Reimagining Essential Computing Content for High School Students

Abstract

There are several changes anticipated in computer science (CS) education over the next decade, including updated student standards, rapidly changing impacts of artificial intelligence (AI), and an increasing number of school systems requiring a CS class for graduation. In order to prepare for these changes – as well as to address the equity issues that have plagued CS since its inception – we engaged in a project designed to reimagine content and pathways for high school CS education. As a collaborative project, we hosted multiple events for relevant parties (including K-12 educators and administrators, higher education faculty, industry professionals, state and district CS supervisors, and CS education researchers). These events were designed to collaboratively seek input for the creation of a series of reports recommending what a CS course that satisfies a high school graduation requirement should include, how that course should align with Advanced Placement (AP) and post-secondary CS instruction, and what pathways should exist for students after that introductory high school course.

The portion of the project highlighted in this article contains an analysis of data collected from focus groups ($n = 21$), interviews ($n = 10$), and an in-person convening of participants from K-12, post-secondary, industry, and administrative roles ($n = 35$). The data is centered on determining what CS content is essential for all high school students. Participants considered knowledge, skills, and dispositions across a range of CS and CS-adjacent topics and, through a variety of activities, described what new content should be taught when viewing through the lens of teaching CS to high school students in the year 2030 and what content should be prioritized. Our analysis sought to delineate and synthesize their sentiments. Six major priorities emerged from our analysis: societal impacts and ethical issues, algorithmic thinking, data and analysis, inclusive computing culture, AI, and career knowledge. The significance of our findings is that they present a broad overview of what a variety of relevant parties consider to be the most important CS content for high school students; this information is important for educators, administrators, and those who develop curriculum, standards, and/or teaching tools.

1 Introduction and Background

The field of computer science (CS) has undergone rapid changes in the last decade as new tools and technologies have come to prominence and as computing has expanded its reach into new areas, such as autonomous vehicles. Further, the effort to encourage CS for all students has resulted in an increased diversity in the population of high school students who study CS; it has also led these students to have more pre-high school CS experience than was previously the case. This trend is likely to continue: in recent years, eight states have adopted CS graduation

requirements, and it is a pending policy priority in many others [1]. More changes are anticipated as generative AI changes how people – including software developers – interact with computers and as other technologies are developed. All of these changes – including the ones that are difficult to anticipate – can impact how and what CS should be taught to students in high school.

Over half of U.S. states now require CS to be taught in all of their high schools, with the majority of those states requiring it in their elementary schools as well [1]. And, students enjoy CS and want to learn it. More high school students like CS classes than any other subject outside the arts [2]. As younger generations are born into and raised in a technologically advanced society, today's students understand that CS will be crucial for their careers and lives [3]. As a result, more than half (58%) of high schools in the US offer a foundational CS course, representing significant growth (from 35% in 2018) [4] in a subject that is critical to the nation's economic and security health.

There is now unprecedented support from business, nonprofit, and community leaders advocating for CS education. In July 2022, a letter encouraging governors and education leaders to make CS part of the standard K-12 curriculum across the U.S. was released with more than 800 signatories (a full list can be found at `www.CEOsforCS.com`). An excerpt of the letter follows:

> The undersigned commit our support by collectively creating employment opportunities for CS students in every city in the USA, and in every sector, from manufacturing to banking, from agriculture to healthcare. Many of us offer internships to help these students find their career pathway. Many of us have funded efforts in CS education, to support underserved communities. But there is only so much industry can do by ourselves [5].

This project seeks to align these various groups, at a time when momentum is on CS education's side and the College Board is likely to revisit its course frameworks, to create an updated version of its courses that could increase access, participation, high school and college course credits, and have it aligned to industry needs. This effort can broaden participation at numerous levels across the CS education field.

To achieve this, our team gathered input from a variety of relevant parties to articulate what CS content is essential for high school students in the next five to ten years, with follow on work focused on defining pathways for CS study beyond an initial high school CS course. The recommendations generated by this project will be used to inform future CS standards and course alignment. This paper describes the first portion of the project's effort: to define essential CS content for all high school students, including in settings where a CS course is a graduation requirement.

2 Methods

To ensure a collaborative process, we leveraged four strategies: engaging with participants at in-person convenings, in virtual focus groups of participants who did not attend the in-person convenings, with undergraduate students, and by providing opportunities for others to review our findings asynchronously.

For the professional participants, we recruited through a variety of avenues. Participants could

select how they wanted to be engaged: attend the in-person convenings, participate in focus groups, and/or review interim and final reports. We invited participants via a variety of professional associations for those working in high school and higher education CS education roles; we received about 300 applications for participation. From that group, we selected participants for the focus groups and for the in-person convenings, seeking to create participant pools that were diverse in terms of demographics, professional roles, and areas of expertise.

In addition, we conducted three focus groups in the fall of 2023: one each for high school CS teachers, higher education CS faculty, and those with experience working in industry. Participants in the focus groups were not attending the convenings. Each focus group met three times, for one hour each. Most of the time was devoted to facilitated discussion, but other activities – such as an exercise in choosing a limited set of content for a high school CS course – were used to prompt thought and discussion. Focus group meetings were recorded, and the recordings were transcribed and analyzed; the focus of the analysis was to determine what CS content participants prioritized for a foundational high school CS course.

To ensure that student perspectives were also included in our findings, we asked in-person convening participants to share an invitation to college students studying CS asking them to participate in an interview; of the four dozen student applicants, about one dozen were invited to participate, and nine were interviewed. One CS faculty member who could not attend the focus groups was also interviewed.

Professional participants were not compensated, but many were provided with travel support to attend the convening. Student participants were offered a $20 gift card for their participation.

2.1 Focus Groups

We hosted three different focus groups (one each for high school, higher education, and industry professionals), with each group holding three one-hour virtual meetings in the fall of 2023. Sessions focused on discussion between participants, with some variation between the groups based on participant expertise. For example, the industry group considered what skills employers were likely to need in the coming years, while high school teachers explored the characteristics of an exemplary CS class. All groups discussed what CS content they did (and did not) believe to be essential for all high school students.

Audio and video recordings of the sessions were created and later transcribed. These transcriptions were coded abductively [6], following the process for abductive thematic analysis articulated by Thompson [7]. We centered the analysis on identifying (1) what CS content participants prioritized for a required high school CS course and (2) disagreements and tensions between participants about those priorities.

2.2 Convenings

We selected 35 convening participants; their primary roles are shown in Figure 1 and their experience in Figure 2 (note that participants may have indicated experience in more than one area). Participants were chosen to represent a diversity of demographic groups and experiences (68% identified as women; 51% identified with a racial/ethnic group other than white, and 14% identified as disabled).

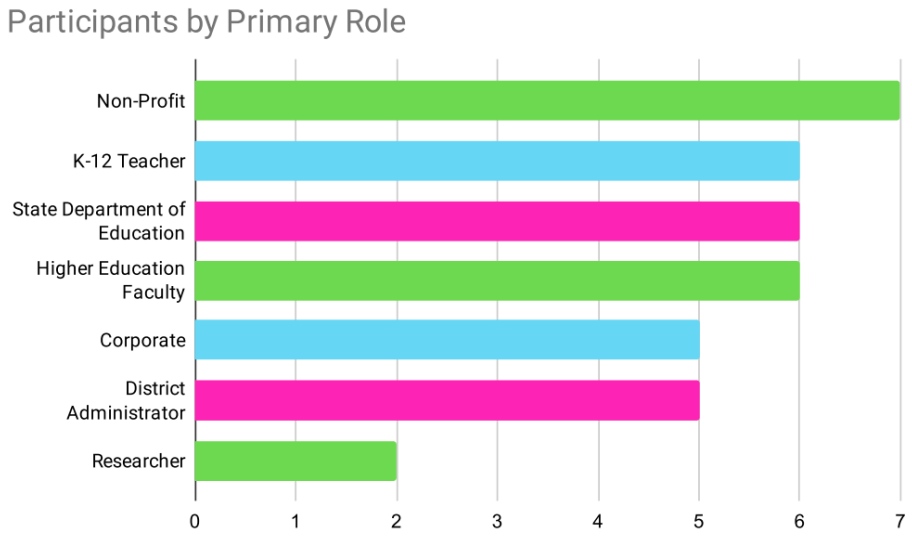

Figure 1: Primary role of convening participants.

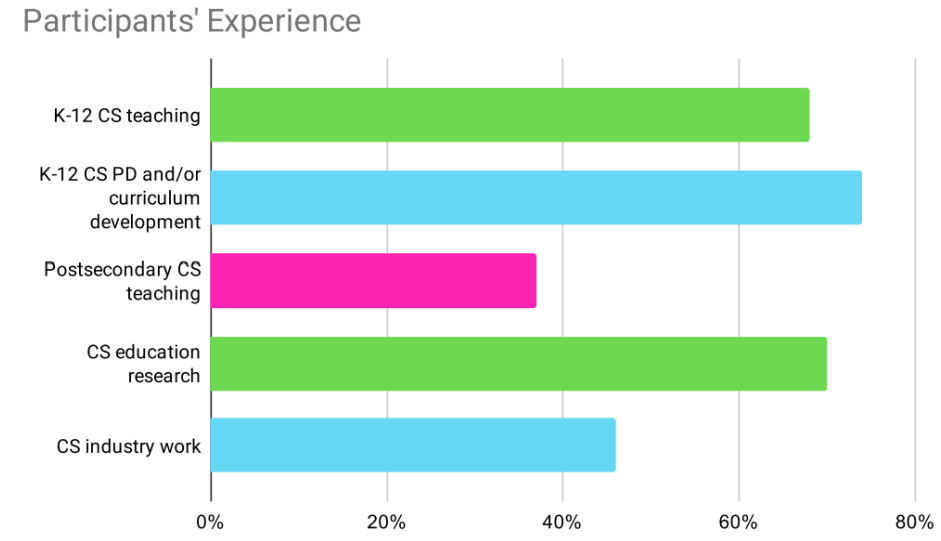

Figure 2: Experience of convening participants.

Figure 3: One of the personas used at the convening to frame the discussion of essential high school computer science content

The in-person convening was held over two days in November, 2023. The event was framed around a set of a dozen personas, each depicting a person one decade into the future. The personas featured a diverse set of individuals, working in a variety of careers, with different interests and backgrounds. Figure 3 shows one of the personas.

Participants used the personas to consider what high school CS content would have best prepared each persona for their professional and personal pursuits. Then, participants generated, organized, and prioritized that content. The result was a listing of over 200 possible topics to cover in a foundational high school CS course. A variety of artifacts, including digital feedback and physical votes, were collected and analyzed from the convening.

2.3 Student Interviews

The student interviews ($n = 9$) involved students who had taken at least one CS course in college. Table 2.3 contains demographic information about the students who were interviewed; note that the gender and race/ethnicity were free response questions in the application form, so the categories chosen reflect participants' self-designation.

Participant	Gender	Race/Ethnicity
1	Female	Hispanic
2	Female	White
3	Male	Hispanic
4	Male	Asian
5	Female	White
6	(Transgender) Female	Caucasian
7	Male	White
8	Male	Black African
9	Male	Asian

Table 1: Student interview participant self-designations for gender and race/ethnicity.

These interviews focused on exploring what CS courses the participant had taken in high school

and in college, including how prepared they felt for these courses. Each interview was 20 - 30 minutes long. Students were also asked whether they thought a CS course should be required in high school and, if so, what content it should include. These interviews were recorded, transcribed, and analyzed.

2.4 Asynchronous Reviews

We invited those who applied to participate but who were not selected for the in-person convenings or for the focus groups to provide asynchronous feedback. We invited this group to provide feedback in the form of voting on what they would choose at the top priorities for a foundational high school CS course from dozens of items that were generated by the focus groups and/or are part of current CS standards (participants could also add their own items). Respondents ($n = 138$) were each given 10 votes to apply to any combination of content items (i.e., they could apply more than one vote to an item). Items garnering at least 40 votes are shown in Table 2.4.

Item	Vote Count
Understand basic principles of computational thinking	71
Define and demonstrate sequence, iteration, and selection	54
Complete an individual programming project	49
Debug a program	46
Be persistent in solving computing problems	45
Work collaboratively with peers on computing problems	44
Create, test, and refine computational artifacts	43
Be familiar with different branches of computer science	42
Identify computer science terminology	41
Recognize, define, and analyze computational problems	40

Table 2: Items earning at least 40 votes in asynchronous feedback.

3 Results

We identified six major themes as described below.

1. Societal impacts and ethical issues

Particularly for the participants at the in-person convening, CS content related to the impacts of computing on society and ethical issues related to computing technologies were heavily prioritized, more so than any other topic. This broad topic includes content related to equity, social justice, and ethics, among other concerns. Content related to societal impacts and ethics was prioritized both as its own topic and as integrated into other topic areas.

2. Algorithmic thinking

Algorithmic thinking is defined by Lockwood et al. as "a logical, organized way of thinking used to break down a complicated goal into a series of (ordered) steps using available tools" [8]. More specifically, it can be conceptualized as a set of skills for addressing a problem that includes analysis, specification, determining needed actions, algorithm construction, special case (e.g., edge case) consideration, and efforts to improve solution efficiency [9]. While basic programming

skills (e.g., the use of conditionals) were included by participants in the prioritized content, algorithmic thinking skills were generally given a higher priority.

3. Data and analysis

Handling data (including cleaning and processing) and analyzing it were topics prioritized by participants. This prioritization reflects how data is increasingly generated, gathered, and analyzed in a variety of contexts, including those with serious societal implications (e.g., facial recognition in policing).

4. Inclusive computing culture

Participants consistently prioritized the development of an inclusive computing culture as key content for a foundational high school CS course. This topic covered a variety of knowledge and skills, including developing accessible computing artifacts, recognizing and addressing biases, and respecting diverse perspectives.

5. Artificial intelligence

In contrast to other topics, AI was not prioritized as a discrete topic but rather was often included in other topic areas, particularly data analysis, algorithms, ethics and impacts, and programming. AI was referenced as both a tool and as a subdomain of CS. That is, the skill to use AI tools (e.g., Copilot) to aid in computing was prioritized, as was AI as a subdomain of CS (similar to how cybersecurity is considered a subdomain of CS).

6. Career knowledge

Participants prioritized content that exposed students to careers – both those within computing as well as those in other fields that are likely to involve computing skills. This prioritization may reflect the burgeoning variety of careers in computing in addition to ways in which computing technologies (e.g., generative AI) are likely to be used in non-technical roles.

During the in-person convening, we asked participants to compare how what is currently taught in foundational high school CS courses compares to what CS content is essential for all high school graduates in 5 - 10 years. The results are shown in Figure 4.

4 Discussion

Figure 5 is a visualization of the content prioritized by participants, particularly participants of the in-person convening. Surrounding all other content is *dispositions*; the dispositions identified as particularly important include persistence, reflectiveness, creativity, curiosity, critical thinking, a sense of belonging in computing, and resourcefulness.

Surrounding the five topic areas are four cross-cutting concepts; these concepts are not taught on their own but rather are integrated into all topic areas. These concepts are:

- *Impacts and Ethics* involves discussions of societal impacts of computing, ethical issues in computing, social justice issues, access and equity concerns, ensuring that computing benefits all members of society (especially the most vulnerable), and safety and privacy issues.

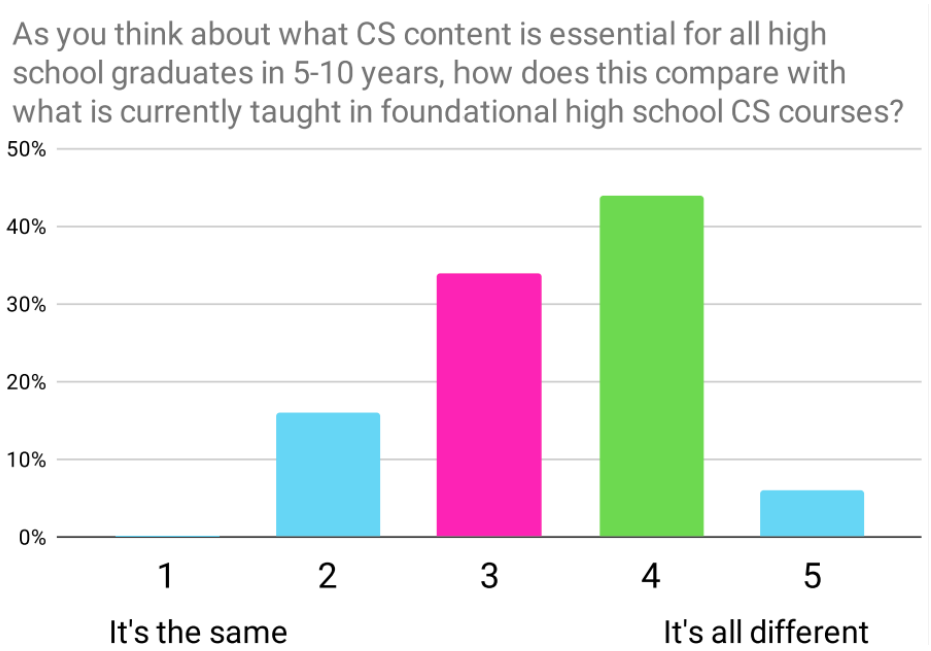

Figure 4: In-person convening participants' sense of the similarity between (1) what is currently taught and (2) what content is essential in the near future.

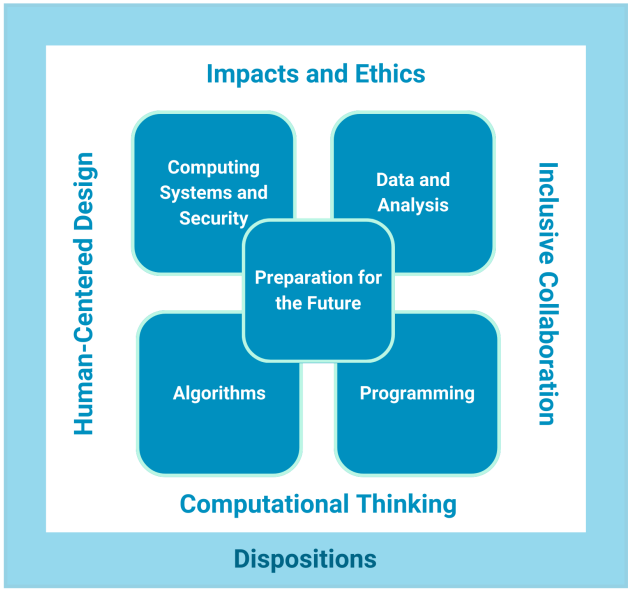

Figure 5: Participants' recommendations for essential content

- *Inclusive Collaboration*, in the words of one participant, is to "engage with diverse perspectives with respect and empathy." It includes:
  - awareness and empathy with others, including accommodating a variety of identities and perspectives (including from those with disabilities and/or from different cultural backgrounds), recognizing and mitigating personal biases, providing services to other

people and groups via computing, supporting the learning of others, and designing and developing with accessibility in mind.

  - collaboration skills, including recognizing different roles on a team and being able to assume different roles, seeking out and using feedback from others, providing others with constructive feedback, advocating for the needs of others, using appropriate tools (including digital tools) for collaboration, using a variety of models and methods for collaboration (including pair programming), being able to communicate about technology in a variety of contexts and with those with limited technical knowledge, being able to document products and processes, applying principles of digital citizenship (including data security, responsible communication, information evaluation, and respect for intellectual property).

- *Computational Thinking* includes practices such as abstraction, decomposition, and pattern recognition.
- *Human-Centered Design* involves the following skills: understanding principles of effective design, including identifying problems and understanding underlying causes; empathizing with those impacted by problems, including applying strategies for accessible design; thinking of everything and approaching solutions as a system; ideating to solve the identified problems; prototyping, testing, and iterating; and understanding how design decisions shape computing.

The five topic areas are Computing Systems and Security, Data and Analysis, Preparation for the Future, Algorithms, and Programming. We used Bloom's Revised Taxonomy [10], including the application of the taxonomy specific to computing education developed by Tang et al. [11]. Table 3 contains the content associated with each topic area.

In general, participants' priorities overlapped substantially with content commonly included in foundational CS courses (e.g., in the CSTA K-12 Standards), particularly for the in-person participants. The main divergence from this trend is the prioritization that in-person participants placed on knowledge of careers, which is a topic not widely covered in other efforts to define essential CS content. It may be that framing the convening around personas led participants to think more directly about the issue of careers, especially since most of the personas did not work in tech roles (e.g., software developer). It may also be the case that career changes anticipated in the wake of generative AI made the topic of careers more salient to participants.

We note two main challenges to this work. First, participants frequently expressed that it was difficult to project a decade into the future and determine what CS content would be beneficial; this made articulating CS course content difficult. Second, it was difficult for both participants and for the project team to organize prioritized content into categories in a way that minimized gaps and duplication.

5 Conclusion

Future work on this project will involve two additional convenings, which will explore pathways for students beyond the foundational CS course and refine work on essential content and pathways. We will also seek input from interested parties on interim reports of the convening, in preparation for issuing recommendations for a foundational course and for CS pathways.

The significance of our findings thus far is that they present a broad overview of what a variety of relevant parties consider to be the most important CS content for high school students; this information is important for educators, administrators, and those who develop curriculum, standards, and/or teaching tools.

6 Acknowledgements

This material is based upon work supported by the U.S. National Science Foundation under Grant No. 2311746. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

[1] Code.org, CSTA, and ECEP Alliance. 2023 state of computer science education, 2023. URL https://advocacy.code.org/stateofcs.

[2] Hadi Partovi. What classes do students like the most?, June 16 2016. URL https://blog.code.org/post/146020540698/whatclassesdostudentslikethe-most. Code.org Blog.

[3] Google and Gallup. Computer science learning: Closing the gap: Rural and small-town school districts results from the 2015-2016 Google-Gallup study of computer science in U.S. K-12 schools (issue brief no. 4), 2017. URL https://services.google.com/fh/files/misc/computer-sciencelearningclosingthegapruralsmalltownbrief.pdf.

[4] Code.org, CSTA, and ECEP Alliance. 2022 state of computer science education: Understanding our national imperative, 2022. URL https://advocacy.code.org/stateofcs.

[5] CEOS for CS. A message from over 800 business and nonprofit leaders. URL https://www.ceosforcs.com.

[6] Stefan Timmermans and Iddo Tavory. Theory construction in qualitative research: From grounded theory to abductive analysis. *Sociological theory*, 30(3):167–186, 2012.

[7] Jamie Thompson. A guide to abductive thematic analysis. 2022.

[8] Elise Lockwood, Anna F DeJarnette, Autumn Asay, and Matthew Thomas. Algorithmic thinking: An initial characterization of computational thinking in mathematics. *North American Chapter of the International Group for the Psychology of Mathematics Education*, 2016.

[9] Gerald Futschek. Algorithmic thinking: the key for understanding computer science. In *International conference on informatics in secondary schools-evolution and perspectives*, pages 159–168. Springer, 2006.

[10] Mary Forehand et al. Bloom's taxonomy: Original and revised. *Emerging perspectives on learning, teaching, and technology*, 8:41–44, 2005.

[11] Cara Tang, Markus Geissler, and Christian Servin. Bloom's for computing: crafting learning outcomes with enhanced verb lists for computing competencies. *Journal of Computing Sciences in Colleges*, 38(1):114–115, 2022.

	Algorithms
Remember	Define algorithm and explain what algorithms are used for
	Recognize that computational solutions take in information, store and process it, and produce a result
Understand	*Describe the difference between traditional and AI/ML algorithms and, at a high level, describe how AI/ML algorithms work*
	Explain why/how sequence matters in an algorithm
	Describe patterns/commonalities in problems, data, and programs
	Interpret algorithms
Apply	Modify an algorithm (e.g., to add functionality)
	Apply strategies for learning what is inside of an opaque system when it is necessary to do so
Analyze	*Decompose a problem into multiple subproblems*
Evaluate	Evaluate (at a high level) the trade-offs (e.g., speed) of different algorithms
	Evaluate the appropriateness, reasonableness, and/or effectiveness of an algorithm for a specific task, including via algorithmic auditing
	Assess societal impacts of the application of computational thinking and related ethical issues (e.g., use of AI algorithms to choose job candidates, use of abstraction to obscure important context)
Create	Compose algorithms using sequence, selection, and iteration
	Create a variety of abstractions and models to represent a system
	Programming
Remember	Locate common programming constructs (e.g., using online tools)
Understand	Convert an algorithm to code
Apply	Apply programming skills in text-based and non-text-based programming contexts (e.g., block-based, kiosk, prompt engineering)
	Modify a program (e.g., to add functionality or improve usability or accessibility)
	Use programming assistive technologies (e.g., Copilot) to plan, write, test, and debug code
Analyze	Articulate whether a program solves a given problem
Evaluate	*Systematically test and debug a program, including the use of skills such as code tracing*
	Evaluate whether and how computation can or cannot help to solve a problem
	Assess societal impacts of programming and related ethical issues (e.g., how might modifications to a program impact various groups of users?)
Create	*Develop programs using sequencing, selection, and iteration*
	Data and Analysis
Remember	*Identify and define data types (e.g., string, numeric, Boolean)*
	Identify basic data formats (e.g., tables, schemas, JSON)
Understand	Describe, at a high level, the role of data in AI/ML applications
	Understand the difference between data and metadata
Apply	Manipulate (e.g., normalize, transform, clean) data
Analyze	Trace how data moves through a program
Evaluate	Evaluate approaches to cleaning data in a given context
	Assess whether and how a given question can be answered with data, and what specific data is needed
	Assess societal impacts of data analysis and related ethical issues (e.g., biased data used to train AI systems, attribution related to products of generative AI)
	Evaluate data visualizations for clarity, potential biases, etc.
Create	*Select, organize, interpret, and visualize large data sets from multiple sources to support a claim and/or communicate information*
	Devise plans for using data to solve a problem
	Computing Systems and Security
Remember	Identify various types of hardware (including components) and software (including operating systems)
	Enumerate security practices (e.g., safe passwords, two-factor authentication)
Understand	Describe why cybersecurity is important
	Explain what networks (including the Internet) are and how they work
	Explain how an operating system, other software, and hardware work together
Apply	Manipulate operating systems and other software settings to achieve goals
	Apply knowledge of the structure and function of various technologies (e.g., cloud computing, sensors, GPS, embedded/IoT, phones/tablets, gaming consoles, medical devices, VR, robotics) to optimize their use (for example: explain why GPS can be used without Internet access)
	Use documentation and other resources to guide tasks such as installation and troubleshooting
Analyze	Detect vulnerabilities in networks
	Analyze a problem to determine appropriate troubleshooting strategies
Evaluate	Assess societal impacts of networks and related ethical issues (e.g., digital divide)
Create	Design projects that combine hardware and software that collect and exchange data
	Preparing for the Future
Remember	*Identify pathways and careers that involve computing*
Understand	Explain how computing enables emerging technologies such as autonomous vehicles and how these emerging technologies are applied in various industries
Apply	Apply computing concepts to other disciplines
Analyze	Examine how emerging technologies are impacting a variety of practices (e.g., use of facial recognition in policing, AI-generated news products)
Evaluate	Assess societal impacts and related ethical issues of emerging and future developments in computing (e.g., the impact of quantum computing on security)
	Evaluate the use of emerging technologies (e.g., generative AI) for accuracy (e.g., detect hallucinations) and to meet specific needs
Create	Plan how an emerging technology could meet a need
	Generate a personal career plan that involves computing

Table 3: Prioritized content by level of Bloom's Revised Taxonomy; italics indicate top priorities.