frontiers in Education

METHODS
published: 25 June 2021
doi: 10.3389/feduc.2021.695376

Utilizing Evidence-Centered Design to Develop Assessments: A High School Introductory Computer Science Course

Sunni Newton[1]*, *Meltem Alemdar*[1], *Daisy Rutstein*[2], *Doug Edwards*[1], *Michael Helms*[1], *Diley Hernandez*[1] *and Marion Usselman*[1]

[1]Center for Education Integrating Science, Mathematics, & Computing (CEISMC), Georgia Institute of Technology, Atlanta, GA, United States, [2]SRI International, Menlo Park, CA, United States

Evidence-centered design (ECD) is an assessment framework tailored to provide structure and rigor to the assessment development process, and also to generate evidence of assessment validity by tightly coupling assessment tasks with focal knowledge, skills, and abilities (FKSAs). This framework is particularly well-suited to FKSAs that are complex and multi-part (Mislevy and Haertel, 2006), as is the case with much of the focal content within the computer science (CS) domain. This paper presents an applied case of ECD used to guide assessment development in the context of a redesigned introductory CS curriculum. In order to measure student learning of CS skills and content taught through the curriculum, knowledge assessments were written and piloted. The use of ECD provided an organizational framework for assessment development efforts, offering assessment developers a clear set of steps with accompanying documentation and decision points, as well as providing robust validity evidence for the assessment. The description of an application of ECD for assessment development within the context of an introductory CS course illustrates its utility and effectiveness, and also provides a guide for researchers carrying out related work.

Keywords: evidence-centered design, item development, K-12 education, computer science, assessment

Edited by:
*Robbert Smit,
St.Gallen University of Teacher Education, Switzerland*

Reviewed by:
*Yuning Xu,
Kidaptive Inc., United States
Jonathan Schmidgall,
Educational Testing Service, United States*

***Correspondence:**
*Sunni Newton
sunni.newton@ceismc.gatech.edu*

Specialty section:
This article was submitted to Assessment, Testing and Applied Measurement, a section of the journal Frontiers in Education

Received: *14 April 2021*
Accepted: *14 June 2021*
Published: *25 June 2021*

Citation:
Newton S, Alemdar M, Rutstein D, Edwards D, Helms M, Hernandez D and Usselman M (2021) Utilizing Evidence-Centered Design to Develop Assessments: A High School Introductory Computer Science Course. Front. Educ. 6:695376. doi: 10.3389/feduc.2021.695376

INTRODUCTION

Assessment of student learning is an essential part of K-12 education. Assessments, if well designed and used appropriately, have been shown to improve student achievement (Cizek, 2010). Formative and summative assessments are common tools that teachers use to assess student learning of new material and knowledge of state standards (Dixson and Worrell, 2016). Teachers struggle to develop high-quality assessments due to many challenges such as lack of time, lack of clarity around defining knowledge domains, and difficulties in setting parameters of acceptable student performance (Khattri et al., 1995). Further, assessment in interdisciplinary subjects such as computer science (CS) is especially challenging since there is no consensus around the best methods to measure student performance (So et al., 2020). Over the last decade, there have been substantial international efforts to expand CS instruction from being primarily a university level discipline to widespread adoption at the K-12 level. As a result, an increasing number of countries have focused on developing K-12 CS curriculum materials (Heintz et al., 2016). The United States, through its work to develop a national level Framework for K-12 Computer Science Education (K-12 CS Framework Committee, 2016), has emphasized the importance of exposure to CS and computational thinking skills, not just for students intending to pursue CS careers, but for all students as a general educational imperative for their future (Heintz et al., 2016).

This increased development and adaptation of K-12 CS curricula in the U.S. over the past decade has included: creating the K-12 CS Framework (K-12 CS Framework Committee, 2016), development of semester long or full year standalone CS courses in high school, integrating CS notions and practices into K-12 mathematics and science curricula (Sengupta et al., 2013), and the development of Advanced Placement courses (Astrachan and Briggs, 2012; Arpaci-Dusseau et al., 2013). As CS curricula continue to make their way into K-12 schools, the issue of assessing student learning remains a challenge (Denning, 2017). With the development of new curriculum and interventions comes the need for assessments that can measure CS concepts and practices.

In their recent white paper on improving CS instruction by promoting teachers' skills in CS formative assessment literacy, Basu et al. (2021) note that CS teachers, especially those without a formal CS background, struggle with various aspects of assessment, including writing appropriate items, accessing useful assessment tools, and distilling CS content into well-articulated and measurable learning goals. Assessments are needed to aid teachers in determining what their students know and can do as well as to help determine the strengths and weaknesses of different curriculum interventions. These needs span both formative assessments, which provide frequent, immediate feedback to teachers during the course of instruction intended to inform their teaching and learning as it unfolds, as well as summative assessments, which are typically cumulative in nature, given at the end of a unit of instruction, and provide an overall gauge of what students know and are able to do (Cizek, 2010).

Revised Course Overview

In 2016, the National Science Foundation funded the development of curriculum to revise a year-long introductory level high school CS course. This redesigned course, as well as many other K12 CS educational interventions, seeks to affect distal outcomes of degree earning and career pursuit by tackling the most proximal CS-related juncture for high school students: the first course in the CS pathway at the high school level within the state where this research took place. The key aspect of the redesign aligned with widening its appeal and personal relevance to all students was the use of an inquiry-driven, problem-based learning approach accompanied by the inclusion of culturally authentic practices. Students capitalize on their personal experiences and interests by selecting a problem relevant to them that serves as the focus for their work on four large projects throughout the course: a narrated PowerPoint presentation, a website, music to accompany the website, and an app-based game. The narrative continuity and interconnectedness of these projects, all centered around a single topic of the students' choosing, is intended to increase student engagement and interest while covering the same technical content included in the original course. This course is taught across our state and serves as the first course in the computer science course sequence, also referred to as the computer science pathway. The course is primarily intended for 9th graders, but students in 10th–12th grades can and do take the course as well. It is appropriate for a wide audience of students at various ability levels. Student performance and interest level in this course often serves as a driver of or impediment to pursual of later computer science courses, including the highly valuable Advanced Placement (AP) computer science courses. The overarching goal of the project for which this course was developed is to increase the presence of girls and under-represented minority students in later courses within the CS pathway, especially AP CS courses.

The curriculum focuses on students' voice and choice primarily through allowing students to select a problem of interest that will be the focus of four digital artifacts to be created throughout the school year. Problem selection is of critical importance, as students are intended to work on this single topic throughout the year-long course. Students are encouraged to select a problem of personal interest to them that might also relate to something in their background or experiences. Selecting a problem of interest with the appropriate scope that is likely to retain student interest over a long period of time is critical; teachers need to provide guidance with this process to optimize topic selection. The problem needs to be complex enough that there is sufficient depth to support creation of four digital artifacts, yet the problem cannot be so complex and large in scope that students are unable to fully explore the problem area and envision possible means of addressing the problem or raising awareness about the problem. Sample topics selected by student groups include anxiety, sleep deprivation, obesity, and school shootings.

In Unit 1, students create a narrated PowerPoint presentation describing and exploring their selected problem using the Microsoft Word PowerPoint software. The key aim of this PowerPoint Presentation is to inform the audience about the selected problem. The standards covered in this unit are largely general and not highly technical in nature. Students work on the learning goal "use of online resources and technology" by carrying out internet searches to gather information about their topic for their websites; teachers provide instruction on how to use search engines and how to determine whether information on the internet is likely to be reliable. The learning goals related to communication skills and use of presentation software are also addressed as students work on how to best present their ideas and compile the information they have found in a PowerPoint presentation.

In Unit 2, students work on website development, again related to their focal problem, using the Google sites platform. The intended aim of this website is to raise awareness about the problem and motivate users to work to address the problem. The Unit 2 learning goals of design principles, webpage design, and site usability are addressed via instruction on how various elements of websites impact their messaging and usability, as well as instruction on optimizing website layout and organization. Students then put this instruction into action as they work in groups to create and refine websites around their focal topics.

Unit 3 focuses on computer programming, taught through the music creation platform EarSketch. Students learn about algorithms and programming by using these skills to program two pieces of music. The goal of the music is to enhance engagement with the computational artifacts that have already been created and include: a short musical introduction to the PowerPoint presentation the developed in Unit 1, and a longer piece of background music to be added to the website they created in Unit 2. Lastly, in Unit 4, students study game design and learn additional programming concepts that they use to program a simple game to be played in an app using the program App Inventor. The goal of the game is to raise awareness about their focal topic and students are told they must include a subset of the programming concepts they have learned.

Assessment Context

Similar to other K-12 CS education initiatives, one of the aims of this project is to understand the context-specific relationships between the introductory CS course and student learning. Specifically, as we attempted to increase student engagement and interest through the course redesign and its emphasis on project-based learning and cultural relevance, we needed content assessments to determine the extent to which the core technical content and skills were being adequately conveyed to students through the new curriculum. We sought to determine how well the course supported student learning across learners. When reviewing current assessments, none were found that met all of the objectives of the curriculum. To ensure that the curriculum was aligned with instruction it was important to develop and pilot a new knowledge assessment that would measure how well students were able to engage with the constructs highlighted in the curriculum.

Assessment of CS content at the K-12 level is a relatively new endeavor, given that the broadened inclusion of CS in general K-12 instructional pathways is a recent development. Efforts to create high-quality K-12 CS assessments have been plagued by several issues, including the field's difficulties in reaching consensus on the definition and scope of the various components of and terminology within the CS educational landscape as well as the fact that many skills and ideas in CS are broad, complex, and multi-part (Webb et al., 2017; Tang et al., 2020). Muddiness around the construct definitions and learning progressions can yield assessments that focus on surface-level knowledge rather than deeper application of skills and practices (Denning, 2017), and/or assessments that primarily reflect specifics of a given course rather than more general, course-agnostic knowledge and skills (Snow et al., 2017).

Accurately and clearly articulating and operationalizing the content to be measured lies at the heart of a successful assessment effort; this piece has presented challenges for assessment developers in the CS space (Denning, 2017; Tang et al., 2020). In an attempt to optimize this key piece of CS assessment, we adopted the Evidence-Centered Design (ECD) framework to guide our assessment development efforts. ECD has been used effectively for assessment development in various fields (Grover and Basu, 2017; Hu et al., 2017; Chapelle et al., 2018; Bechard et al., 2019; Oliveri et al., 2019; Snow et al., 2019); it is especially useful for starting assessment development off on the right foot by virtue of its rigorous and scaffolded methodology for clearly defining the content to be measured and what it means for students to engage with this content (Mislevy and Riconscente, 2006). The ECD framework at its core helps assessment developers articulate exactly what is to be measured and what evidence is needed to measure it and provides support to build the argument that the assessment meets its specified purpose (Mislevy, 2007). In adopting ECD for this project, it was our intention to focus time and effort on the content definition phase, thus strengthening the foundation for the assessments to be developed.

The intended use of the newly developed assessments is to serve as summative assessments administered at the end of the corresponding unit and to provide a measure of the extent to which students have mastered the content taught in each unit. Teachers then use the scores from the assessments, along with additional assessment opportunities given during the course, to determine an overall grade for the student. These end-of-unit assessments are not designed to cover every concept in detail, but instead to focus on specific constructs that are critical to students' understanding. The design process is discussed further below.

This paper describes how an ECD approach was used to guide assessment development within the context of a high school introductory CS course. The following sections describe our iterative efforts to develop content assessments in order to provide guidance for researchers. The purpose of this paper is to describe an applied example of ECD utilization; namely, our use of ECD to guide assessment development within the context of a high school introductory CS course.

Theoretical Framework: Evidence-Centered Design

As mentioned above, all the assessments were developed for these units using an ECD approach. ECD is a framework that supports the validity argument for an assessment by highlighting the relationship between the claims made based on the results of the assessment and the evidence needed to support those claims (Mislevy and Haertel, 2006; Mislevy et al., 2003). The framework focuses the developer on three models: the student model (what inferences we want to make about the student), the evidence model (how evidence collected from the student provides information for the student model), and the task model (how the task can be structured to allow students to provide evidence for use in the evidence model), as well as the connections among these models. This is often referred to as the assessment argument because the relationship of the resulting tasks to the developed models can provide support for the validity of the assessment (Mislevy, 2007).

ECD is often considered a layered approach (see **Figure 1**) where details of the assessment increase in specificity as you move through the layers. The first layer, the domain analysis layer, explores the domain of interest. The focus is on gathering information on the important aspects of the domain, the relationship between these aspects, and representations of these aspects. This layer can include identifying standards,

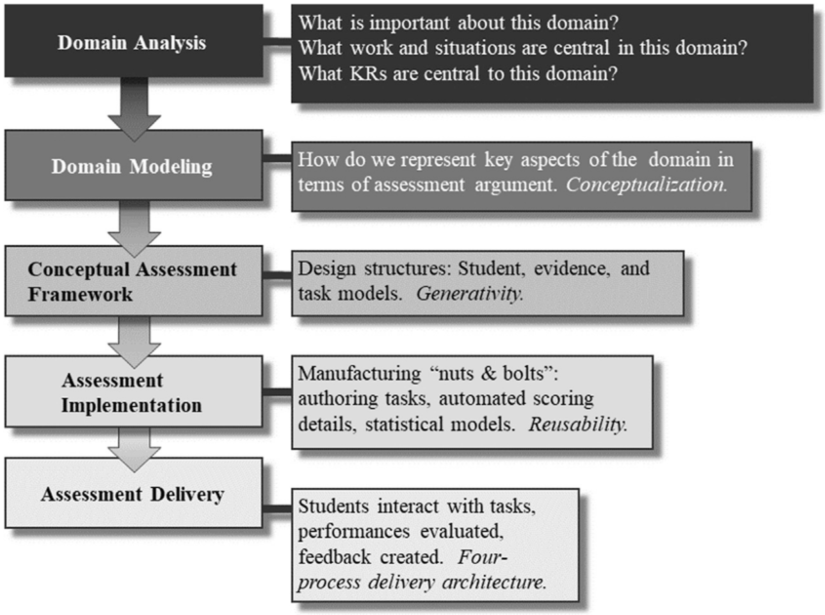

Domain Analysis
What is important about this domain?
What work and situations are central in this domain?
What KRs are central to this domain?
Domain Modeling
How do we represent key aspects of the domain in terms of assessment argument. *Conceptualization.*
Conceptual Assessment Framework
Design structures: Student, evidence, and task models. *Generativity.*
Assessment Implementation
Manufacturing "nuts & bolts": authoring tasks, automated scoring details, statistical models. *Reusability.*
Assessment Delivery
Students interact with tasks, performances evaluated, feedback created. *Four-process delivery architecture.*

FIGURE 1 | Layers of ECD from Mislevy and Riconscente, 2006.

defining learning sequences and grouping concepts that depend on each other.

The domain modeling layer is when the assessment argument is first developed. From the domain analysis a list of focal knowledge, skills, and abilities (FKSAs) are developed. The FKSAs highlight the aspects of the domain that could be the focus of an assessment. It is in the development of the FKSAs that the concepts being measured are further defined. This process not only provides information about what should be assessed, but can also provide information about important aspects for teachers to focus on in their instruction. The concepts and practices of the CS discipline are specified in ways that highlight how these can be integrated, as well as which concepts and practices might want to be focused on separately.

Along with the FKSAs, a list of additional KSAs (knowledge, skills, and abilities; aspects of the domain that may be required of students when engaging in tasks but are not the focus of the tasks) is generated. These are KSAs that developers need to determine if they want to make a requirement of the assessment or support in the assessment. At the domain modeling layer, possible types of evidence required to measure the FKSAs are outlined and features of tasks that will provide that evidence are defined. One common document that is used to contain these specifications is a design pattern (Mislevy and Riconscente, 2006). This document not only specifies different aspects of the models that make up the assessment argument but also highlights the connection between the models.

Aspects of the assessment argument are further defined in the Conceptual Assessment Framework (CAF) layer. At this layer the constraints of the assessments, such as the time required and the format of the tasks, are considered. The layout of the assessment is defined and specified in a document often referred to as a task template (Mislevy and Riconscente, 2006). These specifications include which FKSAs should be included on the assessments, the structure of the assessment and the type of evidence that will be collected from the assessment.

In the assessment implementation layer, tasks and rubrics are developed matching the specifications developed in the task template. These tasks should be reviewed for alignment to the FKSAs and the goals of the assessment. The tasks are then combined to make the final assessment. In the final layer the assessment tasks are administered and the data from that administration is analyzed to determine if revisions are required.

The use of an ECD process for assessment development focuses the designer on the critical aspects of the assessment and helps ensure that the assessment matches the purpose of the assessment. This approach supports the validity of the assessment (Mislevy, 2007) by providing evidence of the construct validity of the assessment. The documents that are generated from the use of the approach provide a record of the decisions that were made when developing the assessment. These documents serve multiple purposes. One purpose is that they demonstrate the alignment of the features of the task with the FKSAs that are being measured, again supporting the validity of the assessment.

The other purpose is that they support the development of similar and/or parallel tasks (Mislevy and Haertel, 2006). An assessment developer can walk through the decisions made on the original assessment and modify the task template to reflect the new decisions. The information gathered at the domain analysis and domain modeling layers can be reused, while the degree of similarity of the new assessment determines how much the task template would need to be modified. Since a fair amount of the

TABLE 1 | Sample section of unit 4 partial design pattern.

Section	Activities (from curriculum website)	Learning goal (from curriculum website)	Computational FKSA	Assessment-unit assessment
4.12b: Game building-output and notifications	Students add an IF-THEN structure to give the user a choice when to end a game	Students learn to apply conditional if-then blocks to create alternative outcomes in a program	Focus is on recognizing conditional IF-THEN coding structures to identify game outcomes	Assessment item 1 Assessment item 3c
			Focus is on applying conditional IF-THEN coding structure to create alternative game outcomes	Assessed with student work products

previous development work can be re-used, new assessments do not take as much time to be created.

Research Questions

Our efforts to develop knowledge assessments for the redesigned curriculum through use of the ECD framework were guided by the following research questions:

1) How can we best utilize the well-established ECD framework to develop assessments that are tightly coupled to the stated learning goals of a redesigned curriculum?
2) How does the evidence generated through the ECD process and accompanying documentation support the validity argument for the newly developed assessments?

DESIGN PATTERN DEVELOPMENT AND ITEM WRITING

As discussed earlier, a key component of the ECD approach is the creation of a design pattern, a document which maps out the learning goals and associated FKSAs on which each item is intended to elicit evidence (Mislevy and Riconscente, 2006). The first step in creating the design pattern was pulling activity descriptions and learning goals from each section of the curriculum. Next, learning goals were expanded and distilled into their underlying FKSAs. Finally, FKSAs were analyzed for the extent to which they were suitable for inclusion on a knowledge assessment, or should be assessed via examination of student work products or some other data source; in some cases FKSAs were deemed outside the scope of the overall assessment strategy due to the overall volume of FKSAs and their variance in terms of being more or less central to the curriculum aims. Item ideas and specifications were then developed for the appropriate FKSAs. At various points through the development of the design pattern for each unit, a draft was reviewed by our collaborator, who is an expert in evidence-centered design. A sample section of the partial design pattern for Unit 4 is provided in **Table 1**.

We can consider the design pattern contents in **Table 1** within the context of the three models inherent in ECD: the student, evidence, and task models. The student model encompasses what students "know and can do" (Snoe et al., 2019, p. 106); in the case of the learning goal presented in **Table 1**, the student model, represented by the computational FKSAs, includes recognizing and applying conditional IF-THEN coding structures to determine specified game outcomes. The evidence model and task models (not shown in **Table 1**) are then elaborated for each of these FKSAs. The evidence model specifies what would need to be seen in the student response that provides information on whether or not the student is able to successfully engage with the FKSA. For example, if students are able to recognize the if/then coding structure, then what we would look for is whether or not students were able to match an if/then statement to a description of the game element where that type of statement is needed, or whether or not students are able to correctly follow the flow of a given if/then statement. Evidence that students are able to apply an if/then structure could be demonstrated by the extent to which students can construct an if/then statement that matches a supplied game scenario.

The task model specifies a range of assessment tasks that students could engage in to generate the evidence specified by the evidence model. Such tasks could entail selecting from a list of game outcomes the one that would be programmed with an IF-THEN conditional coding structure, selecting the correct coding structure needed to generate a given game outcome (that would in this case require a conditional IF-THEN coding structure), or working through an open-ended problem in which they write code to program a given game event and outcome (which in this case would again require a conditional IF-THEN coding structure). These assessment tasks are reflected in assessment items 1 and 3c from the Unit 4 assessment. In item 1, students are asked to select from a list of possible game events the one that would be correctly programmed using an IF-THEN conditional coding structure. A similar type of item is presented within the multipart Item 3c: students review two multi-step algorithms and are asked to select the step that would be programmed with a conditional IF-THEN coding structure. In both of these items, selecting the correct event provides evidence that students can recognize the alignment between a conditional IF-THEN coding structure and its resultant game outcome.

After completing the relevant sections of the design pattern, researchers and curriculum developers collaborated to develop the assessment for each unit. One key consideration aided by the creation of the design pattern was selecting the subset of FKSAs most appropriate for measurement via the formal unit level assessments. To aid in this decision a table was created that included the list of topics, the degree to which the concept was emphasized in the curriculum (e.g., content covered in a short lecture vs. content introduced and practiced through multiple iterations during the creation and refinement of a work product), the opportunities outside of the assessment for students to demonstrate the FKSA, and the complexity of the task required to measure the FKSAs (including reading and writing

TABLE 2 | Unit assessments with item counts, types and sources.

Unit	Technical content	Total # items	Item type		Item source	
			# MC items	# OE items	# New items	# Modified items
1	PowerPoint	9	7	2	7	2
2	Web design (Google sites)	8	4	4	5	3
3	Coding music (EarSketch)	5	2	3	3	2
4	Coding an app (AppInventor)	4	2	2	2	2

requirements). The assessment was then designed to measure FKSAs which were not covered on other assessment opportunities, were emphasized in the curriculum, and/or the level of complexity of the task was appropriate for an assessment that was to be given in one class period. The design pattern document allowed for a clear mapping of each FKSA onto an assessment item and/or student work product, making gaps in coverage as well as FKSAs with possibly excessive coverage readily evident during test construction. A subset of the assessment items was adapted from an existing CS assessment (Bienkowski et al., 2015); in some cases the modifications of existing items were minor, while in other cases they were more substantial (e.g., in the case of the introduction to programming items, the original items were related to programming with the Scratch platform, so the items needed to be rewritten to reflect the platforms taught in the curriculum).

In cases where items from the existing assessments did not assess FKSAs deemed important for inclusion in the unit assessments, new items were developed by researchers and curriculum developers. A combination of multiple choice and open-ended items was used to elicit evidence of a given FKSA from students. In some cases, students were asked to sketch out the design for a website, write or rewrite HTML code, or explain why they answered a question in a certain way. Consideration was also given to limiting the overall writing load required by each assessment, given teachers' reports of students' reluctance to write, as well as the overall length of each assessment. Psychometricians have noted that an over-reliance on open-ended items, as compared to multiple-choice items, carries costs by virtue of requiring additional examinee time (Lukhele et al., 1994; Livingston, 2009). Many items have multiple parts, and many of these multi-part items contain a mix of open-ended (OE) and multiple choice (MC) item types. **Table 2** shows the total number of items in each assessment, broken down by both multiple choice vs. open-ended, and newly developed item vs. modified from the existing assessments. In cases where at least one part of the item is open-ended, the item has been classified as open-ended. For this paper, we decided to focus on the assessment development for Units 2 and 4 to illustrate examples of item development and validation.

Assessment Context, Description, and Sample Items

Assessment Implementation and Sample Description

Unit assessments were piloted in four classes in one of our focal schools during the 2018–2019 academic year; students in these classes were taught by our more experienced partner teacher, who was implementing the curriculum for the second time that year. The unit assessments were provided to our partner teacher, who administered them in a paper and pencil format during a class period at the conclusion of the given unit. The assessments were intended to be roughly 30 minutes in duration. It was expected that students would work individually on the unit assessments and turn them in prior to leaving class. The assessment pilot as well as all other research activities associated with this project were conducted in accordance with a protocol approved by our university's Institutional Review Board.

The Unit 2 assessment was taken by a total of 101 students from across the four class periods. Among these test takers, the bulk were in 9th grade (33 students) and 10th grade (36 students), with smaller groups of students in 11th grade (16 students) and 12th grade (15 students). The Unit 4 assessment was taken by a total of 95 students from across the four class periods. Among these test takers, the bulk were in 9th grade (37 students) and 10th grade (30 students), with smaller groups of students in 11th grade (16 students) and 12th grade (9 students).

Unit 2: Unit Overview, Assessment Description, and Sample Items

In Unit 2, students are introduced to computer systems and web development. Students apply those concepts to the development of a website designed to explore and raise awareness about the focal topic that they selected. As noted above, in many cases, the assessment plan for a given FKSA involves evaluating a student work product (e.g., a worksheet or digital artifact) rather than assessing it through a test item. The FKSAs that were selected to be measured on the assessment are shown in **Table 3**. The Unit 2 assessment contains a total of eight items; of these, three are multiple choice and one is a yes/no question, and the other four are open-ended items in which students are asked to sketch, circle an element of a website, or write text as their response. In **Table 3**, each item is briefly described, and the FKSA(s) it is intended to yield evidence of is specified. Sample items are presented in **Figures 2**, **3** and discussed in detail below.

Item 2 was modified from an existing assessment item focused on web design (Snow, 2016). In the existing version of this item, students are shown a sample website and asked to respond "Y/N" to indicate whether each of a series of website elements is present in the sample website. This item was modified slightly for use in the Unit 2 assessment. Rather than using the sample website and website elements from the existing item, a new website was created for this item to match the type of websites students made during Unit 2 of the curriculum, namely one made in

TABLE 3 | Overview of unit 2 assessment items.

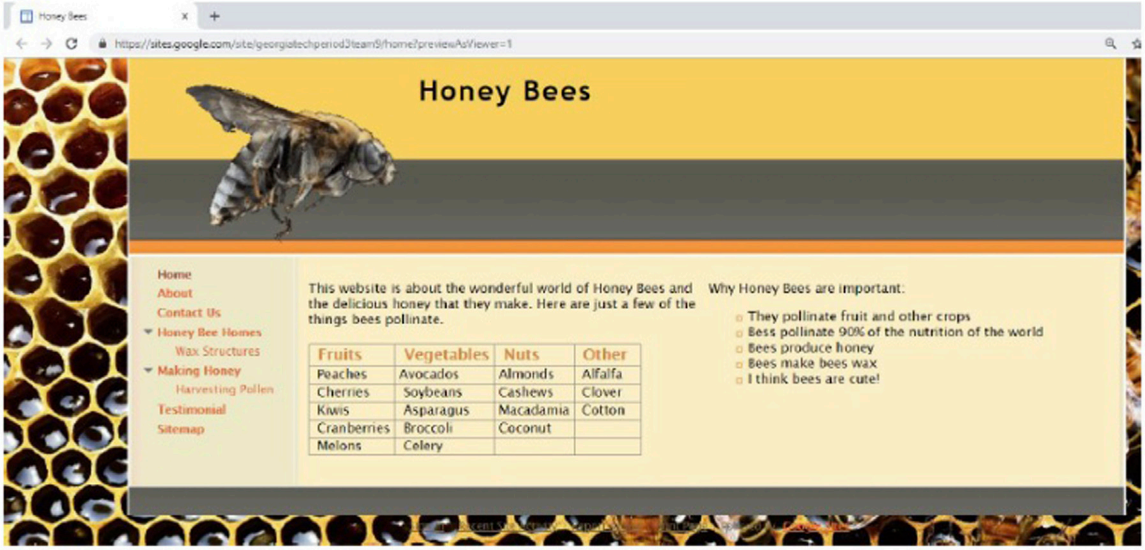
Item number	FKSA(s)	Item type	Item description
1	Focus is on the ability to generate a layout for a website based on a set of required features for that website	Open-ended (sketch)	A website description is provided and required features are listed. Students are asked to sketch out the website, including all required features in their sketch
2	Focus is on identifying various features of websites and how users react to those features	8 parts, Y/N	A sample website is shown and a list of potential website elements is presented. Students respond Y/N as to whether each element is present in the sample website
3	Focus is on the skills of identifying and correcting errors in a website's design, organization, and/or layout in order to improve its functionality and effectiveness (parts a–d) Focus is on explaining the link between HTML code and what appears on a webpage (e.g., HTML code for italicized text results in text appearing in italics on a webpage) (parts e–h) Focus is on the ability to change webpage elements by directly manipulating the existing HTML code (parts e–h)	8 parts, circling content and written explanation	A sample website is shown and requirements for this website are listed; HTML code for the sample website is provided. Students must do the following for two errors: Circle the error on the website, explain why it is an error, circle the HTML code that corresponds to the error, and explain how they would fix the error in the code
4	Focus is on identifying how websites differ in content, design, and structure on the basis of their authors and origin	2 parts, written explanation	Students are asked to imagine creating a specific type of website, to provide two pieces of content for that website for two unique audiences, and to explain why each piece of content is relevant to that audience
5	Focus is on identifying various computer systems, computer networks, and components of them, including stating the definition of various computer system, LAN, and WAN terminology	Multiple choice	Item asks students to select what type of computer system component a printer is
6	Focus is on identifying various computer systems, computer networks, and components of them, including stating the definition of various computer system, LAN, and WAN terminology	Multiple choice	Item asks students about the purpose of two computer network components
7	Focus is on recognizing the responsibilities associated with a set of roles in web design	Multiple choice	Item asks students about the responsibilities of web developer roles
8	Focus is on understanding the pros and cons of using HTML code generators (as compared to hand-coding HTML)	Open-ended, written explanation	Students are asked to provide one benefit and one drawback of using an HTML coding tool as compared to hand-coding HTML

Google sites (classic edition). This was done to ensure alignment between 1) sample websites presented in the curriculum, and the website students created using Google sites, and 2) the website presented in the assessment. The list of website elements for students to assess the presence or absence of in the sample website was also modified to align with the website components focused on within the curriculum.

Item 4 was newly written for this assessment. In this item, students are asked to imagine that a school principal is designing a website about the school cafeteria. They are asked to suggest one piece of information for this website that would be of interest to students, and one that would be of interest to cafeteria staff members. Further, they are asked to explain why each piece of information would be of interest to the relevant group. This is an open-ended item in which students are asked to provide their answer via written text. The scoring rubric for this item is discussed below in the "Rubric Development and Sample Item Scoring" section.

To demonstrate the link between the learning goals, FKSAs, and assessment items, the design pattern rows corresponding to these two sample items are presented in **Table 4**. The learning goal underlying item 2 is "identify and categorize website elements that help to communicate intended messages to a targeted audience." In distilling this learning goal into an FKSA to guide item writing, the identification element and the communicating intended messages elements were prioritized given their alignment with the course activities entailed in teaching this learning goal, which largely revolved around scaffolded steps toward website creation. The scope of the FKSA encompasses both feature identification and the function of those features for intended users. This FKSA was selected for an assessment item primarily due to its critical function as the basis for successful website development: a student must be able to correctly recognize and identify typical website features in order to employ these features in the creation of his/her own website. A significant amount of class time was devoted to students exploring existing websites and studying their features and the utility of the various features in the effectiveness, or lack thereof, of the overall website. In developing assessment item 2, we had originally written a longer, more involved item involving not only the identification of the presence of website elements in a sample website, but also having students label some website features, and having students provide a written explanation of the utility of certain website features in making the website clear and easy to understand; such an item would align with the full scope of the FKSA. After an external review and consideration of teachers' feedback, the overall writing load of the draft assessment was deemed too much for students. Accordingly, this item was cut down to consist of only the Y/N

2. Below is a picture of a web page. Use this web page to complete the tasks below.



Honey Bees

https://sites.google.com/site/georgiatechperiod3team9/home?previewAsViewer=1

Honey Bees

- Home
- About
- Contact Us
- Honey Bee Homes
  - Wax Structures
- Making Honey
  - Harvesting Pollen
- Testimonial
- Sitemap

This website is about the wonderful world of Honey Bees and the delicious honey that they make. Here are just a few of the things bees pollinate.

Fruits	Vegetables	Nuts	Other
Peaches	Avocados	Almonds	Alfalfa
Cherries	Soybeans	Cashews	Clover
Kiwis	Asparagus	Macadamia	Cotton
Cranberries	Broccoli	Coconut	
Melons	Celery		

Why Honey Bees are important:

- They pollinate fruit and other crops
- Bess pollinate 90% of the nutrition of the world
- Bees produce honey
- Bees make bees wax
- I think bees are cute!

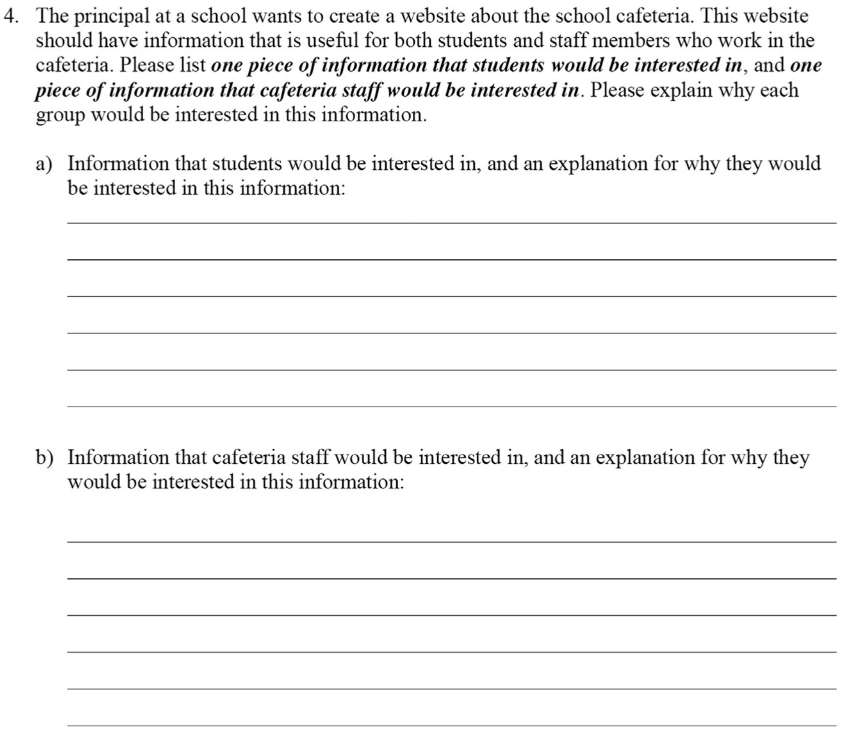
For each element listed below, select "**Yes**" if the element is included on the web page shown or "**No**" if the element is not included.

Element	Included?
a) A web page **heading**	○ Yes ○ No
b) A **horizontal navigation** bar	○ Yes ○ No
c) A **vertical navigation** bar	○ Yes ○ No
d) A **background image**	○ Yes ○ No
e) A **table** in the main text section	○ Yes ○ No
f) An **image** in the main text section	○ Yes ○ No
g) A **video** in the main text section	○ Yes ○ No
h) The **title** of the webpage	○ Yes ○ No

FIGURE 2 | Sample item: Unit 2 assessment, item 2.

identification of the presence of various website features in the sample website.

In developing item 4, we focused on the part of the learning goal and FKSA that had students identifying content differences based on audience. Students' ability to modify design and structure based on audience can be investigated through scoring of student websites. In this case, the scope of the FKSA is broad, covering the various facets of a website that can vary on the basis of its audience and/or author. In order to keep the item of a manageable writing load for the assessment, test developers decided to focus on the content piece, with the expectation that design and structure decisions made by students in creating their websites could be evaluated through a review of the final



4. The principal at a school wants to create a website about the school cafeteria. This website should have information that is useful for both students and staff members who work in the cafeteria. Please list ***one piece of information that students would be interested in***, and ***one piece of information that cafeteria staff would be interested in***. Please explain why each group would be interested in this information.

   a) Information that students would be interested in, and an explanation for why they would be interested in this information:

   ____
   ____
   ____
   ____
   ____
   ____

   b) Information that cafeteria staff would be interested in, and an explanation for why they would be interested in this information:

   ____
   ____
   ____
   ____
   ____
   ____

FIGURE 3 | Sample item: Unit 2 assessment, item 4.

websites. The development of these items illustrates the tension between logistical considerations in test design and FKSA alignment and coverage. Utilizing ECD and the design pattern document helped guide decisions where this tension had to be resolved, in that test developers could easily check for alternate assessment or student work product of a given FKSA and decide if its overall coverage would still be sufficient if a given item was reduced in complexity or eliminated.

Unit 4: Unit Overview, Assessment Description, and Sample Items

In Unit 4, students are introduced to programming concepts and practices which they use to develop a mobile application using the App Inventor program. The application is intended to engage users in the focal problem. For Unit 4, the assessment covered eight FKSAs. It contains a total of four items; of these, two are multiple choice format, and the other two are a combination of multiple choice and open-ended items in which students are asked to write text as their response. In **Table 5**, each item is briefly described, and the FKSA(s) it is intended to yield evidence of is specified. Sample items are presented in **Figures 4**, **5** and discussed in detail below.

Item 1 was newly written for this assessment to assess computational thinking. It is intended to assess students' understanding of the IF-THEN programming structure within the context of the App Inventor program (which

TABLE 4 | Design patterns rows corresponding to unit 2 sample assessment items.

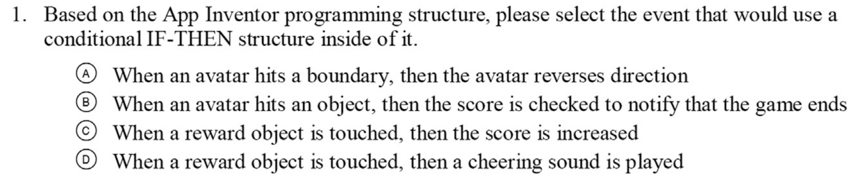
Section	Activities (from curriculum website)	Learning goal (from curriculum website)	FKSAs	Assessment item
2.3: Students identify common website themes and structures	Class reconvenes to discuss what students found during the website analysis; student pairs identify website features they want to use in their website	Identify and categorize website elements that help to communicate intended messages to a targeted audience	Focus is on identifying various features of websites and how users react to those features (need examples here)	Unit 2 assessment, item 2 (identify whether elements are present on a sample website, label them, and explain why a selected set of elements help make the website clear and easy to understand)
2.3: Students identify common website themes and structures	Students compare and contrast websites on the same topic: One professional and one DIY	Learn that websites vary in their design and structure to reach their audience and authors	Focus is on identifying how websites differ in content, design, and structure on the basis of their authors and origin	Unit 2 assessment, item 4 (school cafeteria website content - differences in website created by cafeteria staff and website created by students)

TABLE 5 | Overview of unit 4 assessment items.

Item number	FKSA(s)	Item type	Item description
1	Focus is on identifying features of an event that make it a conditional Focus is on recognizing conditional IF-THEN coding structures to identify game outcomes	Multiple choice	Students are asked to select from a list of game events the one that would use an IF-THEN programming structure within app inventor
2	Focus is on recognizing how game narrative relates to player engagement Focus is on identifying how mobile game design (layout, rules, and victory conditions) attributes can engage the user	Multiple choice (select all that apply)	Students are asked to select from a list of game elements all those that can be used to engage the game player
3	Focus is on identifying features of an event that make it a conditional Focus is on defining an IF-THEN control structure Focus is on recognizing conditional IF-THEN coding structures to identify game outcomes	5 parts; 3 are multiple choice, 2 are multiple choice + written text explanation	Two sample algorithms are described in detail. Students are asked to answer two multiple choice questions in which they select the appropriate output for a given input within each of the two algorithms. Students are asked to answer 2 multiple choice items in which they select which step within a given algorithm would be programmed with a certain programming structure; for one of these items, they are asked to explain their answer. Finally, students are asked a Y/N item about whether a given algorithm could be used to accomplish a given task, and to explain their answer
4	Focus is on using game rule patterns to identify events that are termed as WHEN-THEN (4a) Focus is on identifying features of an event that make it a conditional (4b) Focus is on defining an IF-THEN control structure (4b)	2 parts; multiple response types	Details of a game that is being programmed are described. An example occurrence within this game is described, and students are asked to identify the events included in this occurrence. Next, students are asked to select the event (from a list of events) that would be programmed using a specific programming structure, and to explain their answer

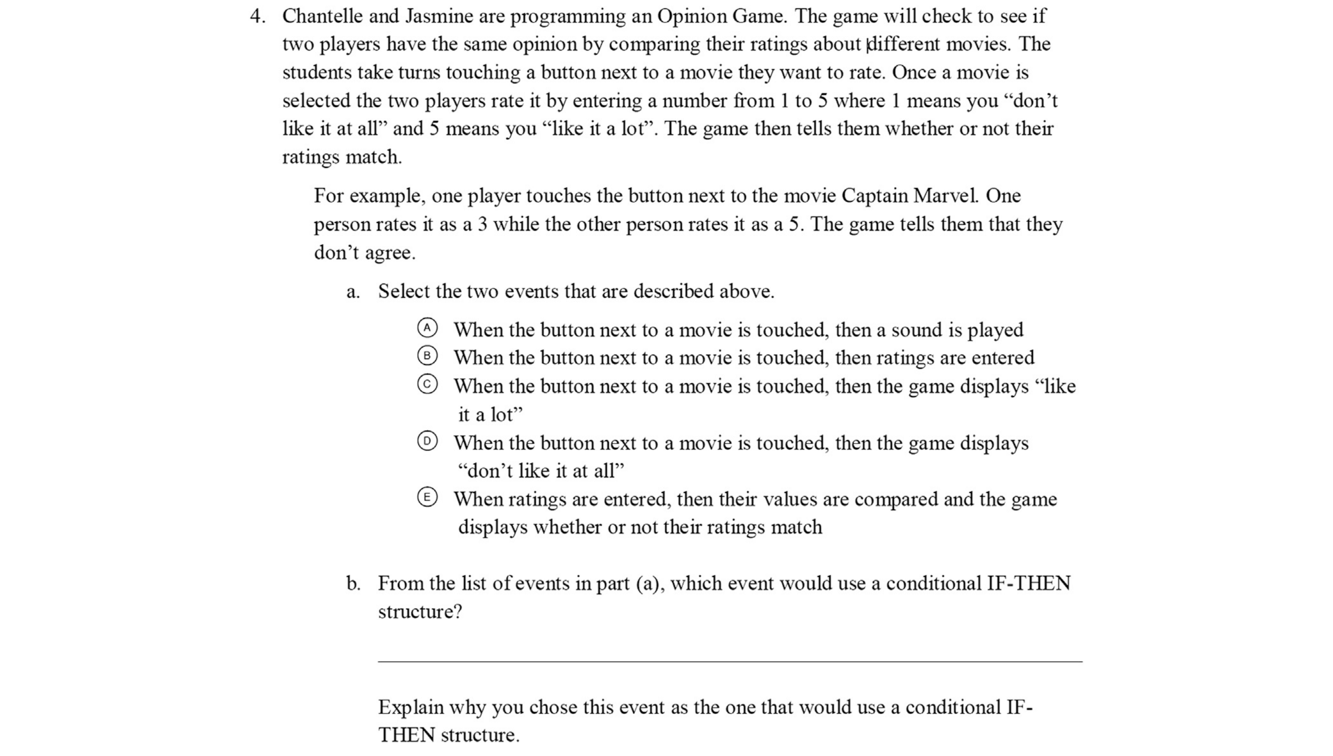
was the focus of Unit 4) by selecting from a series of events the one that would correctly be programmed using the IF-THEN programming structure. Item 4 was modified from an existing assessment item on computer programming (Snow, 2016). In the original item, students are provided with a detailed description of a simple game, and are asked to write out the way they would program this game using Scratch or Alice blocks. They are then asked a series of questions about the programming of the game. For the Unit 4 assessment, the game description, events, programming structures, and programming questions have been modified to align with the App Inventor programming context; the game itself was retained but nearly everything about this item was modified for use in the Unit 4 assessment. In this item, students are asked to match formal event definitions with a description of occurrences within a sample game. Next, they are asked to use their knowledge of the IF-THEN programming structure to select from a list of in-game events the one that would correctly be programmed using the IF-THEN programming structure. They are then asked to explain their selection of the event they chose to be used with the IF-THEN structure.

To demonstrate the link between the learning goals, FKSAs, and assessment items, the design pattern rows corresponding to these two sample items are presented in **Table 6**. For item 1, the aligned FKSA is “identifying features of an event that make it conditional”. This item is also related to students’ ability to identify IF-THEN coding structures as they relate to game outcomes. Emphasizing the identification piece of these two FKSAs lent itself to a multiple choice item in which several game events are described and students select the one that would correctly be programmed using an IF-THEN structure. The nature of this task aligns well with the programming decisions


1. Based on the App Inventor programming structure, please select the event that would use a conditional IF-THEN structure inside of it.

Ⓐ When an avatar hits a boundary, then the avatar reverses direction
Ⓑ When an avatar hits an object, then the score is checked to notify that the game ends
Ⓒ When a reward object is touched, then the score is increased
Ⓓ When a reward object is touched, then a cheering sound is played

FIGURE 4 | Sample item: Unit 4 assessment, item 1.



4. Chantelle and Jasmine are programming an Opinion Game. The game will check to see if two players have the same opinion by comparing their ratings about different movies. The students take turns touching a button next to a movie they want to rate. Once a movie is selected the two players rate it by entering a number from 1 to 5 where 1 means you "don't like it at all" and 5 means you "like it a lot". The game then tells them whether or not their ratings match.

For example, one player touches the button next to the movie Captain Marvel. One person rates it as a 3 while the other person rates it as a 5. The game tells them that they don't agree.

a. Select the two events that are described above.

Ⓐ When the button next to a movie is touched, then a sound is played
Ⓑ When the button next to a movie is touched, then ratings are entered
Ⓒ When the button next to a movie is touched, then the game displays "like it a lot"
Ⓓ When the button next to a movie is touched, then the game displays "don't like it at all"
Ⓔ When ratings are entered, then their values are compared and the game displays whether or not their ratings match

b. From the list of events in part (a), which event would use a conditional IF-THEN structure?

____

Explain why you chose this event as the one that would use a conditional IF-THEN structure.

FIGURE 5 | Sample item: Unit 4 assessment, item 4.

students are required to make as they create their games, and is easily captured with a multiple choice item. The parts of these FKSAs that are at a higher cognitive level (i.e., defining and applying the IF-THEN coding structure) are assessed in a longer, multi-part item. These FKSAs were covered in two assessment items due to both their heavy emphasis in the curriculum and the fact that their scope includes knowledge demonstrated at different levels of cognitive complexity, corresponding to different item types. For item 4a, the goal is to allow students to demonstrate their ability to link game rules patterns with game events. In item 4b, which is somewhat similar to Item 1, the student identifies features of an event than make it conditional by selecting the event from a list of events that would correctly be programmed using an IF-THEN structure.

TABLE 6 | Design patterns rows corresponding to Unit 4 sample assessment items.

Section	Activities (from curriculum website)	Learning goal (from curriculum website)	FKSAs (computational)	Assessment item
4.6: Pattern, structure and function in games	Students identify the WHEN-THENs in a game of UNO game and patterns in their structure-function; students create WHEN-THENs statements from the UNO game and rules, identifying structures that complete functions in the UNO game	Identify how game variety and engagement is largely dependent upon conditional events, or "IF-THEN" events	Focus is on identifying features of an event that make it a conditional focus is on defining an IF-THEN control structure	Assessment item 1 assessment item 3 assessment item 4b
4.6: Pattern, structure and function in games	Students identify WHEN-THEN statements in a game of UNO	Identify and analyze patterns in the design of common card, board, and video games to reveal how their components and structures create variety to engage players	Focus is on using game rule patterns to identify events that are termed as WHEN-THEN	Assessment item 4a
4.12b: Game building - output and notifications	Students add an IF-THEN structure to give the user a choice when to end a game	Students learn to apply conditional if-then blocks to create alternative outcomes in a program	Focus is on recognizing conditional IF-THEN coding structures to identify game outcomes. Focus is on applying conditional IF-THEN coding structure to create alternative game outcomes	Assessment item 1 assessment item 3

ASSESSMENT UTILIZATION AND SCORING

Intended Purpose of Assessments

The intended purpose of the unit assessments is to indicate the extent to which students can successfully engage with and demonstrate mastery of tasks linked to the content taught in each course unit. While individual items were written to be aligned with one or more FKSAs, the intention is not to obtain student scores for each FKSA. Instead, the collection of FKSAs is designed to represent what it means for students to engage with computational thinking in this curriculum. The score on the overall assessment then represents students' ability to engage with computational thinking. It is a measure of students' ability to engage with the content taught in the curriculum, and not a measure of students' strengths and weaknesses with regards to individual FKSAs. While teachers can look at student performance on individual tasks and make inferences about students' challenges and misconceptions, the overall assessment is not designed to provide teachers with detailed information at the FKSA level. Teachers may need to gather additional information to fully understand their students' ability on specific FKSAs. Taken together, the set of unit assessments is intended to provide an overall picture of how well students are able to perform on computational thinking tasks.

Rubric Development and Sample Item Scoring

For each unit, the rubrics were jointly developed by the research team, including the main assessment team, and the curriculum team. Scoring of items fell into three categories: 1) simple Y/N or multiple choice scoring with one or more correct answers, for which only a straightforward answer key was needed; 2) open-ended items modified from existing items, in which case the rubric was modified from the existing scoring rubric to reflect modifications made to the items; 3) newly created open-ended items, for which a scoring rubric was created from scratch. The approach taken in most cases was to create and/or modify scoring rubrics for open-ended items prior to reviewing the student responses. Then a sample of student responses were scored with the draft rubrics, rubrics were revised as needed, followed by scoring of the full set of student responses and additional rubric revisions as needed. Scoring challenges presented mostly as a student response that had not been anticipated and was not captured by the range of options dealt with in the rubric; these scenarios typically resulted in an expansion to the scoring rubric.

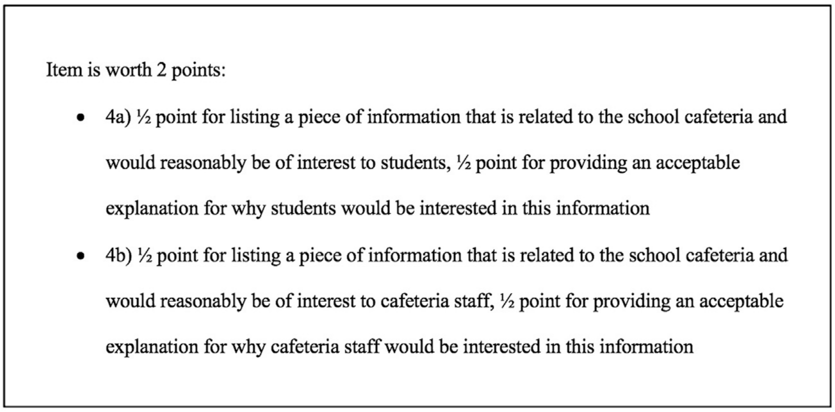
The content and FKSAs covered within the curriculum lent themselves to using a variety of item types (i.e., Y/N, multiple choice with a single correct response, multiple choice with multiple correct responses, open-ended items, and items combining multiple choice and open-ended within a single item), and utilizing this variety of item types allowed for an optimal match between each FKSA and the type of evidence students were able to provide for the corresponding item. This variety of item types, however, presented significant challenges with scoring and item analysis. Decisions about how to weight each item with respect to the total assessment score were somewhat unclear, as were decisions about when and how to give partial credit. Rubrics for the existing assessments provided substantial guidance on these issues. Partial credit was awarded following the convention used in the rubrics provided for the existing assessments from which items were modified. For "select all that apply" items, multiple choice items with multiple correct responses, students received partial credit for selecting some subset of the correct answers, but received no credit if they selected one or more incorrect answers. For example, if there were two correct answers, they received partial credit for selecting one correct answer and no other answers. But if students selected one correct answer and one incorrect answer, they received no credit. For open-ended items, students were awarded partial credit for providing some but not all of the correct information that would constitute a full credit response.

For the most part, longer, multi-part items were scored with either ½ or 1 point being awarded for each part, depending on the complexity of the parts. Single multiple choice items were typically awarded one point each. Less focal FKSAs were assessed with a single multiple choice item, while more critical FKSAs were assessed with longer multi-part items and/or open-ended items (or item parts). This generally resulted in more assessment points being awarded for the more complex and important FKSAs, on which more time was spent in the curriculum. For example, in Unit 2, FKSAs related to generating website layout and identifying website features were each assessed with multi-part items worth 5.5 and 4 points, respectively. An FKSA that is both less complex and less focal in the curriculum, recognizing the responsibilities associated with each web design role, was assessed with a single multiple choice item worth 1 point.

There were also issues with contingencies within items; for example, in item 4b on the Unit 4 assessment, students were asked to select an event and then explain why they selected that event. Students were only able to potentially provide a correct explanation if they had first selected the correct event. We have learned that these issues of scoring, weighting, and contingencies are a necessary component of an assessment using multiple item types, and we would have been better served by working through these issues during item development rather than after the fact.

For assessment item 4 in Unit 2, the rubric was less structured given the extremely wide range of possible student responses to this item. The scoring criteria for this item instruct the scorer to consider whether the piece of information provided is likely to be of interest to the given audience (either cafeteria workers or students), and whether the student provides a reasonable explanation as to why the piece of information would be useful to the given audience. The instructions for scoring this item, taken from the scoring rubric, are displayed in **Figure 6**.

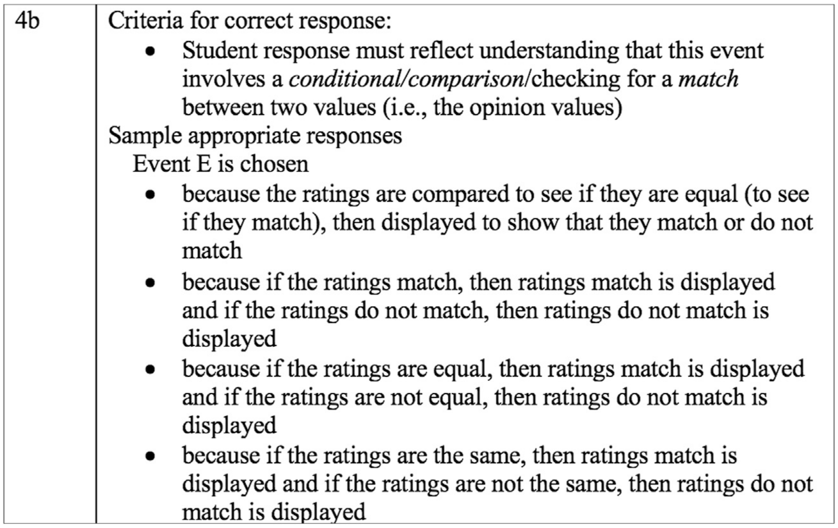
Here are sample student responses from Unit 2, items 4a and 4b, respectively, that received full credit due to providing an appropriate piece of content and corresponding, sufficiently thorough, explanation.



Item is worth 2 points:

- 4a) ½ point for listing a piece of information that is related to the school cafeteria and would reasonably be of interest to students, ½ point for providing an acceptable explanation for why students would be interested in this information
- 4b) ½ point for listing a piece of information that is related to the school cafeteria and would reasonably be of interest to cafeteria staff, ½ point for providing an acceptable explanation for why cafeteria staff would be interested in this information

FIGURE 6 | Scoring rubric for unit 2 assessment, item 4.



4b	Criteria for correct response: • Student response must reflect understanding that this event involves a *conditional/comparison*/checking for a *match* between two values (i.e., the opinion values) Sample appropriate responses Event E is chosen • because the ratings are compared to see if they are equal (to see if they match), then displayed to show that they match or do not match • because if the ratings match, then ratings match is displayed and if the ratings do not match, then ratings do not match is displayed • because if the ratings are equal, then ratings match is displayed and if the ratings are not equal, then ratings do not match is displayed • because if the ratings are the same, then ratings match is displayed and if the ratings are not the same, then ratings do not match is displayed

FIGURE 7 | Scoring rubric for unit 4 assessment, item 4b.

> They would be interested in knowing what is for lunch the next day. They would be interested in this information because this could help them decide if they want to bring lunch from home [4a, information of interest to students].
>
> The cafeteria staff would be interested in nutrition facts about the food. They want to make sure the food being served is nutritious and healthy for students [4b, information of interest to cafeteria staff].

The following sample student response received partial credit, as it contains information students would find useful (menu items and calorie counts), but fails to provide an explanation as to why students would find this information useful:

> A lunch calender to tell studnets whats being had for lunch and its calories [4a, information of interest to students].

Most student responses either earned full credit or earned partial credit for providing information only without an explanation. Some students left the item blank. In a few cases, students provided content that was related to suggested improvements to the cafeteria, rather than current information that would be of interest to students or cafeteria staff. In these cases, the responses were given no credit. A sample of no credit responses is presented below:

> Selling cady at lunch because students love eating candy [4a, information of interest to students].

> Staff would be interested in higher pay to support themselves more [4b, information of interest to cafeteria staff].

For assessment item 4b in Unit 4, the scoring rubric provides a general guideline for the key information that must be conveyed in a correct student response; the student must convey, either by directly stating with the expecting wording or describing using other words the idea that the event involves a conditional, comparison of two values, and/or checking for the presence of a match between two values. Sample appropriate responses taken directly from student responses are also included in the rubric for the scorer's reference. The instructions for scoring this item, taken from the scoring rubric, are displayed in **Figure 7**.

Full credit responses provide evidence of student understanding of the notion that a comparison is being made between two values and a determination is made regarding a match, or lack of a match, between those two values. Here is a sample of full credit student responses:

> *Because It compares the ratings and depending on the ratings given the game displays a certain message.*
>
> *If the ratings match, then the game tells them they agree. If the ratings don't match, then the game tells them they don't agree.*

The student responses received no credit for a variety of reasons. Nearly 30% of students left the item blank. Among those students who wrote a response that received no credit, two of the more commonly occurring categories of these incorrect responses are as follows:

1: Student described one of the events in the game that does not entail an IF-THEN comparison:

> *If a button next to a movie is touched then the game displays "like it a lot" same for choice D.*
>
> *Event B fits perfectly in if-then because it is basically stating "if a button is touched, then a rating is entered". It is only two steps, therefore such structure can be used.*

2: Student provided some general statement justifying the use of an IF-THEN structure that is not sufficiently detailed or specific to be meaningfully interpreted such as "It fits in the structure better"; "Because it would be the easiest to use"; and "It works fine as an if then statement."

Item Validation

Unit 2: Item Analysis and Student Misconceptions

For the most part, student responses to Unit 2, Item 2 revealed few misconceptions. Most students accurately identified the presence or absence of a series of website elements in the sample website. After instruction on the Unit 2 curriculum, the majority of students correctly identified the presence or absence of a web page heading (present; correctly identified by 91% of students), a horizontal navigation bar (absent; correctly identified by 81% of students), a vertical navigation bar (present; correctly identified by 78% of students), a background image (present; correctly identified by 93% of students), a table in the main text section (present; correctly identified by 95% of students), a video in the main text section (absent; correctly identified by 96% of students), and the title of the webpage (present; correctly identified by 85% of students). The only misconception revealed in student responses to this item is around the placement of an image. In the sample webpage, there is an image in the header of the website, while there is no image in the main text section. It appears that this distinction confused students, as only 38% of them correctly identified that an image in the main text section was absent. It seems that they misattributed the location of the image in the webpage's header as being within the main text section. Other than this single misconception surrounding image placement within specific areas of the webpage, students were largely able to correctly identify the presence or absence of website features within a sample webpage.

The multiple-choice items, 5, 6, and 7, focused on measuring specific terminology and concepts from the curriculum, such as computer systems, networks and web development roles. For these items, the difficulty indices were calculated. Question five is an easy item, as 80% of the students answered it correctly. For questions 6 and 7, the item difficulty was moderate: 70% answered item 6 correctly and 50% answered item 7 correctly.

The mean score for the 101 students who took the Unit 2 assessment was 16.13 (SD = 5.59) out of 24.5 possible points, or 65.83%. Full item analysis on all Unit 2 assessment items is presented in **Table 7**. The proportion of students answering an item correctly indicates the difficulty level of the item. For items other than straightforward multiple choice where a simple correct vs. incorrect distinction was not applicable, the difficulty indices were calculated on the basis of students "passing" vs. "not passing" the item (i.e., earning a score of 60% or higher on a given item). The discrimination index (DI) (Kelley, 1939) is a statistic which indicates the extent to which an item has discriminated between the high scorers and low scorers on the test. The index is represented as a fraction and varies between −1 and 1. Optimally an item should have a positive discrimination index of atleast 0.2, which indicates that high scorers have a high probability of answering correctly and low scorers have a low probability of answering correctly. The DI for multiple choice items were computed from equal-sized high and low scoring groups on the test. Subtract the number of successes by the low group on the item from the number of successes by the high group, and divide this difference by the size of a group. DI for open ended items were calculated slightly different. It was calculated as the total number of points received by the test takers in the strong and the weak group respectively, and then divided each of the obtained numbers by the maximum number of points that can be awarded to the students in either group. For this assessment, difficulty indices are primarily in the average category while most items have a good discrimination index. Note

TABLE 7 | Unit 2 item analysis.

Status/Score	Count	Percent	Item difficulty	Discrimination index
Item 1 (max score = 5.5; mean = 3.39; SD = 2.07)				
Not passing (score of 3 or lower)	33	32.7	Difficulty index = 0.67 Average	0.48
Passing (score of 3.5 or higher)	68	67.3		Good
Item 2 (max score = 4; mean = 3.29; SD = 0.68)				
Not passing (score of 2 or lower)	6	5.9	Difficulty index = 0.94 Easy	0.11
Passing (score of 2.5 or higher)	95	94.1		Fair
Item 3 (max score = 8; mean = 5.45; SD = 2.43)				
Not passing (score of 4 or lower)	39	38.6	Difficulty index = 0.61 Average	0.37
Passing (score of 5 or higher)	62	61.4		Good
Item 4 (max score = 2.0; mean = 1.20; SD = 0.73)				
Not passing (score of 1 or lower)	47	46.5	Difficulty index = 0.54 Average	0.40
Passing (score of 1.5 or higher)	54	53.5		Good
Item 5				
Incorrect	20	19.8	Difficulty index = 0.80 Easy	0.52
Correct	81	80.2		Good
Item 6				
Incorrect	31	30.7	Difficulty index = 0.69 Average	0.45
Correct	70	69.3		Good
Item 7				
Incorrect	51	50.5	Difficulty index = 0.50 Average	0.48
Correct	50	49.5		Good
Item 8 (max score = 2.0; mean = 0.90; SD = 0.87)				
Not passing (score of 1 or lower)	70	69.3	Item difficulty = 0.31 Average	0.45
Passing (score of 2)	31	30.7		Good

that there is a range of difficulty of items which provides evidence that the assessment is able to differentiate between students. The discrimination indices are all in the good range other than for the one item that had low difficulty (which is to be expected). This also provides evidence that the assessment is able to distinguish between students.

Unit 4: Item Analysis and Student Misconceptions

Examination of student responses to Item 4 on the Unit 4 assessment reveal some misconceptions in student understanding of the content being assessed. In 4a, a portion of the game as it is being played by two players is described in narrative form. Students are then asked to select from a list of events the two events that are being described in the narrative game description. After receiving instruction on the Unit 4 curriculum, roughly 40% of students got this item correct, selecting the two correct events. Another 10% of students earned partial credit on this item by selecting one of the correct events and no incorrect events.

The remaining 50% of students received no credit because they selected one or more incorrect events. Of the three incorrect events, two were selected at a higher rate than the third: nearly 25% of students selected event C and/or event D, which are, respectively, "when the button next to a movie is touched, then the game displays 'like it a lot'", and "when the button next to a movie is touched, then the game displays 'don't like it at all'". While a player could provide a rating of a movie that is "like it a lot" or "don't like it all", the game itself stores that information as a numeric value and then displays if the ratings agree or not. Students may be confused with the engagement of the player with the rating (i.e., the rating labels corresponding to the numerical ratings that players see) and what the game actually displays (i.e., the presence or absence of a match between the two players' ratings). The least-selected incorrect event (event A), "When the button next to a movie is touched, then a sound is played", was selected by roughly 10% of students. Nowhere in the game description is a sound played, and most students recognized this event was not part of the game.

In Item 4b, students were asked to select from the list of events in 4a the event that would use a conditional IF-THEN structure, and then explain why they selected this event. Many students (roughly 45%) either left this item blank or wrote something that did not match one of the events. Nearly 25% of students selected the correct response, Event E ("When ratings are entered, then their values are compared and the game displays whether or not their ratings match"). Between 6 and 12% of students selected the four incorrect events; the selection of any one of these four incorrect events indicates a misconception about what types of events require IF-THEN programming. All of the incorrect events are of the structure when something happens, the game responds in a certain way. However, in App inventor these types of events do not require an IF-THEN structure. Event E, which involves a comparison of two ratings to determine the presence or absence of a match, is the only one that aligns with the IF-THEN structure: if the ratings have the same value, then the game responds that there is a match. An IF-THEN event must entail a comparison of some kind.

The explanations provided by students who selected incorrect events in Item 4b further reveal the nature of this misconception, namely that an event of the structure "when the user does something, the game responds in this way", aligns with the need for a conditional IF-THEN programming structure. Students' explanations indicate that they incorrectly interpret these types of events within the context of a conditional IF-THEN

TABLE 8 | Unit 4 item analysis.

Status/Score	Count	Percent	Item difficulty	Discrimination index
Item 1				
Incorrect	63	66.3	Difficulty index = 0.34 Average	0.06
Correct	32	33.7		Poor
Item 2				
Incorrect	73*	76.8	Difficulty index = 0.23 Hard	0.10
Correct	22	23.2		Poor
Item 3 (max score = 7; mean = 2.79; SD = 1.83)				
Not passing (score of 4 or lower)	72	75.8	Difficulty index = 0.24 Hard	0.40
Passing (score of 4.5 or higher)	23	24.2		Good
Item 4 (max score = 3; mean = 0.86; SD = 0.99)				
Not passing (score of 1.5 or lower)	78	82.1	Difficulty index = 0.18 Hard	0.42
Passing (score of 2 or higher)	17	17.9		Good

**Note: 59 students received partial credit on this item, selecting one or two of the three correct answers and selecting no incorrect answers.*

statement, misattributing a WHEN-THEN type of event as an IF-THEN type of event:

> *Event B fits perfectly in if-then because it is basically stating "if a button is touched, then a rating is entered." It is only two steps, therefore such structure can be used.*
>
> *Because if the button is touched the game displays "don't like it all".*

For multiple-choice items, the item difficulty and discrimination index were calculated. For the first item, 34% of the students answered it correctly, which shows that it is moderately difficult. Further, the discrimination index shows that the item is not adequate to discriminate between high achieving and low achieving groups, which might be an indication that students chose the correct answer randomly. For item 3, which has two multiple-choice questions a and b, the difficulty level was moderate. Approximately 54% of the students got item 3a correct, and 70% got item 3b correct. The discrimination indices indicate that items 3a and 3b are good questions that discriminate well between the upper third of those scoring on this exam and the lower third.

Item 4 has two parts, 4a and 4b. In part 4a, students are asked to select two events corresponding to a partial game description. There are five possible events, and students are asked to select two. Students received full credit for selecting both correct events, partial credit for selecting one of the correct events and no incorrect events, and no credit for selecting one or more incorrect events. 4a was a moderately difficult item, as 40% of students received full credit, 13% of students received partial credit, and the remaining 47% of students received no credit. In item 4b, students were asked to select the single event that would use a conditional IF-THEN programming structure from the list of items provided in 4a. They were then asked to explain why the event they selected would require an IF-THEN structure. This proved to be a difficult item, as 23% of students selected the correct event, and 16% of students selected the correct event and provided a suitable explanation. Please note that while item difficulty and discrimination index information on the subparts of items 3 and 4 is discussed in the text, the item difficulty and discrimination index information presented in **Table 8** is calculated for the full items, including all multiple choice and open-ended parts.

The mean score for the 95 students who took the Unit 4 assessment was 4.53 (SD = 2.50) out of 12 possible points, or 37.76%. Full item analysis on all Unit 4 assessment items is presented in **Table 8**. For items other than straightforward multiple choice where a simple correct vs. incorrect distinction was not applicable, the difficulty indices were calculated on the basis of students "passing" vs. "not passing" the item (i.e., earning a score of 60% or higher on a given item). For this assessment, a disproportionate number of items are hard per their difficulty indices. Two items demonstrate good discrimination indices and the other two demonstrate poor discrimination indices. It should be noted that we are aware that the teacher was rushed through the Unit 4 content; this may have contributed to the somewhat lower student performance observed on the Unit 4 assessment. These item analyses suggest that in future revisions of this assessment, test developers should consider including additional items that are less difficult.

DISCUSSION

Overall outcomes of and our experience throughout the assessment development process, guided by ECD and resulting in both the final assessments as well as the supporting documentation, will be discussed here as they align with each of the research questions.

> RQ1: How can we best utilize the well-established ECD framework to develop assessments that are tightly coupled to the stated learning goals of a redesigned curriculum?

The ECD framework promoted the development of robust assessments by requiring both rigor and attention to distilling the exact content to be addressed from the earliest steps of the process. Learning goals were taken directly from the curriculum website, and through the creation of the design pattern document, these learning goals were distilled into

FKSAs and accompanying tasks that collectively address the three key questions of ECD: what do we want students to demonstrate that they know and can do; what would evidence of the given knowledge and skill(s) entail; what tasks can we ask students to do that would yield that evidence? The previous discussion of item writing and design pattern development along with the presentation of sample items and their accompanying FKSA demonstrate the tight couplings between FKSAs, evidence, and tasks that are a key feature of a high-quality assessment with the capacity to measure the content intended to be measured.

A challenge when creating an assessment in a new space is that there is often limited guidance on what the important constructs are and how students progress in their learning of these constructs. An ECD approach to assessment development can provide guidance for addressing this challenge. The design documents, in particular the design pattern, provide supports for defining the important constructs to be measured, and aspects of tasks that can be used to measure these constructs. With CS concepts often being taught in tandem with other concepts, a design pattern is particularly useful as it provides a place to not only identify what is to be measured, but also what constructs should not be included. In the work described here, the use of an ECD approach helped the researcher to define the constructs of interest, identify appropriate item types, and develop tasks and rubrics.

One key challenge was determining the appropriate level of specificity for the FKSAs. If the FKSAs were overly specific, then this increased the number of FKSAs needed to provide coverage of the learning goals, and limited options for task development. If the FKSAs were not specific enough, then there was insufficient guidance on the task development to ensure that the important constructs were covered. This challenge was addressed by going through several drafts of many FKSAs, and receiving feedback on them from our co-author who has extensive design pattern experience. As work on the design pattern progressed across the four units, the appropriate scope and level of specificity for the FKSAs became more apparent and it became easier to write them consistently. For example, the FKSA from Unit 2 related to website errors was finalized as "Focus is on the skills of identifying and correcting errors in a website's design, organization, and/or layout in order to improve its functionality and effectiveness." The level of specificity with respect to website errors needed to be determined in writing this FKSA. Ultimately, we concluded that the FKSA should not specify a large number of individual errors that could be included, but rather should refer to three categories of errors (errors in design, organization, and layout). This decision allows the FKSA to highlight the different broad types of errors, but not be overly specific (and thus overly complicated and difficult to assess) in listing a large variety of possible individual errors.

The specification of the FKSAs provided a framework to map existing tasks which helped us to identify where we had tasks and where we needed tasks to provide coverage. Mapping tasks back to FKSAs once they were developed helped to ensure that the tasks developed measured the concepts we wanted to have measured. This helped ensure that the overall assessment had coverage of the critical components of the curriculum. The evidence and task models provide guidance to our item developers when developing the task which helped ensure that the resulting task was designed as intended. Overall, using the ECD framework to ensure coverage and alignment of our assessment provided guidance for item development and helped ensure that we were measuring the desired constructs.

> RQ2: How does the evidence generated through the ECD process and accompanying documentation support the validity argument for the newly developed assessments?

The use of ECD to rigorously articulate FKSAs that are tightly coupled to the learning goals of each unit, and then to design assessment tasks which directly generate evidence reflecting students' ability to interface with those FKSAs, ultimately results in an assessment that is likely to be measuring what the developers intended it to measure. Reviews of the assessment indicated there was alignment with the items, FKSAs and goals of the curriculum. Evidence from students showed that students were interacting with the tasks as expected: students were able to demonstrate ability related to the FKSAs and students who did not get full credit often displayed evidence of common misconceptions (e.g., in Unit 2, Item 2, most students made a single mistake in common, related to the location of an image in the header vs. the website background, and scored consistently high on all other website elements). In addition, the assessment data showed a range of scores reflective of student variance in ability.

Lessons Learned and Conclusions

The development and implementation of these newly written CS assessments, guided by utilization of the ECD framework, has conveyed several valuable lessons and issues around the design pattern and item writing components. In general, it is challenging to develop assessments while the learning goals and objectives are still under development and curriculum pilot testing is still on-going. Further, there is still much debate around defining each state's K-12 CS framework, which informs development of standards and curricula. This also leads to another challenge for assessment development in terms of defining the progression of student learning in the curriculum.

Several issues arose within the context of developing the design pattern that served as the underlying structure and guidance for item writing. Learning goals are critical to the creation of the design pattern, as they provide the basic information from which the FKSAs are derived. These assessments were developed within the context of a curriculum development, which underwent several iterations; learning goals were somewhat of a moving target as a result of addition, removal, and restructuring of the curriculum content that went on across these iterations. In some cases, this resulted in FKSAs and/or assessment items needing to be updated very close to the time of assessment administration. Using ECD on a curriculum being developed and iterated on in real time presents a set of unique challenges and requires some flexibility in production of the design pattern document as well as in assessment development.

Also, while ECD was useful in helping us pick FKSAs and structure assessment tasks, tasks still need to undergo additional piloting and testing. In particular for our Unit 4 assessment, further investigation is needed to determine if students were not taught the concepts, or if revisions to the items could help improve the clarity of the expectations of students for these tasks. While the concepts that are being measured would not change, there may be ways to further scaffold tasks for students that would provide tasks better able to discriminate between students.

In summary, creating and iterating upon the design pattern document provided a fruitful avenue for communication around and clarification of the curriculum goals and FKSAs, and promoted targeted discussions among curriculum developers, assessment writers, and our ECD expert co-author. Following the ECD process resulted in assessment items that were clearly tied to FKSAs, and ensured that the overall assessment strategy covered all FKSAs deemed most critical for assessment. The resulting assessments provided evidence of student challenges and resulted in a useful tool for the researchers. While ECD does not remove all of the challenges of assessment development, its use highlights the decisions that need to be made and supports developers in generating an assessment that matches the desired purpose.

DATA AVAILABILITY STATEMENT

The datasets presented in this article are not readily available because, The datasets generated and/or analyzed during the current study are not publicly available given that our data agreements with the schools in which the data were collected do not allow them to be shared publicly. Requests to access the datasets should be directed to not applicable.

ETHICS STATEMENT

The studies involving human participants were reviewed and approved by the Georgia Institute of Technology Institutional Review Board (IRB). Written informed consent to participate in this study was provided by the participants' legal guardian/next of kin.

AUTHOR CONTRIBUTIONS

SN contributed to the writing and editing of all sections of the manuscript, analysis of data, and construction of tables and figures. MA contributed to the writing and editing of all sections of the manuscript, analysis of data, and construction of tables and figures. DR contributed to the writing of the introductory sections of the manuscript, specifically focusing on the introductory sections related to ECD. DR also contributed to the editing of all sections of the manuscript. DR provided feedback on the ECD documentation and assessments. DE contributed to the writing of the introductory sections of the manuscript. DE is one of the writers of the curriculum on which this research is based. MH contributed to the writing of the introductory sections of the manuscript. MH is one of the writers of the curriculum on which this research is based. DH contributed to the writing of the introductory sections of the manuscript. DH is one of the writers of the curriculum on which this research is based. MU contributed to the writing of the introductory sections of the manuscript. MU is one of the writers of the curriculum on which this research is based.

FUNDING

This material is based upon work supported by the National Science Foundation under Grant No. 1639946. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

ACKNOWLEDGMENTS

We would like to acknowledge the hard work of the research coordinator Emily B. Frobos for ensuring the data quality. We would like to acknowledge Mike Ryan for his contributions to writing the curriculum on which this research is based. Lastly, this work would not be possible without the dedicated work of our CS teachers.

REFERENCES

Arpaci-Dusseau, A., Griffin, J., Kick, R., Kuemmel, A., Morelli, R., Muralidhar, D., Osborne, R. B., Uche, C., Astrachan, O., Barnett, D., Bauer, M., Carrell, M., Dovi, R., Franke, B., Gardner, C., and Gray, J. (2013). "Computer Science Principles," in Proceeding of the 44th ACM technical symposium on Computer science education (Denver, CO: (SIGCSE)), 251–256. doi:10.1145/2445196.2445273

Astrachan, O., and Briggs, A. (2012). The CS Principles Project. *ACM Inroads* 3 (2), 38–42. doi:10.1145/2189835.2189849

Basu, S., Rutstein, D., and Tate, C. (2021). *Building Teacher Capacity in K-12 Computer Science by Promoting Formative Assessment Literacy*. Rockville, MD: National Comprehensive Center.

Bienkowski, M., Snow, E., Rutstein, D. W., and Grover, S. (2015). *Assessment Design Patterns for Computational Thinking Practices in Secondary Computer Science: A First Look (SRI Technical Report)*. Menlo Park, CA: SRI International. Retrieved from: http://pact.sri.com/resources.html.

Bechard, S., Clark, A., Swinburne Romine, R., Karvonen, M., Kingston, N., and Erickson, K. (2019). Use of Evidence-Centered Design to Develop Learning Maps-Based Assessments. *Int. J. Test.* 19 (2), 188–205. doi:10.1080/15305058.2018.1543310

Chapelle, C. A., Schmidgall, J., Lopez, A., Blood, I., Wain, J., Cho, Y., et al. (2018). *Designing a Prototype Tablet-Based Learning-Oriented Assessment for Middle School English Learners: An Evidence-Centered Design approach Educational Testing Service Report*. ETS RR-18-46

Cizek, G. J. (2010). "An Introduction to Formative Assessment: History, Characteristics and Challenges," in *Handbook of Formative Assessment*. Editors H. L. Andrade and G. J. Cizek (Taylor & Francis), 3–17.

Denning, P. (2017). Remaining Trouble Spots with Computational Thinking. *Commun. ACM* 60 (6), 33–39. doi:10.1145/2998438

Dixson, D. D., and Worrell, F. C. (2016). Formative and Summative Assessment in the Classroom. *Theor. Into Pract.* 55 (2), 153–159. doi:10.1080/00405841.2016.1148989

Grover, S., and Basu, S. (2017). *Measuring Student Learning in Introductory Block-Based Programming: Examining Misconceptions of Loops, Variables, and Boolean Logic*. Seattle WA: SIGCSE. doi:10.1145/3017680.3017723

Heintz, F., Mannila, L., and Färnqvist, T. A. (2016). "Review of Models for Introducing Computational Thinking, Computer Science and Computing in K-12 Education," in Proc. IEEE Frontiers in Education Conference (FIE). doi:10.1109/fie.2016.7757410

Hu, Y., Wu, B., and Gu, X. (2017). Learning Analysis of K-12 Students' Online Problem Solving: a Three-Stage Assessment Approach. *Interactive Learn. Environments* 25 (2), 262–279. doi:10.1080/10494820.2016.1276080

K-12 Cs Framework Committee (2016). *K-12 Computer Science Framework*. Retrieved March, 2020 from http://www.k12cs.org.

Kelley, T. L. (1939). The Selection of Upper and Lower Groups for the Validation of Test Items. *J. Educ. Psychol.* 30, 17–24. doi:10.1037/h0057123

Khattri, N., Reeve, A. L., Kane, M. B., and Adamson, R. (1995). *Studies of Education Reform: Assesment of Student Performance*. Washington, D.C.: U. S. Department of Education, Office of Educational Research and Improvement.

Livingston, S. A. (2009). *Constructed-response Test Questions: Why We Use Them; How We Score Them. (ETS R & D Connections, RD-11-09)*. Princeton, NJ: Educational Testing Service.

Lukhele, R., Thissen, D., and Wainer, H. (1994). On the Relative Value of Multiple-Choice, Constructed Response, and Examinee-Selected Items on Two Achievement Tests. *J. Educ. Meas.* 31 (3), 234–250. doi:10.1111/j.1745-3984.1994.tb00445.x

Mislevy, R. J., and Haertel, G. (2006). *Implications of Evidence-Centered Design for Educational Testing*. Menlo Park, CA: SRI International.

Mislevy, R. J., and Riconscente, M. M. (2006). "Evidence-centered Assessment Design: Layers, Concepts, and Terminology," in *Handbook of Test Development*. Editors S. Downing and T. Haladyna (Mahwah, NJ: Erlbaum), 61–90.

Mislevy, R. J., Steinberg, L. S., and Almond, R. G. (2003). *On the Structure of Educational Assessments*. Los Angeles, CALos Angeles: Center for the Study of Evaluation, National Center for Research on Evaluation, Standards, and Student Testing, Graduate School of Education & Information Studies, University of California.

Mislevy, R. J. (2007). Validity by Design. *Educ. Res.* 36 (8), 463–469. doi:10.3102/0013189x07311660

Oliveri, M. E., Lawless, R., and Mislevy, R. J. (2019). Using Evidence-Centered Design to Support the Development of Culturally and Linguistically Sensitive Collaborative Problem-Solving Assessments. *Int. J. Test.* 19 (3), 270–300. doi:10.1080/15305058.2018.1543308

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., and Clark, D. (2013). Integrating Computational Thinking with K-12 Science Education Using Agent-Based Computation: A Theoretical Framework. *Educ. Inf. Technol.* 18 (2), 351–380. doi:10.1007/s10639-012-9240-x

Snow, E. (2016). Release of the ECS Assessments – Cumulative (Units 1-4) Assessments Available Now! [Blog post]. Retrieved from:https://www.csforallteachers.org/blog/release-ecs-assessments-cumulative-units-1-4-assessments-available-now.

Snow, E., Rutstein, D. W., Basu, S., Bienkowski, M., and Everson, T. (2019). Leveraging Evidence-Centered Design to Develop Assessments of Computational Thinking Practices. *Int. J. Test.* 19 (2), 103–127. doi:10.1080/15305058.2018.1543311

Snow, E., Rutstein, D. W., Bienkowski, M., and Xu, Y. (2017). "Principled Assessment of Student Learning in High School Computer Science," in Proceedings of International Conference on Computer Science Education Research, Tacoma, WA, August 2017 (ICER'17), 8. doi:10.1145/3105726.3106186

So, H.-J., Jong, M. S.-Y., and Liu, C.-C. (2020). Computational Thinking Education in the Asian Pacific Region. *Asia-pacific Edu Res.* 29, 1–8. doi:10.1007/s40299-019-00494-w

Tang, X., Yin, Y., Lin, Q., Hadad, R., and Zhai, X. (2020). Assessing Computational Thinking: A Systematic Review of Empirical Studies. *Comput. Education* 148, 103798. doi:10.1016/j.compedu.2019.103798

Webb, M., Davis, N., Bell, T., Katz, Y. J., Reynolds, N., Chambers, D. P., et al. (2017). Computer Science in K-12 School Curricula of the 2lst century: Why, what and when? *Educ. Inf. Technol.* 22, 445–468. doi:10.1007/s10639-016-9493-x

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.