Journal of Computing in Higher Education
https://doi.org/10.1007/s12528-024-09425-1

Examining computational thinking across disciplines in higher education classrooms: learning outcomes from student-generated artifacts

Yifan Zhang[1] · Amanda Mohammad Mirzaei[2] · Chrystalla Mouza[3] · Lori Pollock[4] · Kevin Guidry[5]

Accepted: 7 December 2024

Abstract

To meet the demands of 21st-century societies and economies, faculty across disciplines must engage college and university students with course activities and assignments that foster the development of computational thinking (CT) skills. Toward this end, examining the ways in which CT can be infused into general education courses has been a topic of recent research. However, the question remains about how students in non-computer science courses can use CT skills in course assignments across disciplines. Guided by a rubric aimed to evaluate the development of CT skills including decomposition, algorithms, data, and abstraction, we examined 101 student-generated artifacts in undergraduate courses across four disciplines: mathematics, sociology, music, and English. In this work, we report on assignment prompts and overall CT skills exhibited by participating students. While some disciplines may not fully facilitate the development of *all* CT skills, a range of these skills was reflected in student artifacts. We present representative examples demonstrating CT skill development across various levels, including capstone level (score 4), milestone (score 3), benchmark (score 1), and no usage (score 0). The findings of this work provide insights into ways in which higher education faculty can design assignment prompts that support and scaffold students' development of CT skills, as well as how students across disciplines respond to CT prompts. Findings also have implications for the design of CT-related assessment instruments.

Keywords Computational thinking · Undergraduate education · Impacts of computing · Higher education · General education

Extended author information available on the last page of the article

Published online: 26 February 2025

Introduction

The computing education community traced the term computational thinking (CT) back to Papert in 1980, but it was not broadly studied for many years (Saqr et al., 2021). Wing's (2006) seminal article, *Computational Thinking*, triggered increasing research into CT across all school stages, from K-12 to higher education (Apiola et al., 2022; Ilic et al., 2018). Wing and other researchers argued that CT is a problem-solving methodology that can be applied across disciplines to improve efficiency, and thus, it is a skill for everyone in the twenty-first century (Barr et al., 2011; Mohaghegh & McCauley, 2016; Wing, 2006). In fact, not only academia but also the government (Hsu et al., 2019) and industry (Vergara et al., 2009) reinforced the value of graduates in various disciplines with CT skills.

Although much of the discussion around the development of CT skills has been focused on K-12 students (Guzdial, 2015; Ilic et al., 2018), higher education researchers have increasingly focused on how CT can be infused into general education courses or disciplines outside of computer science (CS) (Lyon & Magana, 2020). For example, DePaul University (Perković, 2010), Towson University (Dierbach et al., 2011), Carroll University (Kuster et al., 2011), and Brock University (Jaipal-Jamani & Angeli, 2017) all conducted pilot studies on infusing CT skills into disciplines such as science, music, art, history, and education. While these studies provide insight into the integration of CT in higher education, the focus is primarily on curriculum design and piloting, and less on *what* and *how* students learn from these efforts. Thus, there is a lack of research that focuses explicitly on examining student CT outcomes in higher education classrooms as a means of evaluating the efforts where faculty in non-CS disciplines are working toward CT integration. To explore how university students in CT-infused courses outside of CS acquire CT skills, we ask the following research question: *In what ways and to what extent do students integrate CT in response to disciplinary course assignment prompts outside of CS?*

Guided by a previously developed CT rubric (see Appendix A) with a focus on decomposition, algorithms, data, and abstraction, we examined 101 student artifacts from 12 assignments across 5 undergraduate courses in 4 disciplines: mathematics, sociology, music, and English. We first present the CT-specific assignment prompts developed by faculty in non-CS disciplines. Subsequently, we present the distribution of overall CT skills demonstrated by students and highlight representative artifacts across a range of CT skills. Our findings have implications for the infusion of CT across disciplines in higher education, by demonstrating ways in which students engage with CT prompts. They also have implications for future research and curriculum design considerations.

Background

In this section, we introduce related literature from three perspectives, including definitions of CT, pedagogical approaches for developing CT in higher education, and assessment of students' CT in higher education.

Computational thinking: literature and definition in the context of this work

In her seminal paper, Wing (2006) suggested that CT is just as important to teach students as the ‘three R’s’ (i.e., reading, writing, and arithmetic) and that given the widespread use of computers, teaching CT skills is as essential now as literacy after the printing press became widely used. As she notes, CT “involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science” (p.33). Similar to Wing, other researchers (e.g., Denning, 2007; DiSessa, 2018; Li et al., 2020) advocated that CT is a thinking process that requires more abstraction than just thinking like a computer; this abstraction is a skill that all people need to develop. Although various definitions of CT have been proposed, they all can be categorized into two groups (Tang et al., 2020): those related to programming and computing concepts (Brennan & Resnick, 2012; Denner et al., 2014; Weintrop et al., 2016), and those focusing on competencies needed in both domain-specific knowledge and general problem-solving skills (Selby & Woollard, 2013; Yadav et al., 2014). This paper adapted the latter definition, treating CT as a problem-solving skill that could benefit different disciplines.

The way CT is defined is important because it impacts the design of course assignments. A number of studies have found that while higher education faculty utilize diverse computational assignments across disciplines, many still use a programming approach, indicating a strong association of CT with programming (Hsu et al., 2018; Jong & Jeuring, 2020; Shute et al., 2017; Tang et al., 2020). However, several scholars argued about the importance of not restricting CT to an association with programming or CS professionals, to make CT truly accessible across disciplines (e.g., Jong & Jeuring, 2020; Li et al., 2020; Lu & Fletcher, 2009; Tang et al., 2020). One of the challenges in separating CT from programming lies in the difficulty of assessing the success of CT initiatives as few standard instruments exist that measure students’ understanding of CT (Jong & Jeuring, 2020). In addition to this challenge, several researchers (e.g., Fennell et al., 2020; Grover & Pea, 2013; Kalelioglu et al., 2016) also concluded that while the mantle of infusing CT across disciplines is being taken up by higher education institutions (and that there are benefits for students), empirical studies need to be conducted to examine the extent to which students are acquiring CT skills in these disciplines. Such assessments need to move beyond programming itself, to include additional CT skills that could benefit students, such as decomposition, algorithmic thinking, data, and abstraction (Hsu et al., 2018; Selby & Woollard, 2013; Shute et al., 2017). This paper addresses the aforementioned research gap by examining CT development beyond programming using a research-based assessment rubric previously developed by our team (Guidry et al., 2019; Pollock et al., 2019).

Pedagogical approaches for developing and assessing CT in higher education

In one of the early studies in the field, Dierbach and colleagues (2011) infused CT into general education courses in a higher education institution by altering the existing non-STEM curriculum, rather than developing a new CT-focused curriculum. Their approach established common CT course objectives, encompassing 5 parts: (a) develop, evaluate, and apply computational models to problems; (b) create, apply, and determine algorithmic techniques for problems; (c) distinguish between problems that can and cannot be solved computationally; (d) evaluate a given algorithm or model; and (e) apply and reflect on the application of CT methods and models. Several courses described in their work also adapted course-specific learning objectives based on these common CT course objectives. To gauge student learning, they asked instructors to use a "computational thinking grading rubric" to map assignment solutions, student responses, and student discussion. Findings indicated that CT could be successfully adapted for non-STEM courses, and at least 75% of students who enrolled in these courses demonstrated knowledge of CT at the end of the course. Findings also indicated significant support from students, instructors, and administrators regarding the value of non-STEM courses that infuse CT. However, instructors did not apply the rubric to all assignments, nor did they show examples of student-developed artifacts to demonstrate the extent in which students understood CT.

Other researchers have had similar success designing new courses that integrate CT. For example, Kuster et al. (2011) restructured the Bachelor of Science degree (B.S.) at a higher education institution to include a more meaningful understanding of CT. In addition to implementing a new CS major and minor, they also infused CT into CS courses that STEM programs outside of CS might require. Similarly, Guzdial (2008) called for specific computer languages created to help students not pursuing degrees in CS engage with CT. Regardless of how higher education institutions might choose to implement CT into their program pathways, it is clear from the above examples that some are answering the call to incorporate CT skills across different curricula and disciplines.

In addition to examining how students understand and acquire CT skills, it is important that researchers also consider how instructors are prepared to infuse CT into their teaching. Angeli and Giannakos (2020), Dierbach et al. (2011), Kuster et al. (2011), and Yadav et al.(2017), all described how they prepared higher education faculty for CT integration, including generating interest, developing discipline-independent CT materials, establishing faculty collaboration, developing discipline-specific course materials, piloting the initial courses, and engaging in evaluation and revision. They also identified existing challenges associated with faculty preparation for CT integration, such as buy-in, interdisciplinary content knowledge, course quality improvement, well-developed textbooks, and administrative support. Yet much of the research for preparing instructors to infuse CT into disciplinary content is directed to K-12 levels through professional development for in-service teachers (Yadav et al., 2017). Thus, more work is needed to understand CT-infused practices in higher education with the potential to foster the development of CT skills among all college students, including those in non-STEM disciplines.

Regardless of how CT is infused and how instructors are prepared to create learning environments that support CT in non-CS courses, it is important to understand how it might be beneficial to students. Typical assessment of CT infusion includes traditional testing, questionnaires, portfolios, interviews, and surveys

(Oliveira et al., 2022; Tang et al., 2020). Additionally, many of the portfolios analyzed in non-CS disciplines are programming-based (e.g., Caballero et al., 2012; Rojas-López & García-Peñalvo, 2018; Romero et al., 2017). These studies suggest that there are many ways to assess student understanding of CT when there is programming involved; however, there is a need for the field to understand how we can assess students' understanding of CT in non-programming-based assignments. There is indeed literature that suggests that non-CS or non-STEM students benefit from infusing CT into general education courses (Dierbach et al., 2011); however, work remains to be done to understand *how* students demonstrate their understanding. This study seeks to address this gap by presenting student work based on course assignments that infuse CT into non-CS higher education courses using practices beyond programming.

Methodology

This section describes our study methodology, including study participants, data collection, and analysis. We briefly describe our participants including the CT integration team, non-CS faculty instructors, and the undergraduate students enrolled in the courses under examination. In addition to student artifacts which are central to our work, we also describe the assignments that non-CS faculty instructors designed to foster the development of CT skills. We extracted samples from the data set and then used a rubric to assess and analyze those student-generated artifacts. All materials analyzed are de-identified and all procedures were approved by our institutional review board (IRB).

Participants

The participants in this study included a CT integration team, non-CS faculty instructors, undergraduate students serving as CT fellows, and undergraduate students enrolled in selected courses (see Table 1). The CT integration team provided leadership for this work, and included faculty from CS, a learning scientist, and professionals from the university's Center for Teaching & Assessment of Learning. The site for this study was a large, public research university in the Mid-Atlantic region of the United States.

With the goal of integrating CT skills into discipline-specific courses, five non-CS faculty teaching undergraduate courses in mathematics, sociology, music, and English worked with the CT integration team during the planning and implementation of their courses. Faculty participants were recruited by the CT integration team based on the following criteria: (a) they represented different disciplines, including STEM (e.g., mathematics) and humanities (e.g., sociology), but not including CS; (b) they taught introductory courses serving a large number of students (e.g., English); (c) they preferably taught courses required for degree completion in the respective majors (e.g., English and mathematics); and (d) they were

Table 1 Participants' experience

Participants	Professional experience	CT experience
CT integration leadership team	1 CS professor, 1 learning sciences professor, and 2 professionals from the university's Center for Teaching & Assessment of Learning	Multiyear experience in CT research and professional development. Provided leadership for the project
Non-CS faculty participants	7 faculty (5 faculty are included in our dataset, 2 faculty are not included because their course was developed before the rubric matured)	Recruited from both STEM and non-STEM majors; participated in professional development for CT and bi-weekly meetings with CT integration team; met with CT Fellows weekly
Undergraduate CT Fellows	5 fellows	Pursued a STEM major and had prior coursework in CS; enrolled in a university course called Field Experiences in Teaching Computing; met with faculty participants weekly
Undergraduate students	219 students	Self-enrolled in CT-integrated courses

willing to participate in professional development and bi-weekly meetings with the CT integration team (see also Pollock et al., 2019). Faculty participants were monetarily compensated for their participation upon completion of the work.

In our dataset, three of the five faculty instructors infused CT into existing introductory courses, including *Contemporary Mathematics* for non-majors (mathematics instructor), *Introduction to Sociology* for honors first-year students (sociology instructor), and a first-year writing course titled *Seminar in Composition* (English instructor). The two music instructors infused CT into *Fundamentals of Music I* and consequently developed a new course, called *Computational Thinking in Music*. These two music courses were heavily influenced by the book "Computational Thinking in Sound" (Greher & Heines, 2014). All courses were delivered face-to-face and ran for the entire semester, either Fall or Spring (typically 15 weeks). There were also two other faculty who participated in this work, but their courses are not included in our dataset—a music professor who also taught *Fundamentals of Music I* and an education professor who taught an education course cross-listed in communication named *Multimedia Literacy*. Their courses served as pilot implementation; they were developed at the beginning stage of this work when the assessment rubric was not mature.

Additionally, we recruited five undergraduate students called *CT Fellows* who were responsible for providing faculty support in both the planning and implementation of CT in the classroom. All CT Fellows pursued a STEM major and had prior coursework in CS. CT Fellows met with faculty participants every week and attended selected course meetings. They subsequently helped faculty design course assignments, identify potential technological resources or tools needed to complete the assignments, and acquire technical skills as needed. Additionally, CT Fellows provided technical support to undergraduates enrolled in the selected courses. As part of their participation in this work, CT Fellows also enrolled in a university course called *Field Experiences in Teaching Computing*, taught by two of the authors. Participants in the course met weekly for 75 min. The field experience course provided the vehicle for connecting CT Fellows' knowledge of computing to disciplinary content and pedagogy. It also facilitated reflection on the implementation of CT-infused activities by participating faculty. In particular, the weekly class meetings focused on (a) CS resources and tools aligned with big ideas in computing, (b) CS pedagogy, and (c) reflections on the work with participating faculty (Mouza et al., 2016, 2023; Pollock et al., 2015). The overall objective was to equip CT Fellows with pedagogical knowledge they can use to support faculty directly in the classroom as they planned and implemented CT-integrated assignments.

Finally, participants included a total of 219 undergraduate students (N = 219) enrolled in the 5 CT-infused courses that served as the basis of this work. Specifically, participants included: 22 students enrolled in *Contemporary Mathematics*, 10 students enrolled in *Introduction to Sociology*, 72 students enrolled in the writing *Seminar on Composition*, 35 students enrolled in *Computational Thinking in Music*, and 80 students enrolled in *Fundamentals of Music I*. We do not have the demographics of these student participants because the student artifacts were provided by faculty without personally identifiable information consistent with IRB procedures.

Data collection

Non-CS instructor-developed assignments

With the support of the CT integration team, non-CS faculty instructors infused CT into various assignments during the semester-long courses. These assignments were designed to prompt students to use and demonstrate knowledge of CT within specific disciplines. We first collected all instructor-developed course assignments and associated files. We used the course syllabus and assignment descriptions to gain an understanding of the context of the students' work. Table 2 presents relevant information for each assignment. Here we describe each of the courses and integrated assignment, first from the three instructors who infused CT into existing courses, and then from the two instructors who designed music courses with CT as a central focus.

Contemporary mathematics This course was designed as an alternative to the traditional three-credit college algebra course to meet mathematics general education requirements. The learning objectives of the course included: (1) to learn how mathematical algorithms are applied to investigate real-world problems; (2) to construct, evaluate, and interpret data; and (3) to appreciate how mathematics is used in a variety of disciplines. In this course, the instructor provided a data-related assignment utilizing Statcrunch,[1] a web-based statistical software. Students were asked to select an online dataset and develop exploratory questions about it. They then used the dataset to generate statistical graphs and summaries to help answer the questions they had posed. Finally, they were asked to summarize and explain their findings.

Introduction to sociology This course aims to provide an overview of the sociological perspective of the study of society, social organization and social institutions with special emphasis on the social causes and consequences of human behavior. It can be used to satisfy general university breadth requirements in social and behavioral sciences. In this course, the faculty instructor developed two different CT-infused assignments: *Gender Display Algorithms* and *PolicyMap*. The *Gender Display Algorithms* assignment asked students to research websites that market toys to children. Students identified patterns across these websites that they thought might cater toward gender (here a binary construct: boy/girl). They then used these patterns to sketch algorithms they thought could determine if other websites were gendered toward boys or girls. The *PolicyMap* assignment had two parts. The first part contained two exercises using PolicyMap,[2] a tool that accesses and visualizes data about communities across the United States. The instructor also decomposed these exercises into smaller subproblems for students to get familiar with the tool, as well as CT decomposition and data skills. The second part asked students to pose a question about a construct they

1 https://www.statcrunch.com/

2 https://www.policymap.com/

could research using PolicyMap and to write a paragraph on how the CT skill of decomposition could be used to help them answer their questions.

English writing: Seminar in Composition This General Education course provides an introduction to the process of academic writing that centers on the composition of analytical, research-based essays. The faculty instructor developed an assignment asking students to reflect on their writing practices over the course of the semester. Students were asked to describe how they might have used CT in their writing; specifically, if and how they broke down assignments to complete them and any other strategies they might have used to make the writing process more efficient.

Fundamentals of music I and computational thinking in music The two music courses developed around improving students' CT skills were *Fundamentals of Music I* and *Computational Thinking in Music*. They both explored CT as a means to better understand, model, and communicate musical processes and design, and how humans interact with music. Instructors here provided us with 7 and 4 assignment descriptions, respectively. For the purpose of this work, we are only focusing on assignments with student artifacts provided by instructors (see Table 2). In addition, *Fundamentals of Music I* was a project-based course where groups of students documented their work and shared reflections on websites they created.

Student CT-integrated artifacts

Student artifacts were provided by faculty instructors in conjunction with the corresponding assignments (however, not all assignments have student artifacts). Of the 182 student-generated artifacts provided by instructors, a sampling process was used, as follows, resulting in the analysis of 101 artifacts for this work: (a) in courses where only a small number of artifacts were available, all were included (i.e., *Contemporary Mathematics* and *Computational Thinking in Music*); and (b) in courses where a large number of artifacts from the same assignment were available, we randomly chose approximately 50% from each assignment set. Table 2 presents an overview of assignment descriptions and student CT-integrated artifacts that were collected and analyzed in this study.

Data analysis

We read and reviewed instructor-developed assignments to establish the context. We coded our sampled student artifacts using a CT rubric previously developed by our team (Guidry et al., 2019; Pollock et al., 2019; see Appendix A for the full rubric). This rubric follows the Association of American Colleges and Universities VALUE rubrics (AAC&U, 2018) and sets up criteria for four CT dimensions, including *decomposition*, *algorithms*, *data*, and *abstraction*. The rubric articulates fundamental criteria for each learning outcome with performance descriptors demonstrating progressively more sophisticated levels of attainment. Specifically, it assesses the type and level of CT skills illustrated in each artifact using the following scale: 1

Table 2 Student CT-integrated artifacts collected and analyzed

Course and assignment	Summary of assignment description	Sample size/full size	Student exhibited CT skills
Contemporary mathematics			
Data analysis using Statcrunch	Research often involves formulating interesting questions, developing a methodology, and collecting data to answer those questions. Statcrunch was used to select data and generate both numeric summaries and graphs representing the data	8/8 artifacts	Decomposition Algorithms Data
Introduction to sociology			
Gender display algorithms	Students browsed the Internet for sites that sell chidlren's toys, and identified three that are highly gendered towards "boys" or "girls". Students wrote an algorithm to tell a machine how to identify gender on children's websites if it were to search the web for them	10/22 artifacts	Decomposition Algorithms
PolicyMap activity	Students chose a sociological topic of interest and utilized PolicyMap to examine the topic and produce a single layer map for a specific geographic area. Subsequently, they explored one additional social, economic, or political indicator related to the initial topic and produced a 3-layer map using these indicators	10/22 artifacts	Decomposition Data
PolicyMap reflection	Students wrote a brief paragraph at the end of the PolicyMap activity that described how they understood the process of decomposition in their work	6/11 artifacts	Decomposition Data
Fundamentals of music I			
Project 1: breaking poetry & building music	Students broke well known poems into smaller pieces and then created music	4/8 artifacts	Decomposition Algorithms Abstraction
Project 3: samples of campus	Students used audio-editing software to compose a piece of music using samples taken from campus or the surrounding town	4/16 artifacts	Abstraction
Projects website	Students created a website containing webpages of processes and reflections for all projects completed in the course	8/18 web pages	Decomposition Algorithms Abstraction

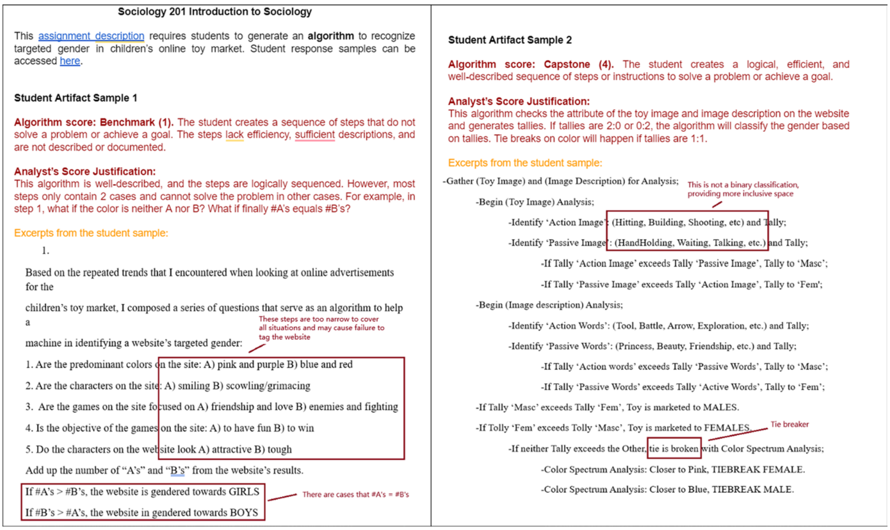
Table 2 (continued)

Course and assignment	Summary of assignment description	Sample size/full size	Student exhibited CT skills
Computational thinking in music			
Reflection on Scratch reconstruction project	Students reflected on the ways in which they put together the blocks they programmed in Scratch, an object oriented programming language	10/10 artifacts	Decomposition Algorithm
Decomposition work	Students described how to use a previously created music dataset to form a new song and reflected on the ways the skill of decomposition could help	3/3 artifacts	Algorithms
Algorithmic composition	Students wrote on ways they could describe how to turn lyrics into music. They were prompted by questions like, *"What musical parameters do you need to consider?"* and *"Describe in detail a series of ordered steps for programmers"*	3/3 artifacts	Decomposition Algorithms
Semester reflections	Students chose one of the CT learning outcomes, defined it in their own words, and reflected on how they connected this outcome to the work they have done in various projects over the course of the semester	10/10 artifacts	Decomposition Data
English writing: seminar in composition			
CT skill reflection on writing	Students wrote a reflection on both the processes of writing and the actual essay that they wrote and connected writing to CT skills	25/51 artifacts	Decomposition Algorithms

(benchmark), 2 and 3 (milestone), and 4 (capstone). The scale (1–4) and accompanied descriptors (benchmark—capstone), are used consistently across all VALUE rubrics to facilitate comparisons among undergraduate students in multiple institutions. Although this rubric has not been approved by AAC&U, it follows the same methodology as it is intended for use across an entire institution. In the context of this work, the rubric was used in two ways: as a scaffold to help instructors design CT-related course content and assignments, and as an analytic framework for assessing CT in student-generated artifacts.

Our data analysis progressed in the following workflow: (a) read through the assignment description to get a sense of what the assignment is prompting, paying attention to the assignment name and CT skills identified in the description; (b) read through the student artifacts and identified instances that demonstrated each of the four CT skills, regardless of whether the assignment included an explicit prompt for the use of a given skill; (c) referring to the rubric, assigned a holistic score to the artifact for each CT skill, recording the reasoning for the score; and (d) created documents for each assignment, including the assignment description, all student artifacts for the assignment, rubric scores for each artifact, and reasoning behind scores. As an example, Figs. 1 and 2 demonstrate how we used the rubric for sociology and math assignments.

For each assignment, two authors first randomly selected and individually coded artifacts (n = 10) using the CT rubric. The agreement from this round of coding was 85%. Subsequently, results were shared and discussed with two additional authors to ensure a clear understanding and application of the CT rubric. After reaching a 100% agreement regarding the coding of the selected ten artifacts, the first author continued to code the rest of the data (n = 91).



Sociology 201 Introduction to Sociology

This assignment description requires students to generate an **algorithm** to recognize targeted gender in children's online toy market. Student response samples can be accessed here.

Student Artifact Sample 1

Algorithm score: Benchmark (1). The student creates a sequence of steps that do not solve a problem or achieve a goal. The steps lack efficiency, sufficient descriptions, and are not described or documented.

Analyst's Score Justification:
This algorithm is well-described, and the steps are logically sequenced. However, most steps only contain 2 cases and cannot solve the problem in other cases. For example, in step 1, what if the color is neither A nor B? What if finally #A's equals #B's?

Excerpts from the student sample:

1.

Based on the repeated trends that I encountered when looking at online advertisements for the

children's toy market, I composed a series of questions that serve as an algorithm to help a

machine in identifying a website's targeted gender:

These steps are too narrow to cover all situations and may cause failure to tag the website

1. Are the predominant colors on the site: A) pink and purple B) blue and red

2. Are the characters on the site: A) smiling B) scowling/grimacing

3. Are the games on the site focused on A) friendship and love B) enemies and fighting

4. Is the objective of the games on the site: A) to have fun B) to win

5. Do the characters on the website look A) attractive B) tough

Add up the number of "A's" and "B's" from the website's results.

If #A's > #B's, the website is gendered towards GIRLS

If #B's > #A's, the website in gendered towards BOYS

There are cases that #A's = #B's

Student Artifact Sample 2

Algorithm score: Capstone (4). The student creates a logical, efficient, and well-described sequence of steps or instructions to solve a problem or achieve a goal.

Analyst's Score Justification:
This algorithm checks the attribute of the toy image and image description on the website and generates tallies. If tallies are 2:0 or 0:2, the algorithm will classify the gender based on tallies. Tie breaks on color will happen if tallies are 1:1.

Excerpts from the student sample:

-Gather (Toy Image) and (Image Description) for Analysis;

This is not a binary classification, providing more inclusive space

-Begin (Toy Image) Analysis;

-Identify 'Action Image': (Hitting, Building, Shooting, etc) and Tally;

-Identify 'Passive Image': (HandHolding, Waiting, Talking, etc.) and Tally;

-If Tally 'Action Image' exceeds Tally 'Passive Image', Tally to 'Masc';

-If Tally 'Passive Image' exceeds Tally 'Action Image', Tally to 'Fem';

-Begin (Image description) Analysis;

-Identify 'Action Words': (Tool, Battle, Arrow, Exploration, etc.) and Tally;

-Identify 'Passive Words': (Princess, Beauty, Friendship, etc.) and Tally;

-If Tally 'Action words' exceeds Tally 'Passive Words', Tally to 'Masc';

-If Tally 'Passive Words' exceeds Tally 'Active Words', Tally to 'Fem';

-If Tally 'Masc' exceeds Tally 'Fem', Toy is marketed to MALES.

-If Tolly 'Fem' exceeds Tolly 'Masc', Toy is marketed to FEMALES.

Tie breaker

-If neither Tally exceeds the Other, tie is broken with Color Spectrum Analysis;

-Color Spectrum Analysis: Closer to Pink, TIEBREAK FEMALE.

-Color Spectrum Analysis: Closer to Blue, TIEBREAK MALE.

Fig. 1 Scoring of student artifact samples. Sample 1 (Left) shows score 1 for algorithms and sample 2 (Right) shows score 4 for algorithms (*Gender display algorithms* assignment)

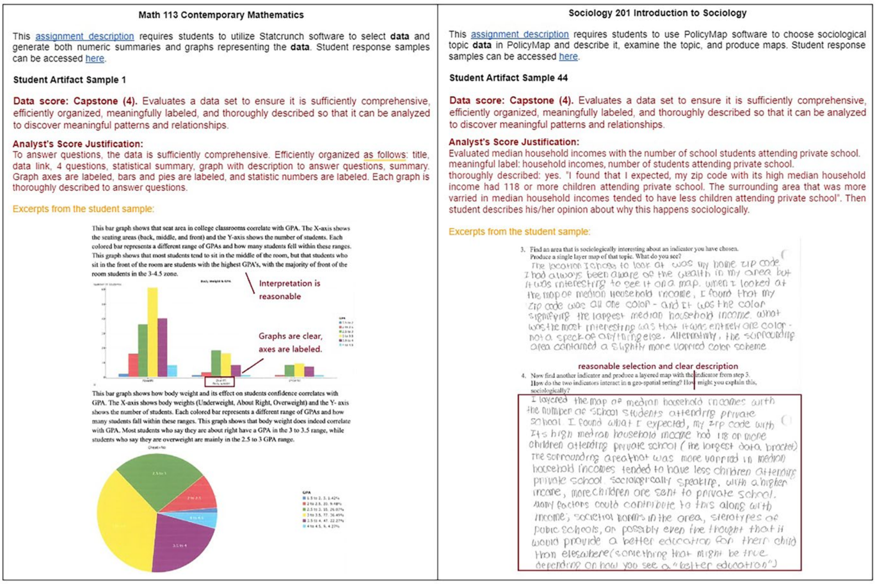


Fig. 2 Example of scoring of student artifact samples. Sample 1 (Left) shows score 4 for data (*Data analysis using Statcrunch* assignment) and sample 4 (Right) shows score 4 for data (*PolicyMap activity* assignment)

In the initial coding, we noted that some artifacts did not illustrate any CT skills. However, since the rubric starts at benchmark (score 1), this created challenges with coding and inter-rater reliability. Upon discussion, we decided to provide a score of 0 if: (a) the CT skill (i.e., decomposition, algorithms, data, or abstraction) was completely absent from the artifact; or (b) the artifact included specific keywords such as "break down" for decomposition or "steps" for algorithms; however, the artifact as a whole did not illustrate any further description fitting into the benchmark level of the rubric. This decision is consistent with the overall guidance of the rubric which encourages raters to assign a 0 to any artifact that does not meet benchmark (score 1) level performance (see https://tinyurl.com/CTRubricHE).

Findings

This section describes how students exhibited CT in their course assignments, supported by examples of student-generated artifacts. We categorize our findings into two types of assignments, illustrating how students expressed CT skills through (a) responses to assignment *prompts,* and (b) *reflections* on assignment solutions. Results indicate that for certain CT skills (i.e., decomposition and algorithms), students demonstrated their understanding in different formats within the same assignment. It is important to note, however, that our analysis examined artifacts only in relation to dimensions of CT and did not assess disciplinary content knowledge (e.g., mathematics knowledge, etc.).

Decomposition

Table 3 depicts the assignment prompts related to the CT skill of decomposition. The "N/A" values in the table indicate that we cannot interpret any prompts for decomposition in this assignment. Figure 3 depicts the distribution of the CT decomposition skill demonstrated across all 101 analyzed student artifacts, organized by assignment. The numbers in each bar segment indicate the number of artifacts with that score.

As shown in Table 3, assignment prompts addressing decomposition are either described explicitly (e.g., "decomposition", "break", and "separately") or implicitly by referencing smaller parts of the problem (e.g., *Sociology—Gender Display Algorithms*). Consequently, Fig. 3 shows that decomposition was evident across student artifacts in nearly all assignments and disciplines. It is important to note that no student artifacts were scored at the benchmark level (score 1): "fail to completely solve the original problem". All student artifacts in our data set either successfully solved the problem (score 2 and above) or showed no evidence of using decomposition (score 0). Artifacts at the milestone level (score 2) generally did not include sufficient descriptions of their decomposed subproblems. No artifacts received a score of 3, as this level assesses the efficiency of decomposition. Efficiency either requires specialized content knowledge beyond the scope of this study (e.g., how to efficiently decompose a piece of music into different parts, how to efficiently decompose an essay structure into smaller parts), or was not relevant to the intent of the assignment (e.g., the assignment description explicitly asked students to decompose a website into images, colors, slogans, etc.). Finally, the percentages of capstone level (score 4) artifacts are shown, with a median percentage of 25%.

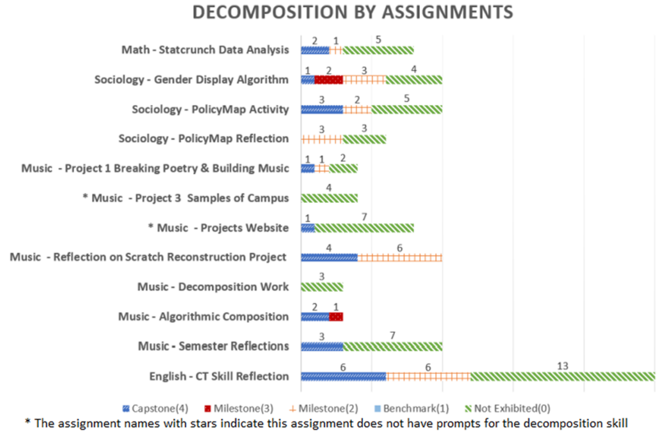
In addition to artifacts scoring between 1 and 4, nearly 50% of the assignments exhibited no evidence of decomposition (score 0). This indicates that although the assignment prompts provided students an opportunity to demonstrate their decomposition skills, not all students exhibited these expected skills. Three reasons may explain this finding. First, students might interpret the assignment prompts as suggestions for consideration rather than requirements. For example, in the *Math—Statcrunch Data Analysis*, students were explicitly asked to write a summary of what they learned from the specific data set they have chosen and about data analysis broadly. However, the CT decomposition prompt simply asked students whether they had *considered* problem decomposition or algorithmic thinking in their work. This prompt did not explicitly require a response, and thus students might have treated it as a scaffold when writing rather than a requirement. Second, the focus of the assignment on data analysis may have directed students towards data alone and not decomposition. Third, the assignment prompt did not distinguish between decomposition and algorithms. For example, the *Sociology—PolicyMap Reflection* asked students to, *"describe how you understand the process of decomposition"*. In response to this prompt, students would easily identify the steps they followed toward decomposition. Identifying, step-by-step actions, however, is the basis of algorithms, not decomposition.

Table 3 Assignment prompts for decomposition

Assignment	Assignment prompts for decomposition
Math—statcrunch data analysis	Write a summary of what you have learned from this assignment. What have you learned from the specific data set which you have chosen? What have you learned about data and analyzing it, in a more general sense? This summary is expected to be 1—2 paragraphs in length Did you consider problem decomposition of algorithmic thinking in working on this assignment? Is there any part of the assignment in which this would be helpful? How did you work together to complete the assignment?
Sociology—gender display algorithms	Note the websites you have selected and start to take a few screen shots of the gendered aspects of the site. These include images of the toys, images of children on the site, colors, games, slogans, and the toys themselves that create an emotional, gendered, commodified experience for the user
Sociology—policymap activity and Sociology—policymap reflection	Write a brief paragraph at the end that describes how you understand the process of decomposition in your work; that is, how you were able to demonstrate that the social conditions you identified in a geographic place can be broken down into smaller parts, revealing their connection to one another
Music—project 1: breaking poetry & building music	Participants will "break" a poetic text and recombine its phonetic materials to create a new piece of music
Music—project 3: samples of campus	N/A
Music—projects website	N/A
Music—reflection on scratch reconstruction project	The design of this project asked you to group your harmonic progressions as blocks that were separate from the larger blocks containing lyrics (and drums, if applicable). What is the purpose of programming the harmony blocks separately first? Do you think it is worth the time to program the harmony as separate blocks?
Music—decomposition work	If you had to take the dataset of form structures that your group created in Project C and use it as source data to create a form for a new song (using a Markov model to actually create the new form), how would you go about this? 1. Decompose the above problem–consider carefully everything you need the computer to do in order to solve this problem 2. Propose an ordered series of steps to solve your newly decomposed problem
Music—algorithmic composition	Turning text into music is a rather large/unusual task. What smaller questions do you need to answer in order to approach this question?

Table 3 (continued)

Assignment	Assignment prompts for decomposition
Music—semester reflections	Choose one of the CT learning outcomes (data, decomposition, abstraction, algorithm) and define it in your own words. Discuss how you have seen this learning outcome work in the various projects we have completed
English writing—CT skill reflection on writing	How did you break this assignment down into smaller parts as you worked on this essay?



	Capstone	Milestones		Benchmark
	4	3	2	1
Decomposition *Breaks a problem into its constituent subproblems*	Breaks a complex problem into clearly described, well-defined, and distinct-but-related subproblems that are easier to solve than the original problem but when combined efficiently solves the original problem.	Breaks a complex problem into clearly described subproblems that are distinct-but-related but lack efficiency, although they solve the original problem.	Breaks a complex problem into subproblems that lack efficiency, fail to have sufficient descriptions, and overlap, although they solve the original problem.	Breaks a complex problem into subproblems that are inefficient, described poorly, overlap or closely related, and fail to completely solve the original problem.

Fig. 3 Score distributions of decomposition in student artifacts and the rubric for decomposition. *Note* The numbers in each bar segment indicate the number of artifacts with that score

Representative capstone student artifacts

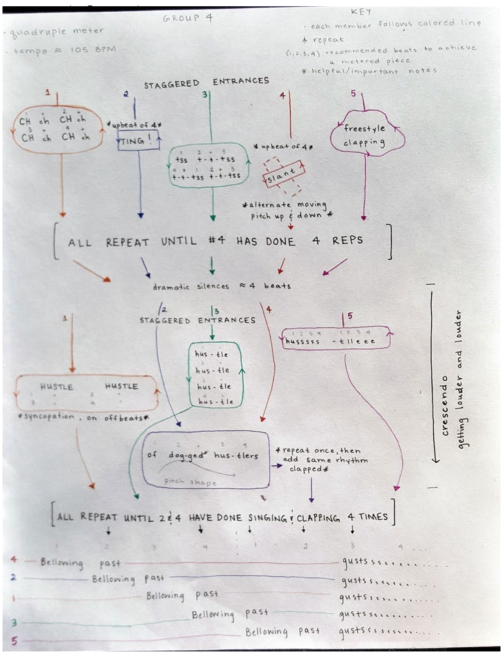
One way in which students exhibited decomposition skills is in response to specific prompts in the assignment. We provide several examples with a score of 4 (capstone) which illustrate different demonstration formats (i.e., text description and drawing notation) depending on disciplinary expectations and instructions in the course assignments.

The example in Table 4 with a score of 4 (capstone) demonstrates a student's response to the Sociology-PolicyMap Exercise assignment, which asked students

Table 4 Student response example for introduction to sociology for introduction to sociology—PolicyMap assignment illustrating CT concepts of decomposition

Assignment prompt	Student response
Introduction to Sociology—PolicyMap Exercise: "Identify 2 topics that you can examine in PolicyMap. List them and how they're measured, and what the options are for presenting that data."	Score 4 (capstone): "Demographics—can be measured by populations, households, families, electrons, religion. For each of those it gets more specific—race, gender, age, diversity, household type/size, who was voted for, denominations, & then those get more specific yet."

Table 5 Student response example for music—project 1: breaking poetry & building music assignment illustrating CT concepts of decomposition, algorithms, and abstraction

Assignment prompt	Student responses
Fundamentals of music I—project 1: "Participants will 'break' a poetic text and recombine its phonetic materials to create a new piece of music…You will take a stanza of a new poem and work in groups of four or five students to refashion it into a single movement of a new eight-movement musical work… The final composition will be submitted in two forms: the performance itself (which will be video recorded), and a notated score that indicates the organization of sounds, gestures, timing, and any other musical elements (tempo, meter, etc.)."	Score 4 (capstone): 

to identify PolicyMap topics sociologically and describe their usage. This example illustrates the CT skills of *decomposition* and *data* simultaneously, as it involves breaking down sociological topics into several indicators and describing how students measured these topics.

The second example of the response to assignment prompts comes from *Fundamentals of Music I—Project 1*, which asked students to break down poetry and build music. As shown in Table 5, this example with a score of 4 (capstone) illustrates the CT concepts of *decomposition, algorithms,* and *abstraction*. The student-created music was decomposed into 5 tracks, each performed by a different student, indicating *decomposition*. The drawing features arrows and repetitions to express steps in finishing the music, which indicates the *algorithms* being used. The music was represented by this drawing with notation, which is clear and clean, indicating the use of *abstraction*. It is worth noting that multiple CT skills could be observed simultaneously in one artifact.

Another way in which students exhibited the use of decomposition skills is through reflection on assignments. Typically, instructors asked students to write a reflection on their assignment, to specifically elicit students' perspectives of their usage of decomposition. The example shown in Table 6 illustrates a student's reflection on the *Mathematics—Statcrunch Data Analysis* assignment, an artifact scored 4 (capstone). In this reflection, the student responded to the value of the instructor's handout (i.e., the handout describes the purpose of the assignment and then decomposes it into 5 sub-problems), which distinguished the difference between *decomposition* and *algorithmic* thinking, describing these two skills separately.

We observed that different students utilized different ways of applying decomposition in response to the same assignment prompt. For example, we observed three ways of using decomposition in the reflection assignment for the introductory writing course offered by the English instructor. These three examples shown below demonstrate how students applied decomposition skills to their writing based on the *structures* of the essay, the writing *topics*, and the writing *process*, respectively (Table 7).

Table 6 Student response example for mathematics—statcrunch data analysis assignment illustrating CT concepts of decomposition

Assignment prompt	Student responses
Mathematics—statcrunch data analysis: 4. Write a summary of what you have learned from this assignment. What have you learned from the specific data set which you have chosen? What have you learned about data and analyzing it, in a more general sense? This summary is expected to be 1 – 2 paragraphs in length 5. Did you consider problem decomposition of algorithmic thinking in working on this assignment? Is there any part of the assignment in which this would be helpful? How did you work together to complete the assignment?"	Score 4 (capstone): "At first, we felt a little overwhelmed at how many parts made up this assignment, so in order to tackle it, we took the problem and broke it into smaller parts, in similar ways that the handout did. We wanted to break the assignment up into data, visual representation and then verbal descriptions to try and make the assignment easier. (Author notes: up to this point students exhibited decomposition; after this section students exhibit understanding of algorithms) To do this, we used algorithmic thinking to break each category we created into a step by step procedure. In order to tackle the data section, we listed out our tasks in order…we moved on to the visual representation section… To tackle the verbal descriptions, we analyzed each graph first…"

Table 7 Three student response examples for english writing assignment illustrating different ways of applying CT concepts of decomposition

Assignment prompt	Student responses
English writing: "How did you break this assignment down into smaller parts as you worked on this essay?"	One reflection which scored 4 (capstone) described the ways that the student broke down the sections based on the *structures* of the finished essay: "I really broke it up into eight parts, my opening paragraph, five sources—each examined one at a time, then conclusion and research proposal." (Score 4) Some students exhibited decomposition skills in relation to the writing *topics* of the essay: "I originally asked questions like: 'what is it that compels an individual to produce a linguistic creation that is so specific to their identity?' and 'do people produce poetry as a means of emotional escape?' … each one of my questions lead to an in-depth speculation on branching topics." (Score 4) Other students described how they broke down the writing *process* instead of the essay structure: "The pieces that I broke this process into essentially consisted of two phases: the source search and the writing itself." (Score 4) "I broke this assignment into sections of research, sources, writing and revision." (Score 4)

Algorithms

Table 8 presents assignment prompts related to algorithms. Like decomposition, the "N/A" values in the table indicate that we did not find any prompts for algorithms in this assignment. Figure 4 shows the distribution of CT algorithm use across all 101 student artifacts in all disciplines and assignments.

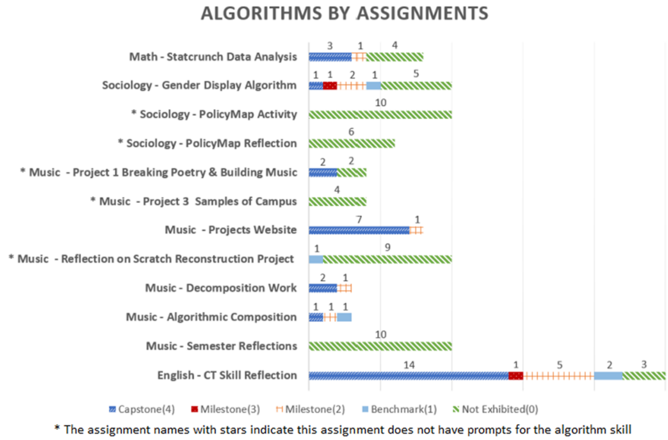
As shown in Fig. 4, a higher percentage of artifacts were scored at the proficient level (Score 4) for algorithms compared to decomposition (median = 44%). The highest number of capstone-level artifacts were evident in the disciplines of *Music* and *English*. This finding may be attributed to the fact that reflection prompts in music and English assignments explicitly made connections to algorithms. Like decomposition, fewer artifacts were scored at the baseline level (Score 2), compared to score 3 or benchmark (Score 1). This finding is consistent with results on decomposition; artifacts with a score of 2 did not provide sufficient description of the algorithmic steps even when the assignment did solve the problem.

A review of artifacts with no evidence of algorithms (score 0) revealed three primary reasons contributing to this outcome: First, some assignments did not give students a prompt to demonstrate algorithmic skills. These assignments are listed in the Table 8 as "N/A". Second, in some assignments, the instructors' prompt was vague and did not explicitly invite connections to algorithms. Third, while the instructions may have been clear, students sometimes had difficulty distinguishing between decomposition and algorithms. For example, the *Sociology—Gender Display*

Table 8 Assignment prompts for algorithmic skills

Assignments	Assignment prompts for algorithms
Math—statcrunch data analysis	Did you consider problem decomposition of algorithmic thinking in working on this assignment?
Sociology—gender display algorithms	Using the above criteria, write an *algorithm* to tell a machine how to identify gender on children's websites if it were to search the web for them. How could you teach a machine to identify gender?
Sociology—policymap activity and sociology—policymap reflection	N/A
Music—project 1: breaking poetry & building music	N/A
Music—project 3: samples of campus	N/A
Music—projects website	Student will submit both of these products to static web pages on their individual websites, which will also contain a reflection on the process, product, and learning that took place working on this project
Music—reflection on scratch reconstruction project	N/A
Music—decomposition work	If you had to take the dataset of form structures that your group created in Project C and use it as source data to create a form for a new song (using a Markov model to actually create the new form), how would you go about this? 1. Decompose the above problem; consider carefully everything you need the computer to do in order to solve this problem 2. Propose an ordered series of steps to solve your newly decomposed problem
Music—algorithmic composition	Lastly, describe in detail a series of ordered steps–an algorithm–that you could hand to your friendly neighborhood programmer and ask them to write into code for you. Consider carefully the sequence of steps and be specific about both the purpose and structure of each step
Music—semester reflections	Choose one of the CT learning outcomes (data, decomposition, abstraction, algorithm) and define it in your own words. Discuss how you have seen this learning outcome work in the various projects we have completed
English writing—CT skill reflection on writing	Once you broke it [assignment] up, how did you proceed? How did you tackle each of those pieces? What advice would you give future students about a writing an essay like this one? Specifically, what detailed writing process would you recommend that future students undertake?

Algorithms explicitly asked students to write an algorithm, but some students only listed bullet points that the algorithm needs to check, without describing how to check these points. Decomposition focuses on *what*—decomposing the problem into subproblems, and algorithms focus on *how*—solving subproblems in steps. Decomposition and algorithms could be naturally linked to one another, causing students without a deeper understanding of CS principles difficulty in distinguishing between them.



	Capstone	Milestones		Benchmark
	4	3	2	1
Algorithms *Creates a series of ordered steps to solve a problem or achieve a goal*	Creates a logical, efficient, and well-described sequence of steps or instructions to solve a problem or achieve a goal.	Creates logical sequence of steps that are well-described (e.g., unambiguous, precise) and solve a problem or achieve a goal but the steps are inefficient e.g., not in an optimal sequence, overlapping, duplicative, or unnecessary.	Creates logical sequence of steps that solve a problem or achieve a goal but the steps are poorly described (e.g., ambiguous, vague).	Creates a sequence of steps that do not solve a problem or achieve a goal. The steps lack efficiency, sufficient descriptions, and are not described or documented.

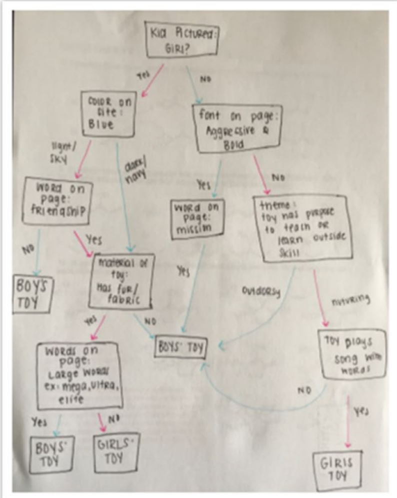
Fig. 4 Score distributions of algorithms in student artifacts and the rubric for algorithms. *Note* The numbers in each bar segment indicate the number of artifacts with that score

Representative capstone student artifacts

A great example at the proficient level (Score 4) is the *Gender Display Algorithms* assignment from the sociology course, which as its name suggests, has explicit prompts related to algorithms. Students designed various forms of algorithms to complete this assignment as shown below. Table 9 shows two examples illustrating the CT skill of algorithms with a score of 4 (capstone). The top example outlines three steps for analyzing the website: toy image analysis, image description analysis, and tally count with a tiebreaker. The bottom example draws a graphical flowchart to analyze the website, where the flowchart has a start point and ends with either boys or girls (unfortunately, the raw data is originally blurred). These examples illustrate two different demonstration formats for presenting algorithms (i.e., verbal description and graphical flowchart).

Like decomposition, student reflections on assignment solutions also illustrated their understanding of algorithms. In these reflections, students usually described how they solved the assignment with steps in time order, using language that signaled understanding of algorithms (i.e., *first… then…*). Table 10 presents examples with score 4 (capstone) from the *Contemporary Mathematics—Statcrunch Data Analysis* assignment and *English writing*, respectively.

Table 9 Two student response examples to sociology—gender display algorithms assignment illustrating CT concepts of algorithms

<table>
<tr><th>Assignment prompt</th><th>Student responses</th></tr>
<tr><td>Sociology—gender display algorithms:
“Browse the Internet for sites that sell children’s toys and find three that are highly gendered towards ‘boys’ or ‘girls.’ … Write an algorithm to tell a machine how to identify gender on children’s websites if it were to search the web for them.”</td><td>Score 4 (capstone):
-Gather (Toy Image) and (Image Description) for Analysis;
-Begin (Toy Image) Analysis;
-Identify ‘Action Image’: (Hitting, Building, Shooting, etc) and Tally;
-Identify ‘Passive Image’: (HandHolding, Waiting, Talking, etc.) and Tally;
-If Tally ‘Action Image’ exceeds Tally ‘Passive Image’, Tally to ‘Masc’;
-If Tally ‘Passive Image’ exceeds Tally ‘Action Image’, Tally to ‘Fem’;
-Begin (Image Description) Analysis;
-Identify ‘Action Words’: (Tool, Battle, Arrow, Exploration, etc.) and Tally;
-Identify ‘Passive Words’: (Princess, Beauty, Friendship, etc.) and Tally;
-If Tally ‘Action Words’ exceeds Tally ‘Passive Words’, Tally to ‘Masc’;
-If Tally ‘Passive Words exceeds Tally ‘Active Words’, Tally to ‘Fem’;
-If Tally ‘Masc’ exceeds Tally ‘Fem’, Toy is marketed to MALES.
-If Tally ‘Fem’ exceeds Tally ‘Masc’, Toy is marketed to FEMALES.
-If neither Tally exceeds the Other, tie is broken with Color Spectrum Analysis;
-Color Spectrum Analysis: Closer to Pink, TIEBREAK FEMALE.
-Color Spectrum Analysis: Closer to Blue, TIEBREAK MALE.

</td></tr>
</table>

Unfortunately, the raw image of the bottom example is blurred

Data

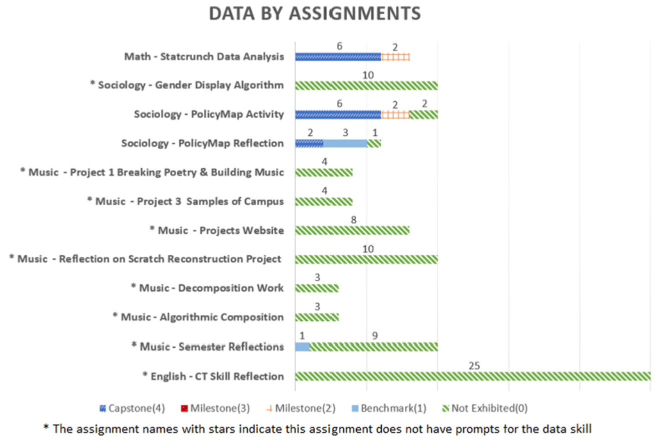
Table 11 presents the assignment prompts related to the use of data, while Fig. 5 shows evidence of data skills in student artifacts. Data skills were not frequently prompted by instructors, and thus were not commonly demonstrated by students. They were only evident in assignments that directly asked for the use of data-related skills. Among three assignments exhibiting data skills, students generally performed better on assignments that included direct data prompts compared to those assignments prompting reflection on solutions. This outcome may be attributed to the nature of the rubric which focuses primarily on showing evidence of evaluating a data set, describing

Table 10 Two student response examples to mathematics—statcrunch data analysis and english writing assignments illustrating CT concepts of algorithms

Assignment prompt	Student responses
Mathematics—statcrunch data analysis: "Did you consider problem decomposition of algorithmic thinking in working on this assignment?"	Score 4 (capstone): "First, we made sure we founded the dataset that we were interested in and looked it over. Next, we created the statistical summary. Lastly, we analyzed the findings. After this, we moved on to the visual representation…Next, we plugged each question we had…Next we outlined what each graph said, and last, we described each graph…"
English Writing: "Once you broke it [assignment] up, how did you proceed? How did you tackle each of those pieces?"	Score 4 (capstone): "First, I had to find a topic that I cared about and was interested in, and finally settled on childhood literacy. Once I had my topic figured out, I searched for different sources and compiled them into a list. From these sources, I made a thesis about childhood literacy. Then, as soon as I knew my thesis, I sat down in the library for three hours and typed. In that time, I finished the entire first draft of my paper in one sitting. After this draft was turned in and peer edited, I made slight changes here and there."

Table 11 Assignment prompts for data

Assignments	Assignment prompts for data
Math—statcrunch data analysis	Research often involves formulating interesting questions, developing a methodology, and collecting data to answer those questions. Although we will be using data that has already been collected, we will use this assignment to explore how data can be used to answer questions that we have
Sociology—policymap activity	As a pair, your chief task is to identify a sociological issue of interest to you and explore how it coincides with at least one other important social, political, or economic indicator and data points. You will do this by bringing together related phenomena, informed by sociological reasoning, to show how these constituent social issues interact with one another in a geographic place
Sociology—policymap reflection	Write a brief paragraph at the end that describes how you understand the process of decomposition in your work; that is, how you were able to demonstrate that the social conditions you identified in a geographic place can be broken down into smaller parts, [revealing their connection to one another, visualized through mapping]



	Capstone	Milestones		Benchmark
	4	3	2	1
Data *Evaluate a data set to ensure that it facilitates discovery of patterns and relationships*	Evaluates a data set to ensure it is sufficiently comprehensive, efficiently organized, meaningfully labeled, and thoroughly described so that it can be analyzed to discover meaningful patterns and relationships.	Evaluates a data set to ensure it is sufficiently comprehensive, meaningfully labeled, and thoroughly described but fails to ensure that it is efficiently organized.	Evaluates a data set to ensure it is sufficiently comprehensive and meaningfully labeled but fails to ensure that it is thoroughly described and efficiently organized.	Evaluates a data set but fails to ensure that it is sufficiently comprehensive, efficiently organized, meaningfully labelled, and thoroughly described so patterns and relationships are obscured.

Fig. 5 Score distributions of data in student artifacts and the rubric for data. *Note* The numbers in each bar segment indicate the number of artifacts with that score

meaningful patterns, and discovering relationships. Reflections, in contrast, are generally designed to help students articulate their thoughts and processes, thus providing fewer opportunities for students to exhibit evidence of evaluating data.

Representative capstone student artifacts

Overall, students demonstrated their understanding of data quantitatively and qualitatively through analysis with tools and description of the results. Data, however, is not a standalone CT skill; it is also influenced by students' disciplinary knowledge and ability to interpret the data. We provide two examples related to data from mathematics and sociology. Table 12 presents the first example, a student artifact scoring 4 (capstone) for quantitative data analysis. The student used disciplinary knowledge and tools (i.e., Statcrunch) to statistically summarize a dataset, generate graphs, and verbally describe the results. The second example shown below from the *Sociology—PolicyMap Usage* assignment illustrates how students exhibited their understanding of data by evaluating public sociological data to qualitatively explain sociological issues. The example shown below reflects data skills from a CT perspective only; we did not assess its accuracy from a content knowledge perspective.

There was only one reflection assignment in our dataset where students had an opportunity to exhibit their understanding of data skills in sociology. An artifact

Table 12 Two student response examples to mathematics—statcrunch data analysis and sociology—PolicyMap assignments illustrating CT concepts of data

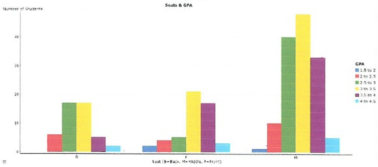
<table>
<tr><th>Assignment prompt</th><th>Student responses</th></tr>
<tr><td>Mathematics—statcrunch data analysis assignment:
“Statcrunch will be used to select data and generate both numeric summaries and graphs representing the data… Use the Statcrunch software to come up with a statistical summary… Use the software to create four different graphs to display the data…”</td><td>Score 4 (capstone):
MATH DATA ASSIGNMENT
Body Image & GPA Data Set
Link to data set: https://www.statcrunch.com/app/index.php?dataid=2612822
Questions
• Does the area you sit in class affect GPA?
• Does body weight affect student confidence and GPA?
• Could student honesty about others cheating affect GPA?
• Could high school GPA affect college GPA?
Statistical Summary
<table>
<tr><th>Column</th><th>n</th><th>Mean</th><th>Variance</th><th>Std. dev.</th><th>Std. err.</th><th>Median</th><th>Range</th><th>Min</th><th>Max</th><th>Q1</th><th>Q3</th></tr>
<tr><td>GPA</td><td>236</td><td>3.1080763</td><td>0.28167695</td><td>0.53073246</td><td>0.034547741</td><td>3</td><td>2.47</td><td>1.91</td><td>4.38</td><td>2.8</td><td>3.5</td></tr>
</table>
Out of a sample size of 236 students, the average (mean) GPA is 3.1080763. The range of GPA is 2.47, and the standard deviation is 0.53073246. The maximum GPA reached is 4.38, and the minimum GPA is 1.91.
Graphs

</td></tr>
<tr><td>Sociology—policymap usage:
“Produce a single layer map of your selected indicator for your specified area… What are some of the interesting geo-spatial patterns that you see?”</td><td>Score 4 (capstone):
“When looking at language, specifically, non-English speakers, I see it is prevalent in the southwest in the same way the hispanic population is. This can be explained by the idea that these states border Mexico, so there are more hispanics in that region and in turn more people who don’t speak English.”</td></tr>
</table>

Table 13 Student response example to sociology—PolicyMap assignments illustrating CT concepts of data

<table>
<tr><th>Assignment prompt</th><th>Student responses</th></tr>
<tr><td>Sociology—policymap reflection:
“Write a brief paragraph at the end that describes how you understand the process of decomposition in your work; that is, how you were able to demonstrate that the social conditions you identified in a geographic place can be broken down into smaller parts, revealing their connection to one another, visualized through mapping.”</td><td>Score 4 (capstone):
“By creating a two-layer map of high African American concentration and high IMR (Infant Mortality Rate), we were able to see that Baltimore City fit these criteria, while the surrounding areas did not. Therefore, to see how race was playing a role in IMR, we created another two-layer map of the surrounding Baltimore County. By creating the second map, we were able to see that in areas with a high white concentration, even in close proximity to a high black concentration, have a much lower IMR. Thus, through the construction, analysis, and comparison of these two maps, we were able to identify how race affects IMR in ways that is not consistent between whites and blacks and that race is truly an essential determinant of IMR outcomes.”</td></tr>
</table>

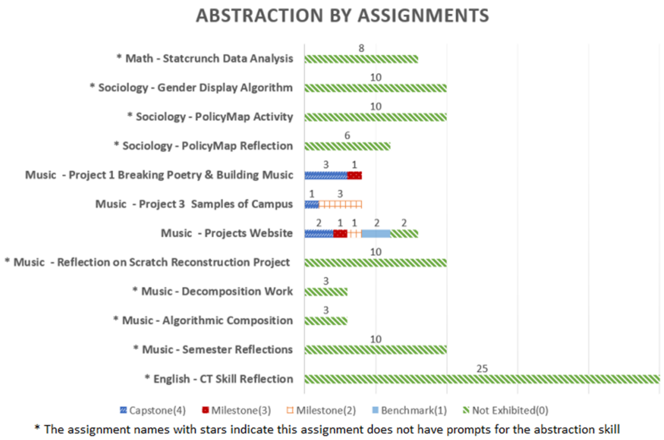
demonstrating capstone (4) level skill in Table 13 demonstrates how the student analyzed the data and provided general descriptions of the results. The student response illustrates how they created a two-layer map to identify meaningful patterns and then produced a second map to discover meaningful relationships, exemplifying a capstone-level use of data.

Abstraction

Table 14 shows the assignment prompts that involve abstraction skills; all assignments are in the music course. Figure 6 shows the score distributions for abstraction.

Table 14 Assignment prompts for abstraction

Assignments	Assignment prompts for abstraction
Music—project 1: breaking poetry & building music, Music—project 3: samples of campus, and Music—projects website	The final composition will be submitted in two forms: the performance itself (which will be video recorded), and a notated score that indicates the organization of sounds, gestures, timing, and any other musical elements (tempo, meter, etc.)



	Capstone	Milestones		Benchmark
	4	3	2	1
Abstraction *Reduces complexity to create a general representation of a process or group of objects so it is not only appropriate for the immediate purpose or goal but can also be used in different contexts*	Creates an accurate-but-simplified representation of a process or group of objects to solve the problem or meet the goal. Selects essential characteristics by filtering out unnecessary information. Can be used to solve other problems or goals.	Creates an accurate-but-simplified representation of a process or group of objects to solve the problem or meet the goal. Selects essential characteristics by filtering out unnecessary information. Cannot be used to solve other problems or goals.	Creates an accurate-but-simplified representation of a process or group of objects to solve the problem or meet the goal. Fails to select all essential characteristics by filtering out unnecessary information. Cannot be used to solve other problems or goals.	Creates a representation of a process or group of objects that is not accurate, not sufficiently simplified, or fails to solve the problem or meet the goal.

Fig. 6 Score distributions of abstraction in student artifacts and the rubric for abstraction. *Note* The numbers in each bar segment indicate the number of artifacts with that score

According to the rubric, abstraction is defined as "accurate-but-simplified representation". We identified two reasons contributing to a lower level of abstraction demonstrated in course assignments. First, there was missing information which made it difficult to assess abstraction in student artifacts. For example, students used symbols and drawing notations to represent their created music, but they did not provide any instructions on how to read the symbol and the whole notation. Second, student responses relied heavily on text. In one instance, while the student provided an abstract picture to represent the element of their music, they also wrote 6 paragraphs describing the picture, causing the picture itself to fail to represent their music.

Representative capstone student artifacts

Figure 7 presents an example from *Fundamentals of Music I—Project 1: Breaking Poetry & Building Music*. In this example, students modeled the music piece as a flow chart, which abstracts the musical components including sounds and beats as symbols in the flow chart. Students provided the meaning of different symbols, making this notation simplified but accurate, which contributes to the capstone level (score 4). Another example of how students used abstraction skills in music artifacts was previously discussed in the decomposition section (see Table 5 above).

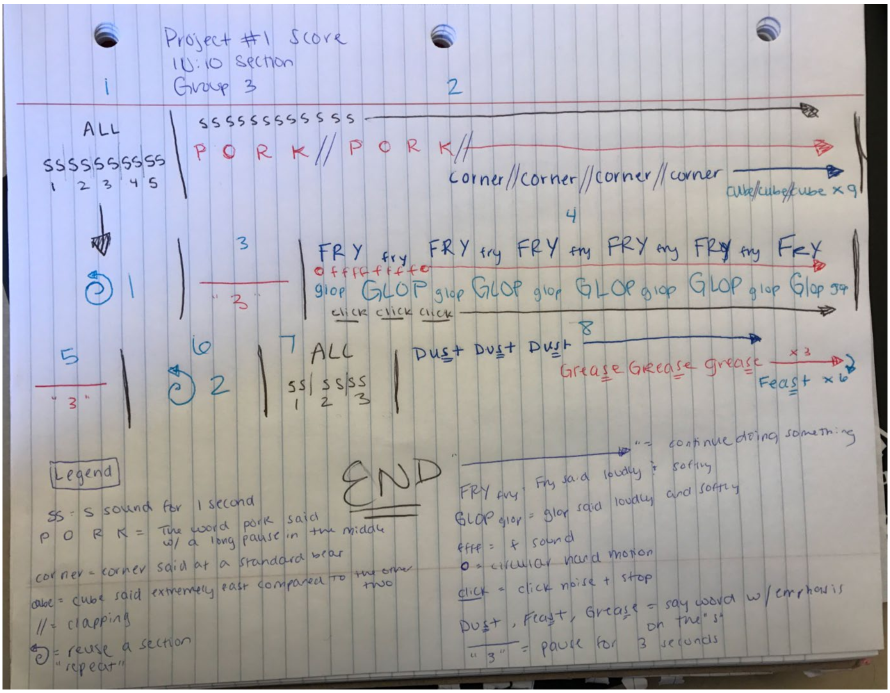


Fig. 7 Student response example to music assignments illustrating CT concepts of abstraction

Discussion and implications

In this section, we discuss several key aspects of the findings and their implications for research and practice. In particular, findings from our work contribute significantly to the integration of CT into higher education by identifying CT skills more frequently exhibited in student non-programming artifacts, illustrating strengths and weaknesses associated with different types of CT-integrated assignments, and pointing toward assessments with the potential to capture student learning.

CT skills exhibited in student artifacts

Findings from this work indicated that the nature of CT skills exhibited in course assignments differed by discipline. Although CT skills are applicable across many fields, some disciplines may more readily lend themselves to using these skills (Czerkawski & Lyman, 2015; Jong & Jeuring, 2020). Specifically, findings revealed that decomposition and algorithmic skills appeared more frequently than other CT skills across both instructor-developed assignments and student artifacts. Although literature comparing the prevalence of specific CT skills across disciplines is limited, some studies suggest that instructors develop course objectives and materials to reflect a limited number of CT skills, that are not evenly distributed across disciplines (Dierbach et al., 2011; Mouza et al., 2017). One potential explanation is that decomposition typically represents the initial step in computational problem-solving, often preceding CT skills such as pattern recognition and abstraction (Shute et al., 2017). Another reason may be that faculty and students, both within and outside computing fields, prioritize certain CT skills such as decomposition and algorithms over others (Ater-Kranov et al., 2010). In response to various assignment prompts and reflections, students also utilized different demonstration formats, including text descriptions, drawing notation, and narrative reflections. Often, students exhibited decomposition and algorithmic skills in combination with other CT skills.

Consistent with prior work (Chen et al., 2017), data skills were found to be closely linked to disciplinary knowledge and expressed in various forms, such as statistical analysis, tables and figures, and descriptive explanations, all tailored to discipline-specific features and instructor prompts. Additionally, Chen et al. (2017) emphasized that while data processing is crucial, isolating it can be challenging, leading to its integration with other skills in their assessment rubric. Furthermore, in developing CT-integrated assignments related to data, it became evident that instructors need existing datasets and computing tools (e.g., in our work, Statcrunch and PolicyMap). This finding is consistent with prior work by Castro et al. (2021) who utilized MATLAB, a statistical tool, to represent and interpret data in multiple formats in order to investigate the relationship between data practices and problem-solving in an engineering course. Relatedly, findings indicated that direct prompts provided more opportunities for students to exhibit CT skills around data compared to course reflections.

Finally, the CT-skill of abstraction was not commonly prompted and exhibited in our dataset, appearing primarily in music, and integrated with decomposition and algorithmic thinking. For example, decomposition was shown through the use of separate music tracks corresponding to different colors while algorithmic thinking was demonstrated as the time-sequencing of musical elements. Abstraction itself was represented in the translation of music into sheet music or flowcharts. Our findings suggest that decomposition and algorithmic thinking contributed to the development of abstraction skills to some degree. This finding aligns with theoretical work suggesting that abstraction frequently follows decomposition in the CT problem-solving process (Shute et al., 2017). Further, this finding is aligned with the foundational premise of the assessment rubric guiding this work where decomposition and abstraction are conceptually related, with decomposition a necessary skill for abstraction (Guidry et al., 2019).

The limited use of abstraction in student artifacts is consistent with prior studies focusing on the integration of CT in teacher education courses. For instance, in a study conducted by Mouza et al. (2017) where pre-service teachers designed lessons plans that integrated CT across different disciplines, findings indicated minimum use of abstraction and uneven distribution of CT skills between disciplines (e.g., mathematics, science, literacy). Similarly, Dierbach et al. (2011) presented learning objectives for five CT-infused courses, and these objectives relied more on decomposition and algorithms and less on abstraction. Finally, Swaid (2015) illustrated CT skills for 12 courses across STEM disciplines; of those, 12 courses involved data, 9 courses involved algorithms, and only 7 courses involved abstraction. Although standards connecting disciplines and CT skills are still lacking, more empirical studies are needed exploring synergies between CT and disciplinary content in ways that enhance learning across both domains.

Design of CT-integrated assignments

In this work, we primarily focused on how students successfully exhibited CT skills in their course assignments, as reflected in our rubric. However, we also observed instances where these skills were only partially exhibited. We identified three primary reasons that contributed to this challenge. First, some assignments lacked clarity and did not explicitly prompt students to articulate their CT understanding. Even when they did, some instructors did not specify the CT skill expected from students (e.g., decomposition vs algorithms). Second, we also noted that students struggled to distinguish between different CT concepts even when the prompt was clear, notably decomposition and algorithms. Lastly, some assignments did not provide sufficient detail to assess students' level of proficiency. In some cases, students provided minimal explanation with no associated examples to illustrate their understanding of the CT skills at hand.

This challenge suggests that when prompting students to demonstrate CT skills, instructors might need to provide examples of what a sufficient description looks like. Further, different strategies may be needed for different CT skills. Previous research, for instance, demonstrated promising results when using learning scenarios (Perković, 2010; Rojas-López & García-Peñalvo, 2018), game design (Lockwood & Mooney, 2018), and environment modeling for decomposition skills (Perković, 2010). As a result, course instructors need to review different strategies before deciding which ones hold promise for their specific discipline. Findings from this work also indicated that students struggled to distinguish among different CT concepts, especially decomposition and algorithms. Thus, instructors need to scaffold students' understanding of decomposition and algorithmic thinking. Further, they should ensure assignment prompts are adequately described and explicitly point to the desired outcome.

One key challenge in developing assignments with clear instructions and scaffolding prompts is faculty expertise. University faculty outside of CS bring expertise in their domain knowledge, but typically do not have sufficient knowledge of CT. Even though we provided multiple opportunities to scaffold the development of CT-integrated assignments and support participants' learning of CT, including professional development, bi-weekly meetings, and situated classroom support offered by CT Fellows (Pollock et al., 2019), our findings suggest that more pedagogical support may be needed. Toward this end, the framework of Technological Pedagogical Content Knowledge (TPACK) may be useful. Specifically, Angeli et al. (2016) provided a conceptualization of TPCK for the construct of CT where content knowledge (CK) includes abstraction, generalization, decomposition, algorithmic thinking, and debugging. Future work may utilize the TPACK framework for faculty development, with a more explicit focus on building both CK and pedagogical knowledge in relation to CT integration.

Assessing CT in student artifacts

Findings indicated some challenges in assessing student artifacts associated with our rubric. For instance, some artifacts did not demonstrate any CT skill and were scored at 0. Although the rubric encourages raters to assign a 0 to any artifact that does not meet benchmark (score 1) level performance, future work could consider distinguishing these two levels and adding a level that indicates the lack of evidence for the relevant CT skill. Similarly, "efficiency" is the key difference between scores 3 and 4 in the rubric, but it is difficult to assess it in practical terms because it either requires the rater to have content knowledge expertise to define "efficiency" or it requires the instructor to specifically design assignments that involve "efficiency". As the rubric notes, "efficiency is a particularly challenging skill even for upper class computer science students who explicitly focus on it" (see https://tinyurl.com/CTRubricHE). Although the rubric simplified the concept of efficiency, it is unrealistic to expect that it would be addressed at the advanced level in the majority of CT assignments and student artifacts outside of CS. Nonetheless, it is important to include it in the rubric to demonstrate what efficiency means, allowing instructors to broaden their understanding and begin considering new kinds of assignments as students gain more experience with CT.

Further, findings indicated that some assignments cannot be adequately scored by the rubric. For example, if an assignment asks students to order a series of unordered steps using algorithms, a student response will only result in either correct or incorrect order. Thus, it would be difficult to assess the extent of a student's understanding based on our rubric. This indicates the need for instructors to carefully consider the design of CT-integrated assignments to ensure they demonstrate a range of CT skills while revealing students' thought processes. Further, in addition to direct assignment prompts and indirect reflections analyzed in this study, other assessment strategies could also provide information on CT development, such as structured-response pre-post tests (e.g., Román-González et al., 2017) and interviews to interactively probe students' understanding of CT (Cetin & Andrews-Larson, 2016). Finally, more research is needed that applies our proposed rubric in practice with the explicit purpose of establishing validity and reliability.

Limitations

We have identified several limitations associated with this work. First, all of our data were collected and provided by the instructors, not by the authors. Some instructors provided comprehensive information associated with their course assignments, including the full number of students completing the assignment and the accompanied artifacts, while others only provided descriptions with no accompanied student artifacts. Second, student artifacts are direct responses to the instructors' assignment type and description. We noted instances where the instructors' description did not explicitly ask students to demonstrate their CT skills, thus making it difficult to evaluate evidence of CT skill development. Third, we mainly assessed CT skill development for each artifact and not disciplinary content knowledge. There might be instances in which students' content knowledge is incorrect, despite demonstrating a sufficient understanding of CT. Thus, future work may want to adapt the rubric to include both disciplinary learning and CT skill development. Finally, we collected data only within one institution with 5 courses. The data may not represent other courses in those disciplines.

Conclusion

In this study, we examined how and to what extent students from five non-CS courses used CT skills when completing assignments in those courses. Our data included 101 student artifacts from 12 assignments across five courses in four disciplines: mathematics, sociology, music, and English. Our results indicated that the use of CT was dependent on disciplinary content and assignment prompts, with artifacts reflecting CT skills through various perspectives and formats. Our results also revealed that students outside of CS can demonstrate a high level of CT proficiency (i.e., capstone) across various disciplines.

The rubric used to assess students' CT skills is developed for general assessment across disciplines; however, some CT skills may align better with specific disciplines. Future studies should consider adapting this rubric for specific disciplinary contexts. Further, they should also consider ways in which CT-integrated assignments presented in this work could be iterated to more explicitly support the development of students' CT skills.

Appendix A

See Table 15.

Table 15 The rubric

	Capstone	Milestones		Benchmark
	4	3	2	1
Decomposition *Breaks a problem into its constituent subproblems*	Breaks a complex problem into clearly described, well-defined, and distinct-but-related subproblems that are easier to solve than the original problem but when combined efficiently solves the original problem	Breaks a complex problem into clearly described subproblems that are distinct-but-related but lack efficiency, although they solve the original problem	Breaks a complex problem into subproblems that lack efficiency, fail to have sufficient descriptions, and overlap, although they solve the original problem	Breaks a complex problem into subproblems that are inefficient, described poorly, overlap or closely related, and fail to completely solve the original problem
Algorithms *Creates a series of ordered steps to solve a problem or achieve a goal*	Creates a logical, efficient, and well-described sequence of steps or instructions to solve a problem or achieve a goal	Creates logical sequence of steps that are well-described (e.g., unambiguous, precise) and solve a problem or achieve a goal but the steps are inefficient e.g., not in an optimal sequence, overlapping, duplicative, or unnecessary	Creates logical sequence of steps that solve a problem or achieve a goal but the steps are poorly described (e.g., ambiguous, vague)	Creates a sequence of steps that do not solve a problem or achieve a goal. The steps lack efficiency, sufficient descriptions, and are not described or documented
Data *Evaluate a data set to ensure that it facilitates discovery of patterns and relationships*	Evaluates a data set to ensure it is sufficiently comprehensive, efficiently organized, meaningfully labeled, and thoroughly described so that it can be analyzed to discover meaningful patterns and relationships	Evaluates a data set to ensure it is sufficiently comprehensive, meaningfully labeled, and thoroughly described but fails to ensure that it is efficiently organized	Evaluates a data set to ensure it is sufficiently comprehensive and meaningfully labeled but fails to ensure that it is thoroughly described and efficiently organized	Evaluates a data set but fails to ensure that it is sufficiently comprehensive, efficiently organized, meaningfully labelled, and thoroughly described so patterns and relationships are obscured

Table 15 (continued)

	Capstone	Milestones		Benchmark
	4	3	2	1
Abstraction *Reduces complexity to create a general representation of a process or group of objects so it is not only appropriate for the immediate purpose or goal but can also be used in different contexts*	Creates an accurate-but-simplified representation of a process or group of objects to solve the problem or meet the goal. Selects essential characteristics by filtering out unnecessary information. Can be used to solve other problems or goals	Creates an accurate-but-simplified representation of a process or group of objects to solve the problem or meet the goal. Selects essential characteristics by filtering out unnecessary information. Cannot be used to solve other problems or goals	Creates an accurate-but-simplified representation of a process or group of objects to solve the problem or meet the goal. Fails to select all essential characteristics by filtering out unnecessary information. Cannot be used to solve other problems or goals	Creates a representation of a process or group of objects that is not accurate, not sufficiently simplified, or fails to solve the problem or meet the goal

Acknowledgements This paper is based upon work supported by the National Science Foundation under grant 1611959.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

Ethical approval All materials that we analyzed are de-identified and all procedures were approved by our institutional review board (IRB).

References

AAC&U. (2018). Value Rubrics. https://www.aacu.org/value/rubrics

Angeli, C., & Giannakos, M. (2020). Computational thinking education: Issues and challenges. *Computers in Human Behavior, 105*, 106185. https://doi.org/10.1016/j.chb.2019.106185

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Journal of Educational Technology & Society, 19*(3), 47–57.

Apiola, M., Saqr, M., López-Pernas, S., & Tedre, M. (2022). Computing education research compiled: Keyword trends, building blocks, creators, and dissemination. *IEEE Access, 10*, 27041–27068. https://doi.org/10.1109/ACCESS.2022.3157609

Ater-Kranov, A., Bryant, R., Orr, G., Wallace, S., & Zhang, M. (2010). Developing a community definition and teaching modules for computational thinking: accomplishments and challenges. In *Proceedings of the 2010 ACM conference on information technology education* (143–148). https://doi.org/10.1145/1867651.186768

Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology, 38*(6), 20–23.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American educational research association*, Vancouver, Canada, 1, 25.

Caballero, M. D., Kohlmyer, M. A., & Schatz, M. F. (2012). Fostering computational thinking in introductory mechanics. In *AIP Conference Proceedings.* 1413, 15–18. American institute of physics. https://doi.org/10.1063/1.3679982

Castro, L. M. C., Magana, A. J., Douglas, K. A., & Boutin, M. (2021). Analyzing students' computational thinking practices in a first-year engineering course. *IEEE Access, 9*, 33041–33050. https://doi.org/10.1109/ACCESS.2021.3061277

Cetin, I., & Andrews-Larson, C. (2016). Learning sorting algorithms through visualization construction. *Computer Science Education, 26*(1), 27–43. https://doi.org/10.1080/08993408.2016.1160664

Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & Education, 109*, 162–175. https://doi.org/10.1016/j.compedu.2017.03.001

Czerkawski, B. C., & Lyman, E. W. (2015). Exploring issues about computational thinking in higher education. *TechTrends, 59*, 57–65. https://doi.org/10.1007/s11528-015-0840-3

Denner, J., Werner, L., Campe, S., & Ortiz, E. (2014). Pair programming: Under what conditions is it advantageous for middle school students? *Journal of Research on Technology in Education, 46*(3), 277–296. https://doi.org/10.1080/15391523.2014.888272

Denning, P. J. (2007). Computing is a natural science. *Communications of the ACM, 50*(7), 13–18. https://doi.org/10.1145/1272516.1272529

Dierbach, C., Hochheiser, H., Collins, S., Jerome, G., Ariza, C., Kelleher, T., Kleinsasser, W., Dehlinger, J., & Kaza, S. (2011). A model for piloting pathways for computational thinking in a general education curriculum. In *Proceedings of the 42nd ACM technical symposium on Computer science education*, (257–262). https://doi.org/10.1145/1953163.1953243

DiSessa, A. A. (2018). Computational literacy and "the big picture" concerning computers in mathematics education. *Mathematical Thinking and Learning, 20*(1), 3–31. https://doi.org/10.1080/10986065.2018.1403544

Fennell, H. W., Lyon, J. A., Madamanchi, A., & Magana, A. J. (2020). Toward computational apprenticeship: Bringing a constructivist agenda to computational pedagogy. *Journal of Engineering Education, 109*(2), 170–176. https://doi.org/10.1002/jee.20316

Greher, G.R., & Heines, J.M. (2014). Computational thinking in sound: Teaching the art and science of music and technology. Oxford.

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher, 42*(1), 38–43. https://doi.org/10.3102/0013189X1246305

Guidry, K., Mouza, C., Pollock, L., & Pusecker, K. (2019). Infusing computational thinking into general education using a VALUE-style rubric. *American Association of Colleges & Universities*, February 14–16, San Francisco, CA.

Guzdial, M. (2008). Education paving the way for computational thinking. *Communications of the ACM, 51*(8), 25–27. https://doi.org/10.1145/1378704.1378713

Guzdial, M. (2015). Learner-centered design of computing education: Research on computing for everyone. *Morgan & Claypool Publishers*. https://doi.org/10.1007/978-3-031-02216-6

Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education, 126*, 296–310. https://doi.org/10.1016/j.compedu.2018.07.004

Hsu, Y. C., Irie, N. R., & Ching, Y. H. (2019). Computational thinking educational policy initiatives (CTEPI) across the globe. *TechTrends, 63*, 260–270. https://doi.org/10.1007/s11528-019-00384-4

Ilic, U., Haseski, H. İ, & Tugtekin, U. (2018). Publication trends over 10 years of computational thinking research. *Contemporary Educational Technology, 9*(2), 131–153. https://doi.org/10.30935/cet.414798

Jaipal-Jamani, K., & Angeli, C. (2017). Effect of robotics on elementary preservice teachers' self-efficacy, science learning, and computational thinking. *Journal of Science Education and Technology, 26*, 175–192. https://doi.org/10.1007/s10956-016-9663-z

Jong, I. d., & Jeuring, J. (2020). Computational thinking interventions in higher education: a scoping literature review of interventions used to teach computational thinking. In *Proceedings of the 20th koli calling international conference on computing education research*, (1–10). https://doi.org/10.1145/3428029.3428055

Kalelioglu, F., Gulbahar, Y., & Kukul, V. (2016). A framework for computational thinking based on a systematic research review. *Modern Computing, 4*(3), 583–596.

Kuster, C., Symms, J., May, C., & Hu, C. (2011). Developing computational thinking skills across the undergraduate curriculum. In *44th Annual Midwest Instruction and Computing Symposium (MICS'11)*, Duluth, MN.

Li, Y., Schoenfeld, A. H., diSessa, A. A., Graesser, A. C., Benson, L. C., English, L. D., & Duschl, R. A. (2020). Computational thinking is more about thinking than computing. *Journal for STEM Education Research, 3*, 1–18. https://doi.org/10.1007/s41979-020-00030-2

Lockwood, J., & Mooney, A. (2018). Computational Thinking in Education: Where does it fit? A systematic literary review. *International Journal of Computer Science Education in Schools, 2*(1), 41–60. https://doi.org/10.21585/ijcses.v2i1.26

Lu, J. J., & Fletcher, G. H. (2009). Thinking about computational thinking. In *Proceedings of the 40th ACM technical symposium on Computer science education*, (260–264). https://doi.org/10.1145/1508865.1508959

Lyon, J. A., & Magana, A. J. (2020). Computational thinking in higher education: A review of the literature. *Computer Applications in Engineering Education, 28*(5), 1174–1189. https://doi.org/10.1002/cae.22295

Mohaghegh, D. M., & McCauley, M. (2016). Computational thinking: The skill set of the 21st century. *International Journal of Computer Science and Informaton Technologies, 7*(3), 1524–1530.

Mouza, C., Marzocchi, A., Pan, Y., & Pollock, L. (2016). Development, implementation and outcomes of an equitable computer science after-school program: Findings from middle school students. *Journal of Research on Technology in Education, 48*(2), 84–104.

Mouza, C., Sheridan, S., Lavigne, N., & Pollock, L. (2023). Expanding an equitable pedagogy framework for teaching computer science: Reflections from the field. *Computer Science Education, 33*(1), 3–28. https://doi.org/10.1080/08993408.2021.1970435

Mouza, C., Yang, H., Pan, Y. C., Ozden, S. Y., & Pollock, L. (2017). Resetting educational technology coursework for pre-service teachers: A computational thinking approach to the development of technological pedagogical content knowledge (TPACK). *Australasian Journal of Educational Technology*. https://doi.org/10.14742/ajet.3521

Oliveira, C. M., Pereira, R., Gasparini, I., & Boscarioli, C. (2022). *An Overview of Computational Thinking in Higher Education: a technical report of a systematic mapping study.* Available at: https://www.researchgate.net/publication/364656993_An_Overview_of_Computational_Thinking_in_Higher_Education_a_technical_report_of_a_systematic_mapping_study

Perković, L. a. (2010). A framework for computational thinking across the curriculum. In *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education*, (123–127). https://doi.org/10.1145/1822090.1822126

Pollock, L., Mouza, C., Atlas, J., & Harvey, T. (2015). Field experience in teaching computer science: Course organization and reflections. In *SIGCSE'15: Proceedings of 46th ACM SIGCSE Technical Symposium on Computer Science Education*, February 2015, pp. 374–379.

Pollock, L., Mouza, C., Guidry, K., & Pusecker, K. (2019). Infusing computational thinking across disciplines: Reflections and lessons learned. In *SIGCSE'19: Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, February 2019, 435–441.

Rojas-López, A., & García-Peñalvo, F. J. (2018). Learning scenarios for the subject methodology of programming from evaluating the computational thinking of new students. *IEEE Revista Iberoamericana De Tecnologias Del Aprendizaje, 13*(1), 30–36. https://doi.org/10.1109/RITA.2018.2809941

Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the computational thinking test. *Computers in Human Behavior, 72*, 678–691. https://doi.org/10.1016/j.chb.2016.08.047

Romero, M., Lepage, A., & Lille, B. (2017). Computational thinking development through creative programming in higher education. *International Journal of Educational Technology in Higher Education, 14*(1), 1–15. https://doi.org/10.1186/s41239-017-0080-z

Saqr, M., Ng, K., Oyelere, S. S., & Tedre, M. (2021). People, ideas, milestones: A scientometric study of computational thinking. *ACM Transactions on Computing Education (TOCE), 21*(3), 1–17. https://doi.org/10.1145/3445984

Selby, C., & Woollard, J. (2013). Computational thinking: the developing definition. *University of Southampton*. Available at: https://www.researchgate.net/publication/299450690_Computational_thinking_the_developing_definition

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying Computational Thinking. *Educational Research Review, 22*, 142–158. https://doi.org/10.1016/j.edurev.2017.09.003

Swaid, S. I. (2015). Bringing computational thinking to STEM education. *Procedia Manufacturing, 3*, 3657–3662. https://doi.org/10.1016/j.promfg.2015.07.761

Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education, 148*, 103798. https://doi.org/10.1016/j.compedu.2019.103798

Vergara, C. E., Urban-Lurain, M., Dresen, C., Coxen, T., MacFarlane, T., Frazier, K., Briedis, D., Buch, N., Esfahanian, A.-H., Paquette, L., Sticklen, J., LaPrad, J., & Wolff, T. F. (2009). Aligning computing education with engineering workforce computational needs: New curricular directions to improve computational thinking in engineering graduates. In *2009 39th IEEE frontiers in education conference* (1–6). IEEE. https://doi.org/10.1109/FIE.2009.5350463

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*, 127–147. https://doi.org/10.1007/s10956-015-9581-5

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35. https://doi.org/10.1145/1118178.1118215

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education (TOCE), 14*(1), 1–16. https://doi.org/10.1145/2576872

Yadav, A., Stephenson, C., & Hong, H. (2017). Computational thinking for teacher education. *Communications of the ACM, 60*(4), 55–62. https://doi.org/10.1145/2994591

Yifan Zhang is a postdoctoral researcher at the Faculty of Education, Beijing Normal University. He received his Ph.D. in Computer Science from the University of Delaware. His research centers on educational technology and learning sciences, specifically AI-enhanced learning and computational thinking.

Amanda Mohammad Mirzaei is an assistant professor in mathematics education at Manhattanville University. Their research focuses on teacher preparation, equitable math teaching methods, and calculus instruction. They have published on student engagement and noticing in secondary math classrooms, calculus instruction that considers prior knowledge, and teachers' orientation towards student engagement.

Chrystalla Mouza serves as Gutgsell Professor and Dean of the College of Education at the University of Illinois Urbana-Champaign. Her research focuses on the preparation of inservice and pre-service teachers on the use of technology, computer science education, and student learning in technology-based learning environment. Her work has been funded by the National Science Foundation, Google and other agencies and was honored with the Distinguished Research in Teacher Education award by the Association of Teacher Educators.

Lori Pollock is Professor Emeritus in the Department of Computer and Information Sciences at the University of Delaware. Her research focuses on computer science education and the design of large software systems. Her work has been funded by the National Science Foundation, the Army Research Laboratory, and Google. Throughout her career she has actively worked to improve the participation of women and students from underrepresented groups in computer science. He is the recipient of numerous teaching and research awards from key organizations in her field.

Kevin Guidry serves as Associate Director of Educational Assessment with the Center for Teaching and Assessment of Learning at the University of Delaware. He holds a Ph.D. in Higher Education and specializes in assessment of student learning and development of program educational goals. He works with faculty to align curricula and improve existing teaching practices aimed at enhancing student learning.

Authors and Affiliations

Yifan Zhang[1] · Amanda Mohammad Mirzaei[2] · Chrystalla Mouza[3] · Lori Pollock[4] · Kevin Guidry[5]

✉ Chrystalla Mouza
cmouza@illinois.edu

Yifan Zhang
zhangyifan@bnu.edu.cn

Amanda Mohammad Mirzaei
amanda.mohammadmirzaei@mville.edu

Lori Pollock
pollock@udel.edu

Kevin Guidry
krguidry@udel.edu

1 National Engineering Research Center of Cyberlearning and Intelligent Technology, Beijing Normal University, Jia No. 2 Manjing Road, Changping District, Beijing 102206, China

2 School of Education, Manhattanville College, Purchase, NY 10523, USA

3 University of Illinois Urbana-Champaign, 1310 S. 6th Street, Champaign, IL 61820, USA

4 Department of Computer and Information Sciences, University of Delaware, 18 Amstel Avenue, Newark, DE 19716, USA

5 Center for Teaching and Assessment of Learning, University of Delaware, 317 Hullihen Hall, Newark, DE 19716, USA