Gaps in topic areas and goals suggest opportunities for novel courses

Our descriptions of what CSEd courses can and do cover reveal material that is missing. For instance, among topics and approaches that CSEd journal editors were interested in seeing more of in CSEd research [14], we found not all are being taught to the next generation in the courses in our sample. We did not find any topics related to learning progressions in our sample. For a number of other topics of interest, such as adult learners of CS, ethnographic methods, and embodied cognition, we could only find one course that covered each topic.

At the level of entire courses, we find that the amount of design- and tool-focused content in our sample of CS courses is much smaller than that of our other categories. This paucity of design-focused CSEd courses is particularly surprising considering that "computing educators have been highly active in developing and evaluating tools to support learning," according to a recent summary of tools and environments in CSEd research [27]. However, this lack of focus on tool design in CSEd courses may mirror the decreased focus on tools in the CSEd research community. While various surveys from the 2000s found that 16-26% of CSEd publications were primarily about a tool [35–37, 39], this focus on tools has decreased over time, according to recent editors of CSEd journals [14]. In addition, those editors felt that many tools-focused papers did not meet standards of the field as far as evaluation of effectiveness.

Design-focused CSEd courses present an opportunity to demonstrate the overlap between CSEd and HCI. As CSEd attempts to prove its legitimacy as a domain within CS, it may be a mistake

to forsake a stronger association with HCI, a field that is already accepted in many CS departments.

9 LIMITATIONS

As with any survey, there is a risk of bias among our sample of respondents. While our data illustrates significant variation among CSEd courses and represents several countries, we cannot claim that our sample is fully representative. It is possible that recruiting through the mailing lists we chose led to sampling from a pool that is more likely to be in CS and in the United States. We also cannot make any claims about what proportion of all CSEd courses our sample represents.

Additionally, during data collection, we found that the norms of syllabi design vary by course, as well as across international contexts. This may have resulted in our analyses being over- or under-informed about topics, learning objectives, and other course elements from courses where this information is not included in the syllabus.

10 CONCLUSION

Our analysis provides a snapshot of current CSEd courses: who teaches them, who takes them, and what they contain. Supporters of CSEd courses have reason to be positive. These courses are offered to students across many levels, from undergraduates to graduate students to in-service teachers, resulting in many entry points into the field. The number of syllabi we collected was fairly high, suggesting that CSEd courses may be more frequent than commonly expected.

Our mapping also identifies gaps. Computer Science and Education departments host the majority of CSEd courses, with surprisingly low representation from Engineering Education and Information departments. The topic of tool design for CSEd goals was remarkably under-represented in our sample, suggesting opportunities for future instructors.

Our analysis of 44 diverse CSEd courses provides a menu of course features that future course designers can choose from. While a number of diverse CSEd courses already exist, many more have yet to be created, particularly at the intersections of the lenses of research, design, and instruction. By understanding where we currently stand, we can better plan for the future of CSEd training across all levels.

ACKNOWLEDGMENTS

Thank you to our study participants for sharing their course information with us. Many thanks to the CSEdGrad project (Stephanie Lunn, Maíra Marques Samary, and Alan Peterfreund) for their provision of additional syllabi and contacts.

REFERENCES

[1] Cristina L. Abad, Eduardo Ortiz-Holguin, and Edwin F. Boza. 2021. Have We Reached Consensus? An Analysis of Distributed Systems Syllabi. 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, 1082–1088. https://doi.org/10.1145/3408877.3432409

[2] Brett A. Becker and Thomas Fitzpatrick. 2019. What Do CS1 Syllabi Reveal About Our Expectations of Introductory Programming Students?. 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, 1011–1017. https://doi.org/10.1145/3287324.3287485

[3] Anders Berglund, Päivi Kinnunen, and Lauri Malmi. 2007. A Doctoral Course in Research Methods in Computing Education Research How Should We Teach It?. In *Proceedings of the Seventh Baltic Sea Conference on Computing Education Research - Volume 88* (Koli National Park, Finland) *(Koli Calling '07)*. Australian Computer Society, Inc., AUS, 175–178.

[4] Adam Carberry and Ken Yasuhara. 2023. Engineering Education Departments and Programs. http://engineeringeducationlist.pbworks.com/w/page/27610307/Engineering%20Education%20Departments%20and%20Programs%20%28Graduate%29.

[5] Tony Clear. 2013. Doctoral Work in Computing Education Research: Beyond Experimental Designs. *ACM Inroads* 4, 2 (jun 2013), 28–30. https://doi.org/10.1145/2465085.2465092

[6] Steve Cooper, Shuchi Grover, Mark Guzdial, and Beth Simon. 2014. A Future for Computing Education Research. *Commun. ACM* 57, 11 (oct 2014), 34–36. https://doi.org/10.1145/2668899

[7] CSEdGrad.org. 2021. CSEdGrad: Exploring Pathways of Future CSEd Researchers. https://www.csedgrad.org/.

[8] CSforEd. 2017. Finding a Home for Computing Education in Schools of Education Strategy Workshop. https://www.csfored.org/workshop2017.

[9] Leigh Ann DeLyser. 2016. Building a Computer Science Teacher Pipeline for New York City A Gathering of Teacher Preparation Programs. https://www.csfored.org/report2016.

[10] Leigh Ann DeLyser, Joanna Goode, Mark Guzdial, Yasmin Kafai, and Aman Yadav. 2018. Priming the Computer Science Teacher Pump: Integrating Computer Science Education into Schools of Education. https://www.csfored.org/report2018

[11] Barbara J. Ericson, Lauren E. Margulieux, and Jochen Rick. 2017. Solving Parsons Problems versus Fixing and Writing Code. In *Proceedings of the 17th Koli Calling International Conference on Computing Education Research* (Koli, Finland) *(Koli Calling '17)*. Association for Computing Machinery, New York, NY, USA, 20–29. https://doi.org/10.1145/3141880.3141895

[12] Casey Fiesler, Natalie Garrett, and Nathan Beard. 2020. What Do We Teach When We Teach Tech Ethics? A Syllabi Analysis. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (Portland, OR, USA) *(SIGCSE '20)*. Association for Computing Machinery, New York, NY, USA, 289–295. https://doi.org/10.1145/3328778.3366825

[13] Sally Fincher and Josh Tenenberg. 2006. Using Theory to Inform Capacity-Building: Bootstrapping Communities of Practice in Computer Science Education Research. *Journal of Engineering Education* 95, 4 (2006), 265–277. https://doi.org/10.1002/j.2168-9830.2006.tb00902.x

[14] Sally A. Fincher, Josh Tenenberg, Brian Dorn, Christopher Hundhausen, Robert McCartney, and Laurie Murphy. 2019. Computing Education Research Today. In *The Cambridge Handbook of Computing Education Research*, Sally A. Fincher and Anthony V.Editors Robins (Eds.). Cambridge University Press, 40–55. https://doi.org/10.1017/9781108654555.003

[15] Brian Fowler and Emiliana Vegas. 2021. How Chile Implemented Its Computer Science Education Program. *Center for Universal Education at The Brookings Institution* (2021). https://www.brookings.edu/articles/how-chile-implemented-its-computer-science-program/

[16] Brian Fowler and Emiliana Vegas. 2021. How England Implemented Its Computer Science Education Program. *Center for Universal Education at The Brookings Institution* (2021). https://www.brookings.edu/articles/how-england-implemented-its-computer-science-education-program/

[17] Barney G. Glaser. 2014. The Constant Comparative Method of Qualitative Analysis*. *Social Problems* 12, 4 (08 2014), 436–445. https://doi.org/10.2307/798843

[18] Mark Guzdial and Benedict du Boulay. 2019. The History of Computing Education Research. In *The Cambridge Handbook of Computing Education Research*, Sally A. Fincher and Anthony V.Editors Robins (Eds.). Cambridge University Press, 11–39. https://doi.org/10.1017/9781108654555.002

[19] Orit Hazzan, Noa Ragonis, and Tami Lapidot. 2020. *Design of Methods of Teaching Computer Science Courses*. Springer International Publishing, Cham, 321–348. https://doi.org/10.1007/978-3-030-39360-1_15

[20] Hilary J. Holz, Anne Applin, Bruria Haberman, Donald Joyce, Helen Purchase, and Catherine Reed. 2006. Research Methods in Computing: What Are They, and How Should We Teach Them?. In *Working Group Reports on ITiCSE on Innovation and Technology in Computer Science Education* (Bologna, Italy) *(ITiCSE-WGR '06)*. Association for Computing Machinery, New York, NY, USA, 96–114. https://doi.org/10.1145/1189215.1189180

[21] The iSchools Organization. 2023. Members Database. https://www.ischools.org/members.

[22] Susumu Kanemune, Shizuka Shirai, and Seiichi Tani. 2017. Informatics and programming education at primary and secondary schools in Japan. *Olympiads in Informatics* 11, 1 (2017), 143–150. https://ioinformatics.org/journal/v11_2017_143_150.pdf

[23] Amy J. Ko. 2023. Computing Education Research FAQ. https://faculty.washington.edu/ajko/cer.

[24] David R. Krathwohl. 2002. A Revision of Bloom's Taxonomy: An Overview. *Theory Into Practice* 41, 4 (2002), 212–218. https://doi.org/10.1207/s15430421tip4104_2

[25] Stephanie Lunn, Maíra Marques Samary, and Alan Peterfreund. 2021. Where is Computer Science Education Research Happening?. 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, 288–294. https://doi.org/10.1145/3408877.3432375

[26] Stephanie Lunn, Maíra Marques Samary, Susanne Hambrusch, and Aman Yadav. 2022. Forging a Path: Faculty Interviews on the Present and Future of Computer Science Education in the United States. *ACM Trans. Comput. Educ.* 22, 4, Article 51 (sep 2022), 24 pages. https://doi.org/10.1145/3546581

[27] Lauri Malmi, Ian Utting, and Amy J. Ko. 2019. *Tools and Environments.* Cambridge University Press, 639–662. https://doi.org/10.1017/9781108654555.022

[28] Monica M. McGill, Sloan Davis, and Joey Reyes. 2022. Surfacing Inequities and Their Broader Implications in the CS Education Research Community. In *Proceedings of the 2022 ACM Conference on International Computing Education Research - Volume 1* (Lugano and Virtual Event, Switzerland) *(ICER '22)*. Association for Computing Machinery, New York, NY, USA, 294–308. https://doi.org/10.1145/3501385.3543969

[29] Mia Minnes. 2022. Designing TA Training for Computer Science Graduate Students: Remote and Self-Paced Options for A Supported Introduction to Reflective Teaching. 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, 752–758. https://doi.org/10.1145/3478431.3499342

[30] Diba Mirza, Phillip T. Conrad, Christian Lloyd, Ziad Matni, and Arthur Gatin. 2019. Undergraduate Teaching Assistants in Computer Science: A Systematic Literature Review. In *Proceedings of the 2019 ACM Conference on International Computing Education Research* (Toronto ON, Canada) *(ICER '19)*. Association for Computing Machinery, New York, NY, USA, 31–40. https://doi.org/10.1145/3291279.3339422

[31] Chrystalla Mouza, Aman Yadav, and Anne Ottenbreit-Leftwich (Eds.). 2021. *Preparing Pre-Service Teachers to Teach Computer Science: Models, Practices, and Policies.* Information Age Publishing Inc., United States.

[32] The Carnegie Classification of Institutions of Higher Education. 2023. About Carnegie Classification. https://carnegieclassifications.acenet.edu/.

[33] Eduardo C. Oliveira, Ronaldo C. M. Correia, Rodolfo Azevedo, Simone Telles, Alessandra A. Macedo, and Roberto A. Bittencourt. 2022. A Computational Thinking Course for Pre-Service Teachers. In *2022 IEEE Global Engineering Education Conference (EDUCON)*. 1082–1088. https://doi.org/10.1109/EDUCON52537.2022.9766601

[34] Seymour Papert. 1980. *Mindstorms: Children, Computers, and Powerful Ideas.* Basic Books, Inc., USA.

[35] Judy Sheard, S. Simon, Margaret Hamilton, and Jan Lönnberg. 2009. Analysis of Research into the Teaching and Learning of Programming. In *Proceedings of the Fifth International Workshop on Computing Education Research Workshop* (Berkeley, CA, USA) *(ICER '09)*. Association for Computing Machinery, New York, NY, USA, 93–104. https://doi.org/10.1145/1584322.1584334

[36] Simon. 2007. A Classification of Recent Australasian Computing Education Publications. *Computer Science Education* 17, 3 (2007), 155–169. https://doi.org/10.1080/08993400701538021

[37] Simon. 2009. Informatics in Education and Koli Calling: a Comparative Analysis. *Informatics in Education* 8, 1 (2009), 101–114. https://doi.org/10.15388/infedu.2009.07

[38] Office of the Press Secretary The White House. 2016. FACT SHEET: President Obama Announces Computer Science For All Initiative. https://obamawhitehouse.archives.gov/the-press-office/2016/01/30/fact-sheet-president-obama-announces-computer-science-all-initiative-0.

[39] David W. Valentine. 2004. CS Educational Research: A Meta-Analysis of SIGCSE Technical Symposium Proceedings. In *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education* (Norfolk, Virginia, USA) *(SIGCSE '04)*. Association for Computing Machinery, New York, NY, USA, 255–259. https://doi.org/10.1145/971300.971391

[40] Aman Yadav, Sarah Gretter, Susanne Hambrusch, and Phil Sands. 2016. Expanding computer science education in schools: understanding teacher experiences and challenges. *Computer Science Education* 26, 4 (2016), 235–254. https://doi.org/10.1080/08993408.2016.1257418