The Teacher Accessibility, Equity, and Content (TEC) Rubric for Evaluating Computing Curricula

DAVID WEINTROP, University of Maryland, USA
MERIJKE COENRAAD, University of Maryland, USA
JEN PALMER, University of Chicago, USA
DIANA FRANKLIN, University of Chicago, USA

In response to the growing call to bring the powerful ideas of computer science to all learners, education decision makers, including teachers and administrators, are tasked with making consequential decisions on what curricula to use. Often, these decision makers have not been trained in computer science and are unfamiliar with the concepts taught and tools used. This is especially true in K–12 contexts where computer science expertise is less prevalent. To aid in the decision-making process around computing curricula, this article introduces the TEC Rubric. The TEC Rubric is composed of three main categories: Teacher Accessibility, Equity, and Content designed to support educational decision makers and designers when it comes to computing instruction. Along with presenting the full rubric and the process used in its creation, this article describes two examples of the rubric in action. First, the TEC Rubric is used to evaluate two widespread computer science curricula to demonstrate its evaluative capacity highlighting differences between the two curricula. Second, we show how the TEC Rubric can be used to help inform the design of new K–12 computing curricula. Overall, the TEC Rubric is designed to serve as a useful resource in the ongoing quest to bring effective, equitable, and engaging computing instruction into schools around the world.

CCS Concepts: • **Social and professional topics → Computing education**; **K-12 education**;

Additional Key Words and Phrases: Computing curricula, equity, rubric, K–12 education

ACM Reference format:
David Weintrop, Merijke Coenraad, Jen Palmer, and Diana Franklin. 2019. The Teacher Accessibility, Equity, and Content (TEC) Rubric for Evaluating Computing Curricula. *ACM Trans. Comput. Educ.* 20, 1, Article 5 (December 2019), 30 pages.
https://doi.org/10.1145/3371155

This work was made possible through generous support from the National Science Foundation (Grant no. CNS 1738758). Any opinions, findings, conclusions, and/or recommendations are those of the investigators and do not necessarily reflect the views of the Foundation.
Authors' addresses: D. Weintrop, University of Maryland College of Education and College of Information Studies, 2226H Benjamin Building, 3942 Campus Dr. College Park, MD 20742; email: weintrop@umd.edu; M. Coenraad, University of Maryland, College of Education, 2226 Benjamin Building, 3942 Campus Dr. College Park, MD 20742; email: mcoenraa@umd.edu; J. Palmer, University of Chicago, UChicago STEM Education, 1427 E. 60th St., Chicago, IL 60637; email: hellige@uchicago.edu; D. Franklin, University of Chicago, Department of Computer Science, 5730 S. Ellis Ave, Chicago, IL 60637; email: dmfranklin@uchicago.edu.

1946-6226/2019/12-ART5 $15.00
https://doi.org/10.1145/3371155

1 INTRODUCTION

Around the world, there is a growing push to bring the foundational ideas of computer science into K–12 classrooms [7, 58]. This effort takes various forms depending on the local context. For example, some countries have instituted national computing curricula, such as England, New Zealand, and Israel [6, 18, 51]. Other efforts are decentralized, allowing individual states, districts, or schools to decide how computing is brought into schools, as can be seen in the United States and India [87, 111]. Shared across these initiatives is the underlying belief that providing learners with the opportunity to develop a basic understanding of core computational ideas is an essential part of preparing them to be productive citizens in our increasingly technological world. Over the past decade, the foundational skills and practices associated with computing have been captured by the umbrella term Computational Thinking [112]. While the exact definition of computational thinking continues to be debated, it is generally agreed that computational thinking is a collection of concepts and practices central to computer science, including developing algorithms, working with abstractions, and a systematic approach to problem-solving that identifies how and when computing can be used as part of a solution [49]. Historically, these ideas have been taught in high school computer science classrooms. However, as the role of computing both in and out of the classroom has grown, there are new needs and opportunities to teach computing across the curriculum and across grade bands [67, 99]. One result of the shift towards the language of computational thinking rather than computer science is allowing for more open conversations around how and where these ideas can be integrated across K–12 education. As Wing wrote in the article that began the current computational thinking conversation: "To reading, writing, and arithmetic, we should add computational thinking to every child's analytical ability" [112].

A central component of the modern view of the role of computing in education is the notion that computing is essential for ALL learners. In this article, we use the term computing to describe both computer science content as well as broader computational thinking–related concepts and practices. In emphasizing the universality and broad applicability of the powerful ideas of computing, the push to bring these concepts and practices into classrooms confronts issues of equity and access that have historically plagued technology fields [4, 72, 73, 82, 116]. In doing so, the Computing/Computer Science for All push seeks to reevaluate who learns computing, how it is taught, and where it resides within K–12 education. While the Computing/Computer Science for All language is prevalent in the United States, similar efforts are underway worldwide. Achieving the goal of bringing quality, accessible, and equitable computing instruction to all learners requires confronting several challenges currently faced by those tasked with realizing this lofty goal. These challenges can be unique to the given school, district, state, or country, but there are also shared challenges faced across localities and contexts. Central among these is the challenge of deciding what technology to use, what curriculum to follow, and if and how it needs to be altered to meet the specifics of a given classroom or school.

The consequential and complex decision of what computing curricula to use is often made with little in the way of scaffolds or guides to help decision makers. Decision makers often rely on instinct, reputation, or experience from other disciplines, rather than a deep knowledge of best practices for teaching computing or prior experience with the subject or materials. This article seeks to address this issue through the introduction of the TEC Rubric, a computing curriculum evaluation instrument designed to help educational decision makers and teachers make informed decisions about which computing curriculum to use in their classrooms. The TEC Rubric (https://www.canonlab.org/the-tec-rubric) is comprised of three main categories by which materials are to be evaluated: Teacher accessibility (T), Equity (E), and Content (C). For each of these categories, the TEC Rubric provides specific criteria by which to evaluate a curriculum to structure and facilitate the decision-making process and aid educators and

educational decision makers in making an informed and ultimately successful decision. The rubric was developed as part of a researcher-practitioner partnership between computer science education researchers and a large urban school district. Having this input from key members of the audience, the TEC Rubric was designed to help ensure it would be useful and actionable for the educational decision makers it is targeting. The rubric is informed by existing rubrics for evaluating curricula but is customized to meet the specific challenges associated with computer science and integrating computing into the classroom. It is important to note the intention is not for the TEC Rubric to supplant professional development or serve as a way to transform teachers into computer science experts; instead, it is to support education-related decision makers in making better-informed decisions with respect to computing education materials. A secondary audience for the TEC Rubric is the curriculum designers themselves, serving as a support for helping them ensure their materials meet classroom needs.

The focus of this article is on the TEC Rubric and a demonstration of how it can be used in two distinct capacities: to evaluate existing computing curricula and to inform the design of new computing and computer science curricula. This article begins with a discussion of the motivation and audiences for this work, before presenting the prior work that has informed this effort. Next, the article presents the methodology used to develop the rubric. The TEC Rubric is then presented, including details for each of the three categories and a discussion of how each construct in the rubric can be operationalized. Next, the article presents two distinct applications of the rubric, showing how it can be used to evaluate existing curricula as well as inform the design of a new curriculum. The article concludes with a discussion of the challenges and limitations of the rubric.

2 MOTIVATION

The need for the TEC Rubric emerged through a series of discussions between computing education researchers, computer science teachers, and district leaders around how they make computing curriculum-related decisions. The practitioners in the conversation were particularly focused on the goal of broadening participation in computing but did not themselves have the expertise to know if or how a specific curriculum would address this goal. At the same time, the researchers in the conversation were in the early stage of designing an equity-focused curriculum but had little initial knowledge of the needs of the teachers and district decision makers with respect to what needed to be included to make the curriculum easily adoptable, especially as it related to ways to support teachers with little prior experience teaching computing and schools with little history of computing instruction. In response to these differing needs of both the researchers and practitioners, we set out to create a resource to support both parties. Through these discussions and a review of similar efforts in other disciplines, three interrelated factors emerged as the focus of the rubric: teachers (T), equity (E), and content (C).

The first factor for choosing computing materials for classrooms is a consideration of educators' prior knowledge, experience, and confidence for teaching the computing content. Identifying the supports that are or are not present in a set of instructional materials is important for ensuring teachers are comfortable and able to effectively use them in their classrooms. This is critical, as many instructors tasked with teaching computing courses are new to the discipline and have relatively little prior experience teaching the subject [24, 32, 109].

A second consideration is thinking through how to make the ideas being taught resonate with the learner and align with the goals of equity and inclusiveness that are a cornerstone of the larger movement to bring computing to all. Bringing computing into the curriculum for all learners provides access, an important first step in addressing issues of underrepresentation. Access is necessary but not sufficient for achieving equity. In other words, just making sure that computing classes are offered does not mean all students are receiving the same quality of instruction, nor

that the instruction is suitable for all students. A growing body of research is revealing the importance of designing inclusive and accessible computing materials to support learners from diverse backgrounds in having meaningful, positive learning experiences [29, 57, 63, 64, 95]. As such, the push to bring computing to all students through K–12 education, if done well and in a way that is informed by this work, provides the opportunity to address historical inequities in computing.

A final consideration is related to disciplinary content and whether it is developmentally appropriate for the given context. Frameworks, standards, and research focusing on K–12 education are starting to define what computing instruction looks like across the K–12 spectrum [19, 61, 91, 96], but not all classroom curricula map their materials to these recent documents, so this is a task left to the curricular decision makers and teachers. Identifying and choosing to use curricula that present the content in accessible and age-appropriate ways, consider learners' previous and future computing instruction, and align to computing-related content standards is essential for ensuring consistent and coherent instruction across their K–12 careers.

3 AUDIENCE

Given the origin of this effort, care was taken to ensure that the TEC Rubric would be useful for the two groups involved in its creation: educational decision makers and curriculum designers. For educational decision makers, including teachers, instructional coaches, and administrators, the TEC Rubric is intended to serve as a tool for evaluating potential computing-related curricula. Specifically, the rubric focuses educators' attention on features of the curriculum associated with ease of adoption and effective implementation. Educational leaders who are in a position to make curricular decisions for a school, district, or some other learning context can use the TEC Rubric to make sense of the plethora of new tools and curricula that are continually being introduced. This issue was highlighted by a teacher who participated in a professional development session related to this project who wrote: "*There are so many computer science game-like resources that it is difficult to determine quality, focus, and educational benefits of each.*" In addition to evaluating curricula to inform the decision about whether or not to adopt them, the rubric can also be used to guide instruction by identifying gaps or shortcomings. Teachers with little decision-making authority can perform gap analysis, guiding them to supplement or modify lessons when necessary, which can be particularly important when trying to alter instruction to make it more culturally responsive and relevant to the students.

For computing curriculum designers, the TEC Rubric can serve to inform what to include in materials design and what is missing in later evaluation stages. The term "curriculum designer" is meant to capture all who contribute to the creation of computing curricula, including those who work on the technical aspects as well as those focused on the pedagogical dimensions of the materials. Oftentimes, the software developers and engineers who are building new educational technologies lack a background in education or learning theory to make well-informed design decisions with respect to critical dimensions of the learning environment. In such cases, these creators rely on intuition or lessons learned from their own educational experiences. For this audience, the TEC Rubric can help guide them to attend to specific aspects of the tools and materials they are developing to support equitable and effective outcomes. At the same time, educators and curriculum designers who have the expertise in pedagogy and learning theory often lack the content expertise to create exceptional computing curricula. The TEC Rubric can highlight opportunities to introduce overlooked content areas or assessment opportunities, integrate ideas or practices related to issues of equity and access, or add additional supports for teachers who might have little prior computing experience. Through applying the rubric, new considerations of uses of technology, characteristics of audiences, or types of interactions that can be supported may be identified, thereby positively impacting the resulting design.

4 REVIEW OF LITERATURE

In this section, we present a review of the literature that helped inform this work and upon which the TEC Rubric is built. The section begins by presenting a review of prior work for each of the three dimensions of the Rubric: Teacher Accessibility, Equity, and Content. The goal with this organization is to help justify the structure of the rubric and show how each of the dimensions of the rubric builds off existing research from the field of computing education research. This section concludes with a review of similar efforts to build evaluative curricular rubrics both within computing and beyond showing how the TEC Rubric is situated relative to such efforts.

4.1 Prior Work on Teacher Accessibility in Computing Education

Research shows that teachers play an essential role in introducing learners to a discipline, supporting their intellectual growth, and fostering a positive and equitable learning space [8, 23, 80]. In this way, teachers have the potential to act as change agents [38], which is particularly consequential in computer science, where issues of inequity and a lack of diversity have been persistent [45]. One of the central challenges in bringing computing to all learners is preparing teachers to integrate the topic into their existing classrooms or training them to teach stand-alone computing classes [24, 40, 114]. In the United States, this can be seen in the CS10K initiative, which sought to train 10K new computer science teachers [21, 22]. Other countries undertook similar large-scale computer science teacher training efforts, such as the United Kingdom, Israel, and New Zealand [6, 18, 39, 86].

One of the outcomes of these and related efforts has been increased insights into the specific challenges associated with preparing teachers to teach computing and computer science [22, 40, 56, 81, 86]. In their studies of pre-service computer science teachers, Yadav and colleagues [113, 114] identified a number of challenges new teachers face, including content mastery, pedagogical content knowledge, approaches to assessment, as well as cultural challenges such as isolation and underdeveloped computer science teacher professional development infrastructure. Supporting teachers new to the domain of computer science is essential given the roles they play in implementing curricular changes [5, 107]. A related challenge is a recognition that computer science as a field is rapidly changing, so even teachers who are not new to the domain of computer science face challenges related to rapid curricular and technological change [68]. In response to these new demands placed on teachers and the challenge of preparing them for new material, continuing professional development is playing a significant role [97].

A key component of the effort to train new computer science teachers is identifying ways to support those that are new to the domain and design materials and professional development opportunities to make the content more accessible and support novice teachers in instruction. Sentance and Csizmadia [98] conducted a large-scale survey of computer science teachers and identified a lack of subject knowledge and challenges associated with learning and staying up to date with computing materials as a key challenge they face. Making materials more accessible and more easily adopted for teachers is a direct way to combat this challenge. One potential solution is looking to online computer science instructional materials, which are often made freely available online; however, research shows these materials are largely underutilized and often miss opportunities to make themselves more accessible and easy to use by novice teachers [34, 66, 76].

4.2 Prior Work on Equity in Computing Education

As the field of computer science grows, it continues to be plagued by the underrepresentation of women, minorities, persons with disabilities, and individuals who identify with more than one of these groups. This underrepresentation is on full display when examining the computer science

degrees awarded to students each year. In 2017, just 19.3% of doctoral degrees in computing were awarded to women and less than 3% of all CS doctoral graduates were Black or African American, Native Hawaiian/Pacific Islander, Hispanic, or Multiracial [116]. No doctoral degrees in computing were awarded to students who identified as American Indian or Alaska Native, with undergraduate computing degrees following a very similar pattern in the United States [116].

As computing moves into K–12 schools, the same lack of representation is present. Using the United States as an example, of the students who took an AP computer science exam (either AP Computer Science A or AP Computer Science Principles) in 2017, just under 25% identified as being from a racial category typically considered to be underrepresented (American Indian or Alaska Native, Black, Hispanic or Latino, Native Hawaiian or Other Pacific Islander, Two or More Races, or Other) [104]. The issue extends beyond participation to include performance. The AP test is graded on a five-point scale with one being the lowest score and five being the highest. Students identifying as an underrepresented minority scored just 12.7% of the fives awarded on both AP Computer Science tests and 41.23% of the ones awarded on both tests [103].

The likelihood that a student takes and succeeds in a computer science course is associated with their level of access to courses, qualified teachers, and classroom resources available. All of these factors are often lacking in schools with populations composed of a majority of racial minority students. The absence of minority students in computing courses is typically caused by structural barriers [108], including the disproportionate availability of those classes [73]. Compared to White students, Black students report less access to a dedicated computer science class in school and are more likely to learn computer science through after-school clubs or groups [47]. The underrepresentation of Black and Hispanic students in computer science courses is not due to a lack of interest or confidence in their ability to learn the content. Black students report feeling more confident in their computer science abilities than both White and Hispanic students and are more interested in learning computer science—with surveys finding that one-third of Black and Hispanic students are interested in learning computer science compared to just one-fifth of White students [47]. Additionally, Black and Hispanic parents are more confident their children will learn computer science and are more interested in having their children learn computer science than White parents [47].

The underrepresentation of minorities in computer science courses has led to efforts to broaden participation in computing fields through courses and curricula. Knowing these students want to take computer science courses but are being structurally and systematically left out has resulted in numerous efforts to increase access and opportunities. A major effort is underway to reframe computer science in an effort to broaden participation by changing people's perceptions of computing. This includes making computing tasks and curricula more social and collaborative [9], focusing on physical building rather than just programming on a screen [12, 53], and presenting programming as a creative endeavor [88, 92]. Projects also strive to make computing connect more deeply with students' interests and fit their individual needs. Building on theories related to culturally relevant pedagogy [41] and students' funds of knowledge [77], projects aimed at broadening participation examine students' ways of knowing [53], integrate their communities and cultural capital [35, 63], and try to make the curriculum personally and culturally significant [26, 43, 71, 95]. Researchers have also worked to add new entry points into computing as well as to elongate the computer science pipeline to include younger grades with interventions placed throughout the grade range [31, 43, 50, 110].

4.3 Prior Work on Computing Content

Work on equity informs decisions on how to teach computing, but not what to teach at what ages. While the field of computer science is not new, defining exactly what it is and what should be included in a well-rounded computer science education is a topic of continued debate (e.g.,

References [3, 26]). This is largely due to the rapidly changing nature of the field, as the CSTA writes in the introduction to their standards document: "Computer science standards cannot be static. These standards must be reviewed and updated on a regular basis, and not considered complete and finalized" [19]. Standards documents and disciplinary frameworks are especially important for computer science, as there are widespread misconceptions around the field and what it means to practice computer science [46]. The past decade has seen this discussion expand from higher education to include all of K–12 education. Until relatively recently, computer science in K–12 education has resided in the final years of high school and largely been a programming-focused discipline.

K–12 standards must be informed by research, and research in this area is still emerging. Elementary school instruction often uses Scratch [88], a visual block-based language designed for this age group. Work on the developmental aspects of learning computer science with Scratch began when Seiter and Foreman [96] analyzed Scratch artifacts in the public repository to correlate the grade band of classrooms with the blocks they were using, developing the Progression of Early Computational Thinking (PECT). However, as Brennan and Resnick have shown [10], the presence of a block does not necessarily imply understanding. Dwyer et al. [28] used student focus groups to explore lower anchor points and learning progressions related to algorithmic thinking. Franklin et al. [37] analyzed fourth- through sixth-grade student work to analyze similarities and differences in solutions related to sequence, events, and initialization. Grover et al. [48] also studied misconceptions of loops, variables, and Boolean logic. More recent work by Rich and colleagues has created learning trajectories by extracting learning goals from a broad set of literature and using several heuristics to create dependencies and partial orderings between goals [89–91].

A significant step forward in defining a broader vision of computer science across the K–12 spectrum came with the publication of the K–12 Computer Science Framework [61]. This document is intended to serve as a guiding framework that outlines the major content areas and practices in the field of computer science and provide a structure for the creation of standards across the K–12 spectrum. The utility of this work is that it defines computer science as more than just programming, including five high-level concepts and eight central computing practices. It also situates the discipline relative to the broader concept of computational thinking, showing both the overlap and distinctions between the two. The framework goes on to map out grade-appropriate versions of each concept and practice across K–12. This work is particularly useful, as this more expansive vision of computer science spread across K–12 is quite distinct from the programming-centric version of computer science that has historically been the focus of pre-college computer science.

4.4 Prior Work on Evaluative Rubrics in Education

Rubrics have long been used in the field of education to assist with the fair and consistent assessment of student work [3, 102]. Rubrics can serve as a self-monitoring tool for students by providing a clear set of criteria by which their work will be assessed [59, 60, 70, 83, 84]. The well-defined grading criteria in a rubric make expectations clear to students as well as create a fair and consistent basis on which student projects can be assessed—even between graders [14] or by automated assessment tools [27].

When selecting a new curriculum, schools and districts must weigh the many available options to identify the program that best aligns with standards and meets the needs of their teachers and students [13, 79]. Program evaluation rubrics are a useful tool to help facilitate this process [11, 74]. One such set of rubrics is the suite of EQuIP rubrics for Science, Math, and ELA. The EQuIP rubrics were designed to identify curriculum products that are aligned to the Next Generation Science Standards (NGSS) or the Common Core State Standards (CCSS). The EQuIP Rubric for Science is composed of a set of criteria fitted into three categories that collectively describe the

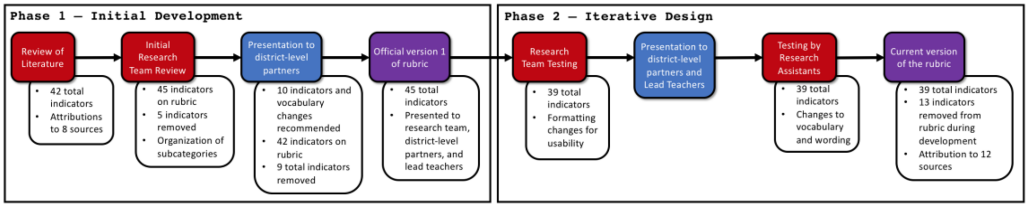

Fig. 1. Rubric design and development process.

type of instruction described by the NGSS with the intent of the user identifying not only the presence of each indicator but also detailed evidence describing the occurrence [74].

As part of the development of a culturally relevant computing curriculum for middle-grades students, the TEC Rubric developers required a program evaluation rubric for computer science that could measure the degree to which each module in the curriculum met their standards for teacher accessibility, equity, and content. The TEC Rubric provided necessary information to developers that they could use as they developed or revised the modules to ensure each met their requirements.

Program evaluation rubrics exist for computer science curricula, but their focus is on measuring teacher effectiveness and what happens in the classroom [17, 105, 106]. These existing rubrics do not measure the potential offered by a curriculum, independent of implementation, nor do they aid in the comparison and evaluation of curricular products. For our purposes, we required a program evaluation rubric with a structure similar to the EQuiP Rubric for Science—with the criteria divided among larger categories. Since no such rubric existed, one was created to inform the revision of the materials. Rather than designing the TEC Rubric to be curriculum-specific, we instead designed it to be curriculum-independent, as we felt there was a gap in the collection of program evaluation rubrics for computer science. The resulting rubric was revised for usability and increased reliability [115].

5 METHODOLOGY

The TEC Rubric was designed by a team of researchers and practitioners with expertise in K–12 computing education as part of a larger researcher-practitioner partnership (RPP) [15]. The core design team was composed of computer scientists, learning scientists, and curriculum developers with extensive experience in K–12 computer science working in close collaboration with the computer science leadership in the partner school district. In bringing together these two populations in an RPP, the team combines the disciplinary and methodological expertise of the researchers with the results and impact orientation and expertise of those working in school districts and classrooms directly. These different sets of expertise draw directly from the complementary sets of experiences between those in academia and those in K–12 education and are a central feature of why RPPs have been so successful and impactful in recent educational interventions broadly [85] and computer science initiatives in particular [52].

The development of the TEC Rubric happened in two phases: (1) initial development based on existing rubrics and literature followed by (2) iterative refinement (see Figure 1). The first phase of the project was led by the research team. Throughout the second stage of development, all members of the researcher-practitioner partnership played active roles in evaluating and refining the rubric. One feature of the core design team is that it included members from the various audiences of the TEC Rubric, including teachers, district-level decision makers, and curriculum designers. The full research team met weekly throughout the development of the rubric to ensure continuous and open lines of communication.

The creation of the TEC Rubric began by reviewing existing curricular rubrics designed to achieve similar outcomes for other disciplines as a means to guide the structure and scope of our efforts. The format of the TEC Rubric is based on the EQuIP Rubric for Science, Version 3.0 [117]. The EQuIP Rubric focuses on three high-level categories: Design, Instructional Support, and Student Progress for the Next Generation Science Standards. For each of these categories, the rubric provides subcategories as well as a means to evaluate a lesson for the subcategory. The EQUIP rubric is widely used and cited as an exemplary rubric for evaluation [115]. The TEC Rubric follows a similar structure; however, modifications were made given the difference in content. Specifically, the EQuIP rubric is designed for the NGSS standards, so its need to evaluate content is not the same as the TEC Rubric, which is not mapped to a specific framework. Also, as the TEC Rubric places an emphasis on issues of equity, equity was promoted to be a top-level category rather than live nested under another category. We also replicated the EQuIP rubric's structure for application by including columns in the resulting document to provide Evidence and Reasoning and Suggestions for Improvement as part of the evaluation process.

To create an initial version of the TEC Rubric, we reviewed materials that are currently available for computer science curriculum and pedagogy evaluation (e.g., References [3, 11, 21, 58, 72]). We also reviewed related efforts for other disciplines, including content-agnostic evaluation tools (e.g., References [97, 99]) to further inform the rubric. While reviewing these materials, evaluation criteria that were identified as pertinent to the design and evaluation of introductory computing curricula were recorded for inclusion in our rubric. Criteria with similar intents from different sources were combined. An initial list of criteria was generated and then divided into the three categories that structure the rubric. The initial review resulted in 42 indicators on the rubric from eight different sources and team brainstorming. The majority of the indicators were attributed to the Achieve NJ interpretation of the Danielson Framework for Science [1] and the EQuIP rubrics for science and literacy [30]. In total, 18 attributions were made to the AchieveNJ Science Instruction Companion to the Danielson Framework, 7 attributions were made to the EQuIP Literacy rubric, and 5 attributions were made to the EQuIP Science rubric. Attributions were also made to Brennan and Resnick [10], Funds of Knowledge [77], the Danielson Framework [105], Hong and Choi [54], and Universal Design for Learning [93]. Some indicators included multiple attributions, because various sources highlighted the same best practices, and 10 indicators did not have attribution in the initial rubric development but were suggestions made based on teaching and curriculum design experience.

The criteria in this initial set were reviewed and evaluated by the research team to decide their fit within the CS context of the TEC Rubric project. Together, the team adjusted the indicators within the rubric, ending with a new version that included 40 indicators and reorganizing from the broad categories of Teacher Accessibility, Equity, and Content into subcategories. Once a first draft of the TEC Rubric was completed by the research team, it was presented to the larger development team, including the practitioner partners who gave input based on their experience working with various computing curricula and teachers with varying levels of prior experience teaching CS in their K–8 classrooms. Based on this feedback, the TEC Rubric was updated to include new sources (e.g., References [54, 66]) and wording was adjusted for clarity and generalizability. Specifically, the practitioner partners suggested three new indicators or wording changes based on Linsey, Graham, Jr, and Jew [69], two new indicators or wording changes based on Hoogstra, Tanyu, Tucker, and Loignon [55], and five indicators or wording changes based on policies and priorities of the school district. These were taken into account and incorporated into the rubric as new indicators or changes to the existing indicators. Based on these adjustments, the rubric was edited to include 42 total indicators. In the final development phase, the rubric came back to the research team where the indicators were purposefully ordered within the sub-categories, and

indicators were added based on identified holes in the evaluation, including the need for an indicator pertaining to youth culture rather than focusing only on heritage and family culture. The final development version of the rubric included 45 indicators.

The second phase of the development process involved the iterative refinement of the rubric by applying it to individual modular components of an elementary computing curriculum under development as part of this same project. This curriculum, titled Scratch Encore, is focused on issues of equity, accessibility, and ease of adoption, making it a good test case for the rubric [36]. The TEC Rubric was applied to one curricular module at a time by multiple researchers working individually, taking note not only of how the module scored but also the experience of using it. Based on their experience, the TEC Rubric was revised to remove repetitive criteria and revise criteria that could not be objectively measured (as identified through poor interrater reliability scores). For example, an indicator about the ability of the curriculum to promote and develop TPACK [62] was removed, because it was not measurable by the everyday person who might not be familiar with this concept and we felt other indicators together subsumed this topic. The visual design of the rubric was also revised to make it easier to apply and more useful for communicating results. Specifically, checkmarks were added for each indicator to help users track when the indicators were and were not met, and the evidence blocks were adjusted to the level of each indicator rather than level of sub-category. Both of these adjustments were made to encourage users to think about indicators specifically rather than subcategories as a whole to improve the feedback received from the rubric. Once this round of development was complete, a candidate final version of the TEC Rubric with 39 indicators was distributed to the full team of researchers and district-level partners for approval and final adjustments. The rubric was also presented to lead teachers working on curriculum development for their suggestions and comments.

The final phase of development was to train a set of users on how to apply the rubric and then have them apply it to CS curricular materials. Five undergraduate computer science students were recruited and trained as research assistants during the development and refinement of the TEC Rubric. While being trained on the rubric, they helped identify places where the wording of criteria needed to be revised to clarify intent. The outcomes of this portion of the development cycle are presented later when we discuss validity and reliability with respect to the rubric. They are mentioned here to document the final round of revisions that went into the development of the TEC Rubric, which is presented in the next section.

6 THE TEC RUBRIC FOR COMPUTING CURRICULA

The TEC Rubric was designed to help educators and educational decision makers and designers evaluate introductory computing curricula. Recognizing that curricula come in different shapes and sizes, the TEC Rubric is designed to be useful to evaluate individual lessons, groups of lessons, and whole curricula. The TEC Rubric is divided into three categories: Teacher Accessibility, Equity, and Content. These categories align with the major needs of introductory computing curricula. Each category, described below, is further broken into subcategories, which themselves are specific and concrete criteria. The goal of the subcategories is to highlight specific characteristics that may or may not be present in a set of curricular materials. In providing these subcategories, the evaluation of a curriculum becomes less subjective and more specific as concrete information about the curriculum emerges. The hierarchical structure (categories composed of subcategories composed of criteria) are intended to provide a way to go from specific features of a curriculum to an overall sense of if and how the curriculum rates with respect to more general characteristics such as equity. Each criterion has a checkbox for the reviewer to track whether or not the criterion has been met. The TEC Rubric also includes space for reviewers to provide evidence and reasoning for their ratings, as well as suggestions for potential improvements. Before presenting

Teacher Support	
	Includes a full lesson plan for teacher preparation and planning
	Materials are educative and accessible for teachers with differing CS content knowledge (i.e., definitions and examples of CS concepts are offered to support teacher learning)
	Makes connections to CS topics covered in past lessons
	Materials provide teachers with common misconceptions and challenges that students have regarding the concepts and potential explanations or solutions
Supplemental Materials	
	Provides student questioning and discussion prompts
	Lessons include student facing activity guides that can be given to students in paper form or digitally in order to direct their work
	Includes relevant worksheets
	Provides teachers with assessment materials

Fig. 2. The Teacher Accessibility dimension of the TEC Rubric.

the rubric itself, it is important to note that the TEC Rubric focuses specifically on curricular materials and does not take into account professional development or pedagogical strategies that may accompany a given curriculum. We agree with others who have argued for the critical importance of the training that accompanies curricular materials [44]; however, we also see the utility and, at times, the necessity of evaluating curricular materials independent of associated professional development. Below, we present each of the three categories, beginning with a figure showing the rubric itself, followed by a discussion of its constituent parts.

6.1 Teacher Accessibility

This first dimension of the TEC Rubric is Teacher Accessibility (Figure 2). Since many teachers who teach computing or computer science courses often have little or no prior experience with the field, it is essential that curricular materials are educative and provide pedagogical support for teachers. This can take the form of full lesson plans with a thorough discussion of both the topics covered (content knowledge) as well as guidance on how to teach the material (pedagogical content knowledge) [100]. These aspects of the curriculum are captured by the Teacher Support subcategory. The Supplemental Materials subcategory relates to additional materials available to teachers, including discussion prompts, student guides, worksheets, and assessment materials. Whether distributed in a physical form or digitally through a student portal, these materials are an essential component of a curriculum, as they support students as they work through projects and educators in their teaching of the curriculum and assessment of student learning.

6.2 Equity

The Equity category of the TEC Rubric (Figure 3) includes three subcategories: Culture, Identity, and Exceptionalities. Given the demonstrated underrepresentation of women and minorities in computing-related fields, it is essential that introductory computing curricula are designed to specifically support these students and allow them to see themselves and their culture represented in the curricula. The Equity category draws attention to the ways in which the curriculum provides the opportunity for students to see and celebrate both their culture and their personal identities

Culture (Community-level)
Reflects and highlights the diverse cultures, perspectives, languages, and community values of students with regards to cultural heritage and/or contemporary youth culture (e.g. popular video games or common student interests/activities)
Gives students the opportunity to share their own culture and cultural heritage
Connects learning to students' homes, neighborhoods, and communities
Identity (Individual-level)
Context is meaningful and authentic to students and connects to students' interests
Provides opportunities for students to contribute their knowledge and perspectives about a lesson's topic and share information about their life experiences
Students see themselves represented in the curriculum and classroom materials
Provides opportunities for students to represent themselves in their projects
Exceptionalities (ELL, Special Ed, etc.)
Provides multiple representations within the lesson by adapting for a variety of different types of learners using alternatives to reading, writing, listening, and speaking such as translations, pictures, or graphic organizers
Provides extensions that allow a deeper understanding of topics for students who meet the performance expectations
Assessment methods are accessible to all students and do not penalize or reward students due to exceptionalities

Fig. 3. The Equity dimension of the TEC Rubric.

as well as the accessibility and adaptability of a curriculum with regards to students who have exceptionalities (e.g., English Language Learners, special education students, and gifted students).

The Culture subcategory relates to community-level cultural resources with regards to both cultural heritage, contemporary culture, and youth culture. In contrast, the Identity subcategory focuses on individual-level cultural resources and the ability of students to see themselves represented in projects that they complete and have opportunities to represent themselves in projects that they make. This subcategory strives to measure whether or not aspects of an individual learner can be incorporated into assignments as part of the curriculum to support their developing identities as budding computational doers. Last, the Exceptionalities subcategory aims to broaden participation with regards to ability by encouraging multiple representations to support the diverse learning needs of students [94] and opportunities for further exploration for students who desire a deeper understanding. Across this category the goal is to ensure the curriculum under consideration is providing opportunities from learners from historically underrepresented populations to identify with computing in authentic and meaningful ways.

6.3 Content

The Content category (Figure 4) includes five subcategories: Computing Content, Instructional Design - Pedagogical Practices, Instructional Design - Content, Theme, and Assessment. These

Computing Content
Content aligns with standards (e.g. K-12 CSTA Computer Science Standards)
Content within the lesson is presented following a trajectory that begins with less complex topics and gradually increases in complexity with time
Uses appropriate disciplinary terminology and promotes students' use of disciplinary terminology
Instructional Design - Pedagogical Practices
Lesson is based on clear, measurable objectives (lesson goals) that are provided to the teacher
Each activity includes time for students to apply the skills that are being taught
Includes a mixture of instructional strategies (e.g. discussions, modeling, student activities, worksheets, projects, etc.)
Provides opportunities for students to collaborate
Instructional expectations are easy to understand and directions are easy for students to use
Students are provided with the opportunity to share their work with classmates and receive peer feedback
Instructional Design - Content
Considers students' prior knowledge to incorporate this knowledge into the lesson and/or cover material not previously covered
Questions promote higher order (apply, analyze, evaluate) thinking
Scaffolded to promote greater student understanding and independence as the learner progresses (e.g. gradually fades supports as student advances, utilizes the Use - Modify - Create sequence, etc.)
Lesson provides opportunities for students to explore and provide solutions to open-ended prompts
Content is appropriate to the grade band and complexity students can handle
Provides opportunities for students to reflect on their learning
Theme
Includes accurate coverage of the non-CS topics used as framing (e.g. historical events, groups, cultures, science topics, etc.)
Activities fit together cohesively with a clear storyline
Assessment
Assessments provide teachers with feedback on student progress towards a learning objective
Rubrics are based on objectives and standards and assist in measuring student proficiency
Objective-based formative assessments (i.e., student responses to question prompts, journal prompts) are present throughout the module and are incorporated within the instruction
Objective-based summative assessments are present in the lessons

Fig. 4. The Content dimension of the TEC Rubric.

Teacher Accessibility				
Sub Category		**Evidence and Reasoning**	**Suggestions for Improvement**	**Score**
Teacher Support				
☐	Includes a full lesson plan for teacher preparation and planning			
☐	Materials are educative and accessible for teachers with differing CS content knowledge (i.e., definitions and examples of CS concepts are offered to support teacher learning)			
☐	Makes connection to CS topics covered in past lessons			
☐	Materials provide teachers with common misconception and challenges that students have regarding the concepts and potential explanations or solutions			
Supplemental Materials				
☐	Provides student questioning and discussion prompts			
☐	Lessons includes student facing activity guides that can be given to students in paper form or digitally in order to direct their work			
☐	Includes relevant worksheets			
☐	Provides teachers with assessment materials			
	Overall Score:	3: Extensive evidence to meet at least one criteria (and at least adequate evidence for other category) 2: Adequate evidence to meet all criteria in the category 1: Adequate evidence to meet at least one criterion in the category, but insufficient evidence for one other criterion 0: Inadequate evidence to meet two or more of the criterion in the category		

Fig. 5. The full rubric presentation including subcategories and fields for Evidence and Reasoning, Suggestions for Improvement, and partial and total Score.

subcategories and the criteria within them represent best practices in lesson planning and curriculum development. They seek to demonstrate what is necessary for a comprehensive set of curricular materials that will support teachers of all levels and lead to classroom best practices. The subcategories cover the content of the lesson as well as the manner through which that content is introduced. It is important to note care was taken to support a diversity of pedagogical and curricular design approaches. As such, the rubric is not intended to be prescriptive or advance one instructional approach over another.

The first subcategory, Computing Content, checks for alignment to existing standards, the use of disciplinary terminology, and a trajectory of increasing difficulty throughout the lessons. The majority of the criteria within this subcategory pertain to the instructional design of the unit both with regards to the pedagogical practices and the content included within the curriculum. The two Instructional Design subcategories are not specific to introductory computing contexts specifically but represent best practices when working within technologically mediated learning contexts to promote positive classroom management and learning outcomes. Instructional Design - Pedagogical Practices relates to inclusions within the curriculum that promote strong teaching practices such as clear and measurable objectives and a variety of instructional strategies. The Instructional Design - Content subcategory relates to the ways in which content is presented to students, including the use of scaffolding and encouraging open-ended prompts for exploration. The Theme subcategory promotes accuracy of the non-computing-related aspects of a lesson through which the content is presented if there are any (e.g., if the computing content were situated within activities related to photosynthesis, then the science is accurate and the computing content is applied within the context in an authentic way). Finally, the Assessment subcategory pertains to the assessment strategies that are employed within the curriculum, both formative and summative.

6.4 Applying the Rubric and Summarizing Evaluations

The full TEC Rubric includes guidelines for how it is intended to be used to evaluate a curriculum (Figure 5). This includes documenting evidence that a given criterion has been met, noting

potential ways to improve or augment the materials, and giving a specific score to each criterion. The first step in using the rubric is to become familiar with it, being sure to note the categories, subcategories, and each criterion. Next, the reviewer should read through the curriculum that is being evaluated, being sure to complete activities as the students would, if possible. Having done that, the reviewer should next evaluate the curriculum based on each category individually. To do this, the reviewer should (1) Use the checkboxes to note which criterion the curriculum meets in each subcategory (note: checkboxes can be replaced by a scale for greater granularity if desired); (2) Record evidence of the curriculum meeting or not meeting each of the criteria. This evidence should be specific and include page numbers, exact activities, and so on; (3) Assign an overall score for each subcategory. The next step is for the reviewer to assign an overall rating for each category to the curriculum based on the individual category scales. Finally, if there are multiple reviewers involved, the full set of reviewers comes together to compare results and discuss discrepancies in ratings.

To assist with summarizing and communicating the results of using the rubric, a system was developed to facilitate quantifying and summarizing a completed evaluation based on similar efforts from the EQuIP rubric [117]. Each subcategory is given a score of inadequate, adequate, or extensive based on the average scores of the indicators that comprise the subcategory. In our system, greater than 3 is extensive, between 2 and 3 is adequate, and less than 2 is inadequate. These subcategory scores are used to assign a final category score: 3 if all but one subcategory receives a score of extensive, 2 if all criteria are rated at least adequate, 1 if one subcategory is rated inadequate with the other subcategories achieving an adequate rating, and 0 if more than one subcategory scores a rating of inadequate. It is not expected that most units will receive ratings of 3 in every category or even a majority of extensive ratings in subcategories; rather, the intention is for users to use the TEC Rubric to identify the strengths and weaknesses of a curriculum. Note, this is just one possible way to summarize results from applying the rubric that we have found useful and accurate for quantifying results; however, in our own experience, we have found the full detail of a completed rubric, particularly the written notes, to be valuable.

6.5 Validity and Reliability

Having presented the contents of the rubric, we now return to our development process and our efforts to ensure the validity and reliability of the rubric. For this effort, we use the term "validity" to refer to how well the rubric captures the reality of the curriculum (i.e., is it measuring what we say it is measuring), and we use the term "reliability" to reflect the consistency of the rubric and its application [78]. The validation of the TEC Rubric draws heavily from the expertise present in the researcher-practitioner partnership, which includes computer science education researchers, current and former computer science teachers, as well as district-level employees responsible for selecting and implementing computing curricula for the district. The validation process began with a test of initial face validity, meaning whether or not contents the rubric matched what the team members expected to be present and if and how it aligned with their prior experiences evaluating materials. Through a series of structured discussions, the rubric was iteratively revised to match both academic and practitioner expectations. The next phase of validation involved using the rubric to evaluate the curriculum under development. The experts on the team evaluated the curriculum on their own, and then the materials were evaluated using the rubric. We then compared the results to ensure the rubric was capturing the issues and oversights identified by our experts. Finally, two well-known computing curricula were evaluated with the rubric, and the results were compared to the reported focus for the materials. The goal with this last effort is to validate the rubric and what it identifies with respect to curricula created by experts outside of our project. Further details about this last phase are presented in the following section.

To evaluate the rubric's reliability, the TEC Rubric was applied to a curriculum by the five undergraduate research assistants working on this project. The assistants had no prior experience with the curriculum or with the TEC Rubric before joining the project. The undergraduates were trained on the use of the TEC Rubric by applying it to sequential modules in the introductory computing curriculum being designed alongside the rubric. This curriculum consists of 14 modules. The five research assistants individually coded a single module then came together to compare results before moving on to the next module. This was repeated for 6 of the 14 modules. For each module that was evaluated, they individually applied the TEC Rubric—denoting which criteria were met, documenting evidence of each criterion, and recording strengths and suggestions for improvement. Once they had each evaluated the same module individually, they came together to discuss their evaluations, sharing their reasonings and coming to an agreement about any discrepancies in their evaluations, which was documented on a new copy of the TEC Rubric to document the group's consensus ratings. The undergraduate research assistants continued to evaluate in this manner until they reached a level of substantial agreement on their independent evaluations using Fleiss' kappa. For each module that the group evaluated, the five research assistants individually rated six modules, with the final module achieving an inter-rater reliability kappa of 0.646, substantial on the interpretation scale [65].

Having reached a training level of reliability, the research assistants applied the TEC Rubric to two existing curricula, Exploring Computer Science [42] and CodeHS [16]. As with the evaluation of the modules evaluated throughout the development of the rubric, the research assistants rated each individually and then met to discuss the differences and come to a consensus on the overall evaluation of the curriculum. When using the TEC Rubric on Exploring Computer Science, the research assistants had a kappa value of 0.21, in the fair agreement range. When using the TEC Rubric on CodeHS, the research assistants achieved a kappa value of 0.72 in the substantial agreement range. The relatively low kappa score found for the Exploring Computer Science materials along with detailed results of these evaluations will be discussed in the next section, but for now serve as a show of reliability among a rather large group (five) of independent evaluators.

7 USING THE TEC RUBRIC

In this section, we show what it looks like for the TEC Rubric to be used in its two primary capacities: to evaluate existing computing curricula and to inform the design of new computing curricula. First, we present the results of applying the TEC Rubric to two well-known computer science curricula as a means to provide a sense of what it would look like for an educator tasked with choosing between these two curricula for their classroom. Next, we present vignettes from our own curricular design efforts showing how the TEC Rubric helped us evaluate and improve our materials.

7.1 Evaluating Existing Curricula

One goal of the TEC Rubric is to help educators and educational decision makers make informed decisions about whether or not a given curriculum would be a good fit for their classrooms. To demonstrate the utility of using the TEC Rubric in this capacity, we present the results of applying the rubric to two widely used, yet quite distinct, high school introductory computing curricula: the Exploring Computer Science (ECS) curriculum (http://www.exploringcs.org) and the CodeHS curriculum (https://codehs.com).

Exploring Computer Science is a year-long introductory computing curriculum that has an explicit focus on broadening participation in computing [42]. As part of this approach, ECS frames computer science as more than programming, including units on problem-solving, design, data analysis, and robotics alongside programming. ECS recently introduced a new unit to the

curriculum based on e-Textiles, further highlighting the focus on recruiting historically underrepresented populations to computing and presenting computing in a broad, inclusive way [33]. ECS is targeted at early high school–aged learners (ages 14+) and intended to introduce the big ideas of computing and lay the foundation for future instruction. For this evaluation, we focused on ECS' six-week programming unit based on the Scratch programming environment. This unit was chosen as it is relatively well aligned with the CodeHS materials evaluated, thus providing a more direct comparison.

CodeHS is an online computer science curriculum and platform that offers teachers and schools everything they need to bring computer science to their classrooms. This includes step-by-step curricula and extensive resources for teachers and administrators as well as support for educators in the form of CodeHS tutors and online support. CodeHS offers numerous curricula spanning grades and content areas, including introductory Java and Python courses, AP Computer Science Principles, and Web Design. For this article, we evaluated Unit 4 of the Introduction to Java course. We chose these materials, as they were part of the materials that CodeHS graciously gave us access to and because of the overlap with the ECS programming unit we evaluated.

Before continuing, it is important to note that this evaluation is not intended to be critical of either of these curricula. They were chosen in part due to the different foci but also due to the admiration we have for each and the widespread and positive impact they have both had. It is also important to note that the TEC Rubric focuses explicitly on curricular materials and does not consider accompanying professional development. Finally, we chose only one unit of ECS, so the totality of its equity content may not have been captured. As such, some of the gaps identified by the rubric may be addressed by other aspects of their own curriculum or the larger ecosystem associated with each curriculum. This decision is discussed in greater detail later in the discussion. Below, we present the results of evaluating the programming-based ECS materials for each of the three TEC Rubric categories. As a reminder, these evaluations were conducted by five independent undergraduate evaluators trained in applying the TEC Rubric. For each subcategory, the five reviewers met to share the results of their independent evaluations. Through discussion, a consensus was reached.

7.1.1 Teacher Accessibility. In discussing their scores for the Teacher Accessibility dimension of the TEC Rubric (Figure 6), reviewers identified strengths and weaknesses of both ECS and CodeHS with respect to teacher accessibility. For ECS, the reviewers identified strengths related to the instructional materials (easy-to-follow lesson plans, clear sequencing and connections to prior lessons, and useful rubrics for both students and teachers) and supplemental materials (useful graphic organizers and discussion prompts). At the same time, the review revealed some potential weaknesses, such as the need for teachers to make sample programs that require varying levels of prior experience and some missing technical guides for navigating the Scratch platform. For CodeHS, the reviewers found strengths in how it provides teachers with problem guides and walkthroughs including common errors along with detailed and useful student-facing guides. The concerns raised by the reviewers were that some lesson plans were relatively light on details, so teachers would need to fill in gaps.

7.1.2 Equity. With respect to the category of Equity, both curricula scored well on Exceptionalities, ECS scored well on Identity, and both had gaps in Culture (Figure 7). For ECS, the reviewers saw value in how projects were open-ended to allow for student expression and how ECS projects connected with youth culture. The reviewers also noted how the ECS materials connected to learners' homes, neighborhoods, and communities. At the same time, the reviewers felt there were missed opportunities to connect to students' cultural heritage and a concern that students with prior Scratch experience would be rewarded in a way that could be frustrating to novices.

ECS	CodeHS	Category/Subcategory
	✓	**Teacher Support**
✓	✓	Includes a full lesson plan for teacher preparation and planning
	✓	Materials are educative and accessible for teachers with differing CS content knowledge (i.e., definitions and examples of CS concepts are offered to support teacher learning)
✓	✓	Makes connections to CS topics covered in past lessons
	✓	Materials provide teachers with common misconceptions and challenges that students have regarding the concepts and potential explanations or solutions
✓	✓	**Supplemental Materials**
✓	✓	Provides student questioning and discussion prompts
✓	✓	Lessons include student facing activity guides that can be given to students in paper form or digitally in order to direct their work
✓	✓	Includes relevant worksheets
✓	✓	Provides teachers with assessment materials

Fig. 6. Teacher Accessibility scores for ECS and CodeHS.

ECS	CodeHS	Category/Subcategory
		Culture (Community-level)
		Reflects and highlights the diverse cultures, perspectives, languages, and community values of students with regards to cultural heritage and/or contemporary youth culture
		Gives students the opportunity to share their own culture and cultural heritage
✓		Connects learning to students' homes, neighborhoods, and communities
✓		**Identity (Individual-level)**
✓		Context is meaningful and authentic to students and connects to students' interests
✓	✓	Provides opportunities for students to contribute their knowledge and perspectives about a lesson's topic and share information about their life experiences
		Students see themselves represented in the curriculum and classroom materials
✓		Provides opportunities for students to represent themselves in their projects
✓	✓	**Exceptionalities (ELL, Special Ed, etc.)**
✓	✓	Provides multiple representations within the lesson by adapting for a variety of different types of learners using alternatives to reading, writing, listening, and speaking such as translations, pictures, or graphic organizers
✓	✓	Provides extensions that allow a deeper understanding of topics for students who meet the performance expectations
✓	✓	Assessment methods are accessible to all students and do not penalize or reward students due to exceptionalities

Fig. 7. Equity scores for ECS and CodeHS.

For CodeHS, the reviewers identified abundant support for exceptionalities and meaningful extensions for advanced students. They also saw the extensions that extended beyond the curriculum as a strength. At the same time, the reviewers felt CodeHS missed opportunities to connect the content with students' cultures or identities. Further, they felt some projects seemed to have "default" solutions, missing opportunities for student self-expression and personalization.

7.1.3 Content. In the Content category, both curricula scored very well, as is to be expected from such widespread and well-respected materials (Figure 8). The strengths for ECS content were how learners are given opportunities to apply concepts learned, the use of varied instructional strategies, and opportunities for collaboration. The few weaknesses identified in this section related to concerns around clarity of instruction and a focus on specific Scratch blocks over the more generalized treatment of the underlying concept. CodeHS also scored well in this category, with the reviewers commenting on comprehensive supports related to vocabulary and concepts, clear objectives, a diversity of activities and assessments, and giving learners the opportunity to modify existing code before having to write their own as a pedagogical strategy. Drawbacks mentioned included a lack of clarity on the mapping of the activities to standards and few opportunities for students to publicly share their work with others.

7.1.4 Evaluating Existing Curricula Discussion. In the above section, we can start to see how the TEC Rubric can provide useful information to curricular decision makers and help them make a more informed decision about whether or not to use a curriculum. Looking across these results, we can also see the priorities of the curriculum designers reflected in the evaluations. ECS, a curriculum that emphasizes equity and broadening participation, had more comprehensive coverage of the TEC Rubric Equity category than did CodeHS. At the same time, CodeHS, which places emphasis on ease-of-adoption and robust supports for novice teachers, had exemplary coverage with respect to Teacher Accessibility. This match between stated priorities and results is another useful data point in validating the rubric in that for both of these curricula, the evaluation results matched what was to be expected based on the stated objectives of the two curricula.

A few final notes before moving on to our second application of the TEC Rubric. First, it is important to reiterate that the Rubric was only applied to curricular materials and not professional development or other supplementary materials. Further, we only evaluated a subsection of the full curricula. This is important to mention, as it is very possible that some shortcomings noted above are covered by these other aspects of the curricula. Another important thing to mention as we close out this analysis is to mention that these are two exemplary curricula. As such, they both scored very well. This can be interpreted as the rubric not being critical enough or too coarse to be able to provide meaningful feedback; however, our explanation is that this is more a feature of the chosen materials and their overall strengths rather than a lack of resolution of the rubric.

7.2 Using the Rubric to Inform Curricular Design

Having shown what it looks like to apply the TEC Rubric to two existing curricula, we now turn our attention to a second, distinct use for the Rubric: aiding in the creation of new curricula. In this section, we present a vignette of how the TEC Rubric was used to inform the design of Scratch Encore, an upper elementary (ages 10–13) computing curriculum [36]. The same group of five undergraduate researchers from the previous section used the TEC Rubric to evaluate an initial version of a module the authors designed. We begin by introducing the overall goal of the module and how the ideas are presented to the learner. We then present the results of the reviewers' evaluation using the TEC Rubric and the resulting changes that were made in response to the feedback.

ECS	CodeHS	Category/Subcategory
✓	✓	**Computing Content**
✓	✓	Content aligns with standards such as the K-12 CSTA Computer Science Standards
✓	✓	Content within the lesson is presented following a trajectory that begins with less complex topics and increases complexity with time
✓	✓	Uses appropriate disciplinary terminology and promotes students' use of disciplinary terminology
✓	✓	**Instructional Design - Pedagogical Practices**
✓	✓	Lesson is based on clear, measurable objectives (lesson goals) that are provided to the teacher
✓	✓	Each activity includes time for students to apply the skills that are being taught
✓	✓	Includes a mixture of instructional strategies (e.g. discussions, modeling, student activities, worksheets, projects, etc.)
✓	✓	Provides opportunities for students to collaborate
	✓	Instructional expectations are easy to understand and directions are easy for students to use
✓		Students are provided with the opportunity to share their work with classmates and receive peer feedback
✓	✓	**Instructional Design - Content**
✓	✓	Considers students' prior knowledge to incorporate this knowledge into the lesson and/or cover material not previously covered
	✓	Questions promote higher order (apply, analyze, evaluate) thinking
✓	✓	Scaffolded to promote greater student understanding and independence as the learner progresses (e.g. gradually fades supports as student advances, utilizes the Use - Modify - Create sequence, etc.)
✓		Lesson provides opportunities for students to explore and provide solutions to open-ended prompts
✓	✓	Content is appropriate to the grade band and complexity students can handle
✓		Provides opportunities for students to reflect on their learning
✓		**Theme**
✓	✓	Includes accurate coverage of the non-CS topics used as framing (e.g. historical events, groups, cultures, science topics, etc.)
✓		Activities fit together cohesively with a clear storyline
✓	✓	**Assessment**
✓	✓	Assessments provide teachers with feedback on student progress towards a learning objective
✓	✓	Rubrics are based on objectives and standards and assist in measuring student proficiency
✓	✓	Objective-based formative assessments (i.e., student responses to question prompts, journal prompts) are present throughout the module and are incorporated within the instruction
✓	✓	Objective-based summative assessments are present in the lessons

Fig. 8. Content scores for ECS and CodeHS.

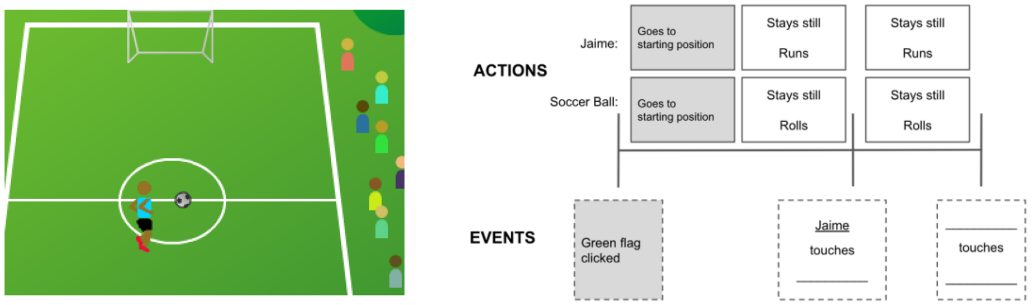

Fig. 9. The Scratch program for our Decomposition-by-Sequence activity (a) and the timeline from the accompanying student worksheet (b).

The lesson we present in this section is called Decomposition-by-Sequence. The main learning objective of this activity is to get students to start attending to the temporal aspects of programs—specifically, what triggers scripts to run and how can we programmatically control when they start and stop. When the sequence of actions involves multiple sprites, it does not work within Scratch to merely program a single script with the actions in order. Through this activity, students will develop an understanding of how to break down larger multi-sprite problems into a series of cause/effect steps and what Scratch blocks can be used to implement such a solution. The lesson is built around a sports theme; specifically, students are trying to get their sprite to score a goal by kicking the soccer ball into the net (Figure 9(a)). The specific sequence of actions the program follows is: (1) the soccer-playing sprite moves until it touches the ball, (2) the ball rolls from the center of the field until it touches the net, and (3) the crowd cheers. To accomplish this behavior, the wait until block is used to sequence each action based on when one sprite touches another (either the person touches the ball or the ball touches the goal). Accompanying the lesson is a worksheet that asks students to fill in a timeline of everything that happens in the program without looking at the underlying code (Figure 9(b)). Over the course of the module, students follow a use-modify-create sequence, starting by using a partially working program before reimplementing portions of it, then authoring a final project that has a similar structure (controlling sequential execution with the wait until block).

7.2.1 Teacher Accessibility. The undergraduate research team identified three criteria as being inadequate with respect to Teacher Accessibility: Materials are educative and accessible for teachers with differing CS content knowledge; Materials provide teachers with common misconceptions and challenges that students have regarding the concepts and potential expectations or solutions; and Lessons include student-facing activity guides that can be given to students in paper form or digitally to direct their work. In response to these shortcomings, a number of revisions were made.

First, educative materials for teachers were added to the lessons, including detailed definitions and a clear explanation of the big ideas of the lesson. Second, student worksheets were created to lead students through the tasks to provide pedagogical guidance for the teachers. We also added a section to the teacher materials outlining potential misconceptions and challenges that students may encounter during the lessons. The lessons are currently being taught by teachers in a pilot study with the intention of adding additional supports based on the results.

7.2.2 Equity. Three criteria from the Equity category were identified as inadequate by the reviewers in their evaluation of our Decomposition-by-Sequence lesson: Gives students the opportunity to share their own culture and cultural heritage; Provides opportunities for students to

represent themselves in their projects; and Provides extensions that allow a deeper understanding of topics for students who meet the performance expectations. Specifically, the evaluation team felt that while the students could connect to the soccer theme, the final project did not provide enough of an opportunity for students to represent themselves in their work.

In response to these issues, we redesigned the final activity of the lesson to make it more open-ended and provide more opportunity for customization and learner voice. In the revised version of the activity, learners are able to select the topic of the final animation rather than being constrained to soccer or even sports in general. This less-structured final project provides students with the opportunity to share their culture or cultural heritage if they choose. To address the evaluators' concerns regarding students with exceptionalities and a lack of support for more advanced learners, additional extensions were added to the lessons. This included optional challenges to incorporate sound into the animation (e.g., cheering when the ball goes in) or additional sprites and action movements (e.g., adding a goalie sprite), giving students a chance to explore using blocks that they had not used previously.

In their written comments, the evaluators also raised concerns about the abstract nature of the timeline graphic organizer, which was initially a blank, loosely structured timeline with space for students to document the events in the timeline and the resulting actions of each sprite. Reviewers thought this format may be potentially confusing and the generally text-heavy nature of some of the accompanying worksheets would be difficult for some learners. In response to these concerns, we redesigned the timeline graphic organizer to better scaffold students through the assignment by asking them to circle the correct action for each sprite and providing fill-in-the-blank sentence starters for the events in the timeline that trigger these actions to start or stop (Figure 9(b)).

7.2.3 Content. The reviewers identified six criteria within the Content category that needed improvement: Uses appropriate disciplinary terminology and promotes students' use of disciplinary terminology; Provides opportunities for students to collaborate; Lesson provides opportunities for students to explore and provide solutions to open-ended prompts; Assessments provide feedback on student progress towards a learning objective; Rubrics are based on objectives and standards and assist in measuring student proficiency; and Objective-based formative assessments are presented throughout the module and are included within instruction. In their written comments, the reviewers also expressed concern regarding the emphasis given to the glide and sound blocks over the overall topic of decomposition.

In response to these identified issues, a number of revisions were made to the lesson. First, we tried to decrease the emphasis on the specific blocks and place a greater emphasis on the concept of decomposition. Second, we included an explicit definition for Decomposition-by-Sequence to teachers as part of the lesson. Third, we added an opportunity for students to collaborate while completing an initial decomposition worksheet to encourage collaborative work and provide students with the opportunity to support each other in building their initial understandings of the topic. The previously described changes to the final project also gave students the opportunity to participate in the open-ended exploration that was previously left out of the lesson.

We also significantly changed the assessments within the unit. We implemented an automatic grading feature for the programs students would write and provided a rubric for student projects. The written assessment was revised to emphasize the concept of Decomposition-by-Sequence, again trying to decrease the importance of Scratch-specific blocks. The written assessment was also made to mirror the timeline graphic organizers that students used in the lessons. Finally, journal prompts and discussion questions throughout the unit were reconsidered to increase the usability of these as formative assessments.

7.2.4 Using the Rubric to Inform Curricular Design Discussion. In applying the TEC Rubric to our under-development lesson on Decomposition-by-Sequence, the reviewers were able to identify a number of opportunities to improve the lesson. This included a range of new content, including the addition of opportunities to include students' own ideas and interests in the lessons, further scaffolds for existing activities, and additional pedagogical and content supports for teachers. The fact that the improvements span different aspects of the lesson speaks to one strength of the TEC Rubric; namely, that it helps the designer attend to distinct, yet important, aspects of creating an effective, accessible, and equitable lesson.

Another important thing to note is how, for the most part, the final version of the lesson is not that different from the initial lesson. That is to say, the modifications made in response to the TEC Rubric evaluation were mostly small additions and tweaks, rather than a thorough overhaul. We see this as another strength of the TEC Rubric in that it allows for curricula to be unique and does not try and steer every instructional activity into the same mold following the exact same set of criteria. More so, the TEC Rubric serves as a useful checklist to remind the designer to attend to all these different aspects of a successful lesson.

8 CHALLENGES AND LIMITATIONS OF THE TEC RUBRIC

While this article argues for the utility and benefits of the TEC Rubric, it is not without limitations and challenges. In this section, we acknowledge some known limitations of the rubric and report some challenges we had and were reported by our collaborators while developing and using it. A first limitation that was previously mentioned is the fact that the TEC Rubric is designed explicitly for the curriculum itself and does not consider outside resources, such as professional development or institutional supports. As discussed above in the case of ECS, the professional development is considered an essential part of the curriculum [42], so a teacher or administrator applying the rubric to ECS should be aware of this and take it into consideration. We think the decision to not include outside efforts such as professional development in the initial version of the TEC Rubric is reasonable as, by the time someone is going through a professional development workshop for a specific curriculum, we expect the decision the TEC Rubric is designed to help with has already been made.

A second limitation of the TEC Rubric is that it is designed for introductory computing curricula, so it may not be well suited for other computer science curricular materials. This includes more advanced topics, such as upper-level university courses or potentially even AP Computer Science in the United States, where students are expected to have already taken at least one course and the teacher is expected to have greater mastery of the content. This is not to say the TEC Rubric would not be useful—just that this is not the type of course it was designed for. A second version of this limitation is thinking about courses that try and blend computing or computational thinking with other subjects. For example, a high school biology course that includes modeling and simulation activities that ask students to program would not be an ideal fit for the TEC Rubric given the different classroom context and set of constraints.

A third limitation is in how the structure of the TEC Rubric treats aspects of teaching, learning, and the curricular materials as separate when in reality they are often intertwined. For example, equity and pedagogy are inextricably linked but are evaluated independently in the rubric. This decision is a reflection of the desire to provide some structure to the messy, complex nature of teaching and learning. We currently exploring ways to better handle these types of interdependencies in the rubric.

A final limitation of the TEC Rubric is a result of the challenge of trying to be general enough to apply to a wide variety of curricular materials while also remaining short and concise enough to not be overly burdensome to use. The result of this is a single checkbox to cover what can be

very nuanced and complicated aspects of a curriculum. This oversimplification, while we deem necessary, is an admitted limitation of the rubric in its current form.

Along with limitations, there are also challenges we experienced while applying the rubric. First and foremost was deciding the right portion of a curriculum to evaluate, and if you choose a large set of materials, how to translate the resulting evaluation into the rubric. For example, to evaluate CodeHS, we chose to look at just the first 10 weeks of one of their lessons; we could have just as easily chosen 5 weeks or 40 weeks. The issue with a smaller slice of a curriculum is that the rubric may be overly harsh, as some features or supports may not be present in the selected subsection but present in other parts of the curriculum. At the same time, choosing to evaluate the full 40 weeks also has drawbacks. If a single lesson from a year-long course allows learners to draw on their cultural heritage, then is that adequate to consider that criteria met? As such, deciding on the right duration of the curriculum is important. In our experience, we settled on chunks ranging from 6 to 10 weeks. When designing materials, we found it much easier to apply the rubric to smaller portions of the curriculum (e.g., a single lesson or series of lessons), as it resulted in more direct and actionable feedback.

A second challenge with the rubric is the amount of time it can take to faithfully apply, especially if it is being used to evaluate a longer curriculum. We recommend completing the student assignments as part of the evaluation process. If you are evaluating a 20-week curriculum, that is a lot of assignments to work through. In the end, it is up to the evaluator to decide how much time they can devote to evaluation, recognizing that more time is better but not always realistic.

A final challenge that we are still struggling with is figuring out the best way to translate results from evaluating a curriculum into improved classroom practice. For example, if you are using a highly structured online curriculum that leads students through a series of activities, how and when can or should the teacher customize and deviate from the lesson? Related to this are contexts where students are given the freedom to proceed at their own pace, resulting in students being at different points in the curriculum at the same time. Again, finding opportunities to bring everyone together and then modify instruction can be difficult. As we are in the midst of designing and piloting a curriculum alongside the development of this rubric, understanding how teachers modify and augment our curricular resources to meet the specific needs of their classrooms is an open research question we are actively pursuing.

One final comment about the TEC Rubric is that while what is presented in this article represented the current form of work, we expect it to grow and change as we learn more via feedback from teachers, designers, researchers, and educational decision makers. For example, as our own understanding of how best to support learners with exceptionalities grows, we expect additional criteria to emerge and existing criteria to be refined. In addition, the TEC Rubric could be customized based on the desired unit of analysis. By this, we mean one form of the rubric could be designed to evaluate individual lessons or small sets of lessons and a second for evaluating whole curricula. This may take the form of different criteria or alternative ways of applying the rubric and summarizing the findings based on frequency or some other useful metric. Given the TEC Rubric was developed as part of an ongoing RPP and active research project, we expect the rubric will continue to grow and improve over time. As changes occur, revised forms of the TEC Rubric will be posted on the project website and shared through academic and practitioner-oriented venues.

9 CONCLUSION

The growing interest in and importance of computing has produced a rapid increase in the presence of computing education in K–12 contexts. In response to this growth, new computing curricula are being designed and advertised as meeting the needs of teachers and students. However, to date, few resources exist to help educational decision makers evaluate different curricular

opportunities. This challenge is especially difficult for the field of computing, as there are relatively few comprehensive frameworks or guidelines for making these decisions, and few decision makers have deep knowledge of the field. Further, given issues with equity and underrepresentation in computing-related fields, it is essential that the curricular decisions made attend to both content mastery as well as issues of equity and accessibility to not perpetuate these issues. At the same time, the responsibility to create effective and accessible educational materials falls on the curriculum designers, who likewise lack clear guidelines or evaluation materials to guide the creation of materials to address these issues.

In response to this challenge, this article introduces the Teacher Accessibility, Equity, and Content (TEC) Rubric. The TEC Rubric is intended to serve as a resource to help educators, educational decision makers, and designers make informed decisions with respect to creating effective, accessible, and equitable learning opportunities. More concretely, the TEC Rubric can be used as an instrument to evaluate existing curricula with respect to three critical dimensions to ensure equitable and effective instruction. The TEC Rubric can also be useful for those who are designing new instructional materials and learning environments by serving as a resource to ensure the created artifacts attend to critical issues related to teacher accessibility, equity, and computing content. To demonstrate these two roles, this article presented the results of the analysis of two widely used and successful curricula to show how it can highlight their strengths while also identifying potential opportunities for instructors to modify their instruction or introduce supplemental materials to improve classroom outcomes. Second, the article provided an example of how the TEC Rubric helped refine and improve the creation of new curricular materials. In this case, the TEC Rubric directed the designers' attention towards missed opportunities, resulting in additional teacher materials, new opportunities for learners to incorporate their own interests and ideas into the project, and provide clearer instructions and better scaffolds in instructional materials.

As computer science and the big ideas of computing continue to make their way into K–12 classrooms around the world, providing supports and guidance for the teachers and designers tasked with carrying out this goal is essential. In developing and sharing the TEC Rubric, we seek to provide a resource that is useful to both researchers working on the computing education space as well as K–12 educators and decision makers charged with bringing computing to all learners. In doing so, this work bridges the researcher-practitioner gap and delivers on the promise of RPPs as a mutually supportive and impactful approach to conducting educational research. Our hope with this work is that the TEC Rubric serves as a useful resource in the toolbelt of educators, educational decision makers, and designers to help them be successful in their quest to introduce today's learners to the field of computing in an effective and equitable way.

ACKNOWLEDGMENTS

We would like to thank our district partners who played a significant role in shaping the rubric and providing invaluable insight throughout the process, especially Andy Rasmussen, Melissa Cobian, and Kristan Beck. We would also like to acknowledge the undergraduate research assistants who helped with this project: Connor Hopcraft, Woorin Jang, Ashley Wang, Max White, and Ruolin Zheng.

REFERENCES

[1] AchieveNJ. 2016. The New Jersey Student Learning Standards for Science. https://www.state.nj.us/education/aps/cccs/science/ScienceInstructionCompanion.pdf.

[2] Joint Task Force on Computing Curricula, Association for Computing Machinery (ACM) and IEEE Computer Society. 2013. *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science.* ACM, New York, NY, USA.

[3] H. G. Andrade. 2005. Teaching with rubrics: The good, the bad, and the ugly. *Coll. Teach.* 53, 1 (2005), 27–31.

[4] C. Ashcraft et al. 2016. *Women in Tech: The Facts*. National Center for Women & Technology (NCWIT). https://www.ncwit.org/sites/default/files/resources/ncwit_women-in-it_2016-full-report_final-web06012016.pdf.

[5] O. Astrachan et al. 2011. The CS10K project: Mobilizing the community to transform high school computing. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*. 85–86.

[6] T. Bell et al. 2010. Computer science in New Zealand high schools. In *Proceedings of the 12th Australasian Conference on Computing Education-Volume 103*. 15–22.

[7] S. Bocconi, A. Chioccariello, G. Dettori, A. Ferrari, and K. Engelhardt. 2016. Developing computational thinking in compulsory education – Implications for policy and practice. EUR 28295 EN. DOI: 10.2791/792158

[8] H. Borko. 2004. Professional development and teacher learning: Mapping the terrain. *Educ. Res.* 33, 8 (2004), 3–15. DOI: https://doi.org/10.3102/0013189X033008003

[9] C. Brady et al. 2016. All roads lead to computing: Making, participatory simulations, and social computing as pathways to computer science. *IEEE Trans. Educ.* 60, 99 (2016), 1–8. DOI: https://doi.org/10.1109/TE.2016.2622680

[10] K. Brennan and M. Resnick. 2012. New frameworks for studying and assessing the development of computational thinking. American Educational Research Association, Vancouver, Canada. http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf.

[11] D. J. Briars. 2014. Curriculum materials matter: Evaluating the evaluation process. *National Council of Teachers of Mathematics*. (2014). https://www.nctm.org/News-and-Calendar/Messages-from-the-President/Archive/Diane-Briars/Curriculum-Materials-Matter_-Evaluating-the-Evaluation-Process/.

[12] L. Buechley et al. 2008. The LilyPad Arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 423–432.

[13] R. Bybee and C. Chopyak. 2017. *Instructional Materials and Implementation of NGSS: Demand, Supply, and Strategic Opportunities*. A report for the Carnegie Corporation of New York. https://www.carnegie.org/media/filer_public/6b/57/6b57c995-e6c7-485d-ad17-dcd807a19926/ngss_report_carnegie_corp_72017.pdf.

[14] V. Cateté et al. 2016. Developing a rubric for a creative CS principles lab. In *Proceedings of the ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE'16)*. 290–295.

[15] C. E. Coburn, W. R. Penuel, and K. E. Geil. 2013. *Research-Practice Partnerships: A Strategy for Leveraging Research for Educational Improvement in School Districts*. William T. Grant Foundation, New York, NY.

[16] CodeHS. 2019. https://codehs.com/.

[17] Computer Science Teaching Rubric. 2019. https://sites.google.com/sfusd.edu/csplc/resources/teaching-rubric.

[18] T. Crick and S. Sentance. 2011. Computing at school: Stimulating computing education in the UK. In *Proceedings of the 11th Koli Calling International Conference on Computing Education Research – (Koli Calling'11)*. 122.

[19] CSTA Standards Task Force. 2016. *CSTA K–12 Computer Science Standards (Revised 2016)*. Association for Computing Machinery. https://drive.google.com/open?id=1-dPTAI1yk2HYPKUWZ6DqaM6aVUDa9iby.

[20] Computer Science Teachers Association. 2017. CSTA K-12 Computer Science Standards, Revised 2017. Retrieved from http://www.csteachers.org/standards.

[21] J. Cuny. 2011. Transforming computer science education in high schools. *Computer* 44, 6 (2011), 107–109. DOI: https://doi.org/10.1109/MC.2011.191

[22] J. Cuny. 2015. Transforming K–12 computing education: An update and a call to action. *ACM Inroads* 6, 3 (2015), 54–57. DOI: https://doi.org/10.1145/2809795

[23] L. Darling-Hammond. 2000. Teacher quality and student achievement: A review of state policy evidence. *Educ. Pol. Anal. Arch.* 8, 1 (2000), 44.

[24] L. A. Delyser et al. 2018. Priming the computer science teacher pump. (2018), 62. https://drive.google.com/file/d/1DXgpLjl_k87TVpQ0cLusfdjnYySIgIjT/view?usp=sharing.

[25] P. J. Denning et al. 1989. Computing as a discipline. *Computer* 22, 2 (1989), 63–70. DOI: https://doi.org/10.1109/2.19833

[26] B. DiSalvo et al. 2014. Saving face while geeking out: Video game testing as a justification for learning computer science. *J. Learn. Sci.* 23, 3 (2014), 272–315. DOI: https://doi.org/10.1080/10508406.2014.893434

[27] K. Donathan and P. Tymann. 2010. The development and use of scoring rubrics:(or how to grade thousands of exams without losing your mind). In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*. 477–477.

[28] H. Dwyer et al. 2014. Identifying elementary students' pre-instructional ability to develop algorithms and step-by-step instructions. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*. 511–516.

[29] R. Eglash et al. 2013. Toward culturally responsive computing education. *Commun. ACM.* 56, 7 (2013), 33–36. DOI: https://doi.org/10.1145/2483852.2483864

[30] EQuIP. 2013. EQuIP Rubric for Lessons & Units: ELA/Literacy (Grades 3–5) and ELA (Grades 6–12). http://www.achieve.org/files/EQuIP-ELArubric-06-24-13-FINAL.pdf.

[31] B. Ericson et al. 2016. State-based progress towards computer science for all. *ACM Inroads* 7, 4 (2016), 57–60. DOI: https://doi.org/10.1145/2994607
[32] C. Fancsali, L. Tigani, P. Toro Isaza, and R. Cole. 2018. A landscape study of computer science education in NYC: early findings and implications for policy and practice. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. ACM, New York, NY, USA, 44–49.
[33] D. A. Fields et al. 2018. Putting making into high school computer science classrooms: Promoting equity in teaching and learning with electronic textiles in exploring computer science. *Equity Excell. Educ.* 51, 1 (Jan. 2018), 21–35. DOI: https://doi.org/10.1080/10665684.2018.1436998
[34] S. Fincher et al. 2010. Repositories of teaching material and communities of use: Nifty assignments and the greenroom. In *Proceedings of the 6th International Workshop on Computing Education Research*. 8.
[35] D. Franklin et al. 2011. Animal Tlatoque: Attracting middle school students to computing through culturally relevant themes. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*. 453–458.
[36] D. Franklin et al. 2020. Scratch encore: The design and pilot of a culturally relevant intermediate scratch curriculum. In *Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education*.
[37] D. Franklin et al. 2017. Using upper-elementary student performance to understand conceptual sequencing in a blocks-based curriculum. In *Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education*. 231–236.
[38] M. G. Fullan. 1993. Why teachers must become change agents. *Educ. Leader.* 50, 6 (1993), 13.
[39] S. Furber. 2012. Shut down or restart? The way forward for computing in UK schools. *The Royal Society, London.* (2012). https://royalsociety.org/~/media/education/computing-in-schools/2012-01-12-computing-in-schools.pdf.
[40] J. Gal-Ezer and C. Stephenson. 2010. Computer science teacher preparation is critical. *ACM Inroads.* 1, 1 (2010), 61. DOI: https://doi.org/10.1145/1721933.1721953
[41] G. Gay. 2014. Culturally responsive teaching principles, practices, and effects. *Handbook of Urban Education*. Routledge, 353–372.
[42] J. Goode et al. 2012. Beyond curriculum: The exploring computer science program. *ACM Inroads* 3, 2 (2012), 47–53.
[43] J. Goode. 2010. Connecting K–16 curriculum & policy: Making computer science engaging, accessible, and hospitable for underrepresented students. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*. 22–26.
[44] J. Goode et al. 2014. Curriculum is not enough: The educational theory and research foundation of the exploring computer science professional development model. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education (SIGCSE'14)*. 493–498.
[45] J. Goode. 2007. If you build teachers, will students come? The role of teachers in broadening computer science learning for urban youth. *J. Educ. Comput. Res.* 36, 1 (2007), 65–88. DOI: https://doi.org/10.2190/2102-5G77-QL77-5506
[46] Google and Gallup. 2015. *Images of Computer Science: Perceptions Among Students, Parents and Educators in the U.S.* https://services.google.com/fh/files/misc/images-of-computer-science-report.pdf.
[47] Google Inc. and Gallup Inc. 2016. *Trends in the State of Computer Science in U.S. K–12 Schools.* https://services.google.com/fh/files/misc/trends-in-the-state-of-computer-science-report.pdf.
[48] S. Grover and S. Basu. 2017. Measuring student learning in introductory block-based programming: Examining misconceptions of loops, variables, and Boolean logic. In *Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education*. 267–272.
[49] S. Grover and R. Pea. 2013. Computational thinking in K–12: A review of the state of the field. *Educ. Res.* 42, 1 (2013), 38–43.
[50] M. Guzdial et al. 2014. Georgia computes! An intervention in a US state, with formal and informal education in a policy context. *ACM Trans. Comput. Educ.* 14, 2 (2014), 1–29. DOI: https://doi.org/10.1145/2602488
[51] O. Hazzan et al. 2008. A model for high school computer science education: The four key elements that make it! *ACM SIGCSE Bull.* 40, 1 (2008), 281–285.
[52] E. Henrick, S. McGee, R. Greenberg, L. Dettori, A. Rasmussen, D. Yanek, and D. Reed. 2019. Assessing the effectiveness of computer science RPPs: The case of CAFE CS. In *Proceedings of the 2019 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT'19)*.
[53] N. Holbert. 2016. Leveraging cultural values and "ways of knowing" to increase diversity in maker activities. *Int. J. Child-Comput. Interact.* 9–10 (2016), 33–39. DOI: https://doi.org/10.1016/j.ijcci.2016.10.002
[54] D. S. Hong and K. M. Choi. 2014. A comparison of Korean and American secondary school textbooks: The case of quadratic equations. *Edu. Stud. Math.* 85, 2 (2014), 241–263.
[55] L. Hoogstra et al. 2011. *Wisconsin CREATE Initiative: Year 2 Evaluation.* Learning Point Associates. https://www.air.org/sites/default/files/downloads/report/CREATE_Year_2_Evaluation_Report-September_2011_FINAL_0.pdf.
[56] M. Israel et al. 2018. School-embedded and district-wide instructional coaching in K–8 computer science: Implications for including students with disabilities. *J. Technol. Teach. Educ.* 26, 3 (2018), 471–501.

[57] M. Israel et al. 2014. Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers and Education* 82 (2014), 263–279. DOI: https://doi.org/10.1016/j.compedu.2014.11.022
[58] S. P. Jones et al. 2011. Computing at school. *Int. Compar.* 7 (2011).
[59] A. Jonsson. 2014. Rubrics as a way of providing transparency in assessment. *Assess. Eval. High. Educ.* 39, 7 (2014), 840–852.
[60] A. Jonsson and G. Svingby. 2007. The use of scoring rubrics: Reliability, validity and educational consequences. *Educ. Res. Rev.* 2, 2 (2007), 130–144.
[61] K–12 Computer Science Framework. 2016. http://www.k12cs.org.
[62] M. Koehler and P. Mishra. 2009. What is technological pedagogical content knowledge (TPACK)? *Contemp. Iss. Technol. Teach. Educ.* 9, 1 (2009), 60–70.
[63] M. Lachney. 2017. Computational communities: African-American cultural capital in computer science education. *Comput. Sci. Educ.* 27, 3–4 (2017), 175–196.
[64] R. E. Ladner and M. Israel. 2016. "For all" in "computer science for all." *Commun. ACM.* 59, 9 (2016), 26–28. DOI: https://doi.org/10.1145/2971329
[65] J. R. Landis and G. G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics* 33, 1 (1977), 159–174.
[66] M. Leake and C. M. Lewis. 2017. Recommendations for designing CS resource sharing sites for all teachers. In *Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE'17)*. 357–362.
[67] I. Lee et al. 2014. Integrating computational thinking across the K–8 curriculum. *ACM Inroads* 5, 4 (2014), 64–71.
[68] N. Liberman et al. 2012. "Regressed experts" as a new state in teachers' professional development: Lessons from computer science teachers' adjustments to substantial changes in the curriculum. *Comput. Sci. Educ.* 22, 3 (2012), 257–283. DOI: https://doi.org/10.1080/08993408.2012.721663
[69] R. B. Lindsey et al. 2008. *Culturally Proficient Inquiry: A Lens for Identifying and Examining Educational Gaps.* Corwin Press.
[70] T. R. Loveland. 2005. Writing standards-based rubrics for technology education classrooms: The use of rubrics goes beyond the simple need for objective grading in classrooms. *Technol. Teach.* 65, 2 (2005), 19–23.
[71] J. H. Maloney et al. 2008. Programming by choice: Urban youth learning programming with scratch. *ACM SIGCSE Bull.* 40, 1 (2008), 367–371.
[72] J. Margolis. 2008. *Stuck in the Shallow End: Education, Race, and Computing*. The MIT Press.
[73] J. Margolis and A. Fisher. 2003. *Unlocking the Clubhouse: Women in Computing*. The MIT Press.
[74] J. C. Marshall et al. 2010. The design and validation of EQUIP: An instrument to assess inquiry-based instruction. *Int. J. Sci. Math. Educ.* 8, 2 (2010), 299–321. DOI: https://doi.org/10.1007/s10763-009-9174-y
[75] M. McDonald and G. McDonald. 1999. Computer science curriculum assessment. *ACM SIGCSE Bull.* 31, 1 (1999), 194–197.
[76] S. M. Mitchell and W. G. Lutters. 2006. Assessing the value of computer science course material repositories. In *Proceedings of the 19th Conference on Software Engineering Education and Training Workshops (CSEETW'06)*. 2–2.
[77] L. C. Moll et al. 1992. Funds of knowledge for teaching: Using a qualitative approach to connect homes and classrooms. *Theor. Prac.* 31, 2 (1992), 132–141.
[78] B. M. Moskal and J. A. Leydens. 2000. Scoring rubric development: Validity and reliability. *Pract. Assess., Res. Eval.* 7, 10 (2000), 71–81.
[79] National Academies of Sciences, Engineering, and Medicine. 2018. *Design, Selection, and Implementation of Instructional Materials for the Next Generation Science Standards: Proceedings of a Workshop*. The National Academies Press, Washington, DC. https://doi.org/10.17226/25001
[80] National Commission on Teaching and America's Future. 1996. *What Matters Most: Teaching for America's Future.* Report of the National Commission on Teaching & America's Future. National Commission on Teaching and America's Future, Woodbridge, VA.
[81] L. Ni and M. Guzdial. 2012. Who AM I?: Understanding high school computer science teachers' professional identity. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education (SIGCSE'12)*. 499.
[82] M. Ong. 2011. The status of women of color in computer science. *Commun. ACM* 54, 7 (2011), 32–34. DOI: https://doi.org/10.1145/1965724.1965737
[83] E. Panadero and A. Jonsson. 2013. The use of scoring rubrics for formative assessment purposes revisited: A review. *Educ. Res. Rev.* 9 (2013), 129–144.
[84] E. Panadero and M. Romero. 2014. To rubric or not to rubric? The effects of self-assessment on self-regulation, performance and self-efficacy. *Assess. Educ.: Princ., Polic. Pract.* 21, 2 (2014), 133–148.
[85] W. R. Penuel and D. J. Gallagher. 2017. *Creating Research Practice Partnerships in Education.* Harvard Education Press.

[86] N. Ragonis et al. 2010. A survey of computer science teacher preparation programs in Israel tells US: Computer science deserves a designated high school teacher preparation! In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*. 401–405.
[87] R. Raman et al. 2015. Computer science (CS) education in Indian schools: Situation analysis using Darmstadt model. *ACM Trans. Comput. Educ.* 15, 2 (2015), 1–36. DOI: https://doi.org/10.1145/2716325
[88] M. Resnick et al. 2009. Scratch: Programming for all. *Commun. ACM* 52, 11 (2009), 60.
[89] K. M. Rich et al. 2019. A K–8 debugging learning trajectory derived from research literature. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. 745–751.
[90] K. M. Rich et al. 2018. Decomposition: A K–8 computational thinking learning trajectory. In *Proceedings of the ACM Conference on International Computing Education Research (ICER'18)*. 124–132.
[91] K. M. Rich et al. 2017. K–8 learning trajectories derived from research literature: Sequence, repetition, conditionals. In *Proceedings of the ACM Conference on International Computing Education Research*. 182–190.
[92] R. Roque et al. 2016. Supporting diverse and creative collaboration in the scratch online community. *Mass Collaboration and Education*. Springer, 241–256.
[93] D. Rose. 2000. Universal design for learning. *J. Spec. Educ. Technol.* 15, 4 (2000), 47–51.
[94] D. H. Rose and A. Meyer. 2002. *Teaching Every Student in the Digital Age: Universal Design for Learning*.
[95] J. J. Ryoo et al. 2013. Democratizing computer science knowledge: Transforming the face of computer science through public high school education. *Learn., Media Technol.* 38, 2 (2013), 161–181. DOI: https://doi.org/10.1080/17439884.2013.756514
[96] L. Seiter and B. Foreman. 2013. Modeling the learning progressions of computational thinking of primary grade students. In *Proceedings of the 9th International ACM Conference on International Computing Education Research*. 59–66.
[97] S. Sentance et al. 2012. Grand challenges for the UK: Upskilling teachers to teach computer science within the secondary curriculum. In *Proceedings of the 7th Workshop on Primary and Secondary Computing Education– (WiPSCE'12)*. 82.
[98] S. Sentance and A. Csizmadia. 2017. Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Educ. Inform. Technol.* 22, 2 (2017), 469–495. DOI: https://doi.org/10.1007/s10639-016-9482-0
[99] A. Settle et al. 2012. Infusing computational thinking into the middle- and high-school curriculum. In *Proceedings of the 17th ACM Conference on Innovation and Technology in Computer Science Education*. 22–27.
[100] L. S. Shulman. 1986. Those who understand: Knowledge growth in teaching. *Educ. Res.* 15, 2 (1986), 4–14. DOI:https://doi.org/10.3102/0013189X015002004
[101] M. Singer and J. Tuomi. 1999. *Selecting Instructional Materials*. National Academies Press.
[102] D. D. Stevens and A. J. Levi. 2013. *Introduction to Rubrics: An Assessment Tool to Save Grading Time, Convey Effective Feedback, and Promote Student Learning*. Stylus Publishing, LLC.
[103] The College Board. 2018. *About AP Scores*. https://apstudents.collegeboard.org/about-ap-scores.
[104] The College Board. 2017. *AP National Summary Report*. https://research.collegeboard.org/programs/ap/data.
[105] The Framework for Teaching Evaluation Instrument. 2013. https://danielsongroup.org/framework/framework-teaching.
[106] The SCRIPT: Strategic Planning Tool for School Districts. 2018. https://www.csforall.org/projects_and_programs/script/.
[107] D. Thompson et al. 2013. The role of teachers in implementing curriculum changes. In *Proceedings of the 44th ACM Technical Symposium on Computer Science Education (SIGCSE'13)*. 245.
[108] J. Wang, S. H. Moghadam, and J. Tiffany-Morales. 2017. Social perceptions in computer science and implications for diverse students. In *Proceedings of the 2017 ACM Conference on International Computing Education Research (ICER'17)*. ACM, New York, NY, USA, 47–55. DOI: https://doi.org/10.1145/3105726.3106175
[109] E. Wheeler, J. Wachen, A. Rasmussen, D. Franklin, and D. Weintrop. 2020. Introducing computer science into K-8 classrooms: Teachers' perspectives from a large, urban school district. In *Proceedings of the 2020 ACM SIGCSE Technical Symposium on Computer Science Education*. ACM, New York, NY, USA. DOI: https://doi.org/10.1145/3328778.3372612
[110] V. White et al. 2018. Illuminating the computing pathway for girls in Mississippi. In *Proceedings of the ASEE Conference and Exposition*.
[111] C. Wilson et al. 2010. *Running on Empty: The Failure to Teach K–12 Computer Science in the Digital Age*. The Association for Computing Machinery and the Computer Science Teachers Association.
[112] J. M. Wing. 2006. Computational thinking. *Commun. ACM* 49, 3 (2006), 33–35.
[113] A. Yadav et al. 2014. Computational thinking in elementary and secondary teacher education. *ACM Trans. Comput. Educ.* 14, 1 (2014), 5.

[114] A. Yadav et al. 2016. Expanding computer science education in schools: understanding teacher experiences and challenges. *Comput. Sci. Educ.* 26, 4 (2016), 235–254. DOI: https://doi.org/10.1080/08993408.2016.1257418
[115] M. Yuan and M. Recker. 2015. Not all rubrics are equal: A review of rubrics for evaluating the quality of open educational resources. *Int. Rev. Res. Open Distrib. Learn.* 16, 5 (2015). DOI: https://doi.org/10.19173/irrodl.v16i5.2389
[116] S. Zweben and B. Bizot. 2018. 2017 Taulbee survey. *Comput. Res. News* 30, 5 (2018), 47.
[117] Achieve and National Science Teachers Association. 2016. EQuIP Rubric for Lessons & Units: Science Version 3.0. https://www.nextgenscience.org/resources/equip-rubric-science.

Received May 2019; revised October 2019; accepted October 2019