2026 CSTA PK–12
COMPUTER SCIENCE STANDARDS

Authors and Leadership

CSTA

Bryan Twarek, Head of Research & Innovation
Jacob Koressel, K–12 Standards Project Manager
Jake Baskin, Executive Director

WestEd

Dr. Yvonne Kao, Senior Director, Learning Sciences and Technology
Dr. Aleata Hubbard Cheuoua, Senior Research Scientist, Learning Sciences and Technology
Megan Schneider, Program Manager, Measurement and Evaluation

Writing Team

Elementary Team

- **Vicky Sedgwick** (Team Lead), Chapter President, CSTA Greater Los Angeles, *Canoga Park, CA*
- **Dr. Raymond Anacaya**, Teacher, Olanta Creative Arts and Science Magnet School, *Florence, SC*
- **Dr. Steven Azeka**, Program Lead, Mozilla Foundation, *New York, NY*
- **Dr. Lea Ann Christenson**, Professor, Towson University, *Columbia, MD*
- **Smita Kolhatkar**, Assistant Head for Innovation, Responsible AI & Ed Tech, Gideon Hausner Jewish Day School, *Palo Alto, CA*
- **Corey Rogers**, Education Producer, Micro:bit Educational Foundation, *Iowa City, IA*
- **Sara Schnell**, Standards Supervisor, Wyoming Department of Education, *Bushnell, NE*
- **Dr. Perry Shank**, Chief Learning Officer, CodeVA, *Harrisonburg, VA*
- **Jessica Yauney**, PhD Student, Stanford University, *Stanford, CA*

Middle School Team

- **Justin Cannady** (Team Lead), Head of Learner Experience and Design, Northern Lights Collaborative for Computing Education, *Birmingham, AL*
- **Rebekah Collipp**, Computer Science Innovation Specialist, New Jersey Department of Education, *Belle Mead, NJ*
- **Dr. Michelle-Noelle Magallanez**, Consultant, *Paris, France*
- **Melisa Nozière**, Educational Technologist, The School at Columbia University, *New York, NY*
- **Dianne O'Grady-Cunniff**, Consultant, University of Maryland Baltimore County, *Waldorf, MD*
- **Amanda O'Mara**, STEM Teacher, Gilmour Academy Lower School, *Cleveland Heights, OH*
- **Dr. John Underwood**, Advocate and Researcher, *Baton Rouge, LA*

High School Team

- **Jacqueline Corricelli** (Foundational Standards Lead), District Computer Science Curriculum Specialist, West Hartford Public Schools, *West Hartford, CT*
- **Dr. Beth Simon** (Specialty Standards Lead), Professor, University of California San Diego, *San Diego, CA*
- **Melissa Bullen**, Teacher, Southeast High School, *Warr Acres, OK*
- **Cindi Chang**, CTE Director, Clark County School District, *Henderson, NV*
- **Tiffany N. Jones**, Teacher, Global Impact Academy STEM Magnet High School, *Fairburn, GA*
- **Deborah Kariuki**, Graduate Program Director, University of Maryland Baltimore County, *Ellicott City, MD*
- **Dr. Sonia Mitchell**, Educator and Coach, Academy of Computer Science and Engineering, *Springfield, MA*
- **Jigar Patel**, Director of Innovations & Special Projects, Tuscarora Intermediate Unit 11, *Jonestown, PA*
- **Andrew Spiece**, Teacher, Genesee Intermediate School District, *Fenton, MI*

Supporters

CSTA gratefully acknowledges the generous support of the following organizations that made the development of these Standards possible. Their commitment to advancing computer science education ensures that teachers and students across the nation and world have the resources they need to thrive in a world powered by computing.

- Association for Computing Machinery (ACM)
- Amazon
- Google
- Microsoft

CSTA also wishes to thank the Kapor Foundation for supporting the *Amplifying Social Impacts of Computing Standards (ASICS)* initiative.

This material is based upon work supported by the U.S. National Science Foundation (NSF) under Awards No. 2444214, 2417779, 2311746, and 2118453. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

Suggested Citation

Computer Science Teachers Association. (2026). *2026 CSTA PK–12 computer science standards*. https://csteachers.org/pk12standards/

DOI: https://doi.org/10.1145/3820482 **ISBN:** 979-8-4007-2780-1

License

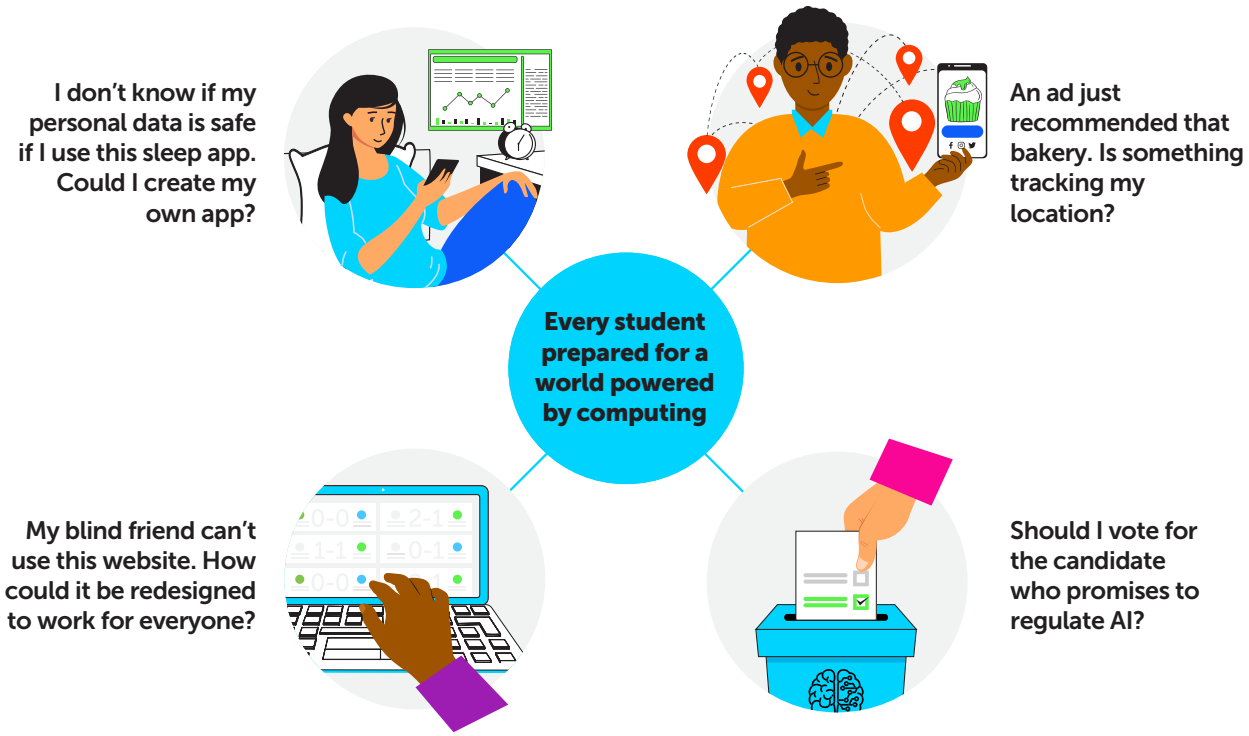
Table of Contents

Vision

Every Student Prepared for a World Powered by Computing

Foundational knowledge of computer science (CS) is necessary for students to understand and address the questions they face today. For example, they need to shape their views on the regulation of artificial intelligen e (AI), have the ability to automate routine tasks, and analyze and visualize data in a variety of contexts. These realities highlight the importance of early, universal CS education, a need that will only grow as society becomes increasingly reliant on computing technologies.

Figure 1. CSTA Vision



At the same time, access to meaningful CS learning opportunities has historically been uneven, with many students—girls, Black, Latine, and Indigenous students, multilingual learners, students with disabilities, and students in rural or economically disadvantaged communities—facing barriers to participation.

In a world powered by computing, students of all identities and career aspirations need high-quality computer science education to become informed participants in society and confident creators. Our vision for PK–12 CS education is to ensure:

- All students are engaged and supported in learning CS, including its impacts on individuals, societies, cultures, and economies.
- Policies, pedagogies, and practices actively support all students in learning CS.
- Standards align with the current and future needs for learning CS.
- Students are empowered to responsibly shape the technology-driven world of tomorrow.

Defining Computer Scienc

Computer science is the study and human-centered practice of using data, algorithms, and computing systems to solve problems, make discoveries, and express ideas.

Computer science integrates scientific easoning with creative expression. It helps people understand and shape a world powered by computing. As a discipline, computer science brings together algorithms, data, systems, and design to help people ask questions, explore patterns, model complex ideas, and build solutions that matter in their communities. It is both a scientific and a c eative discipline: a field g ounded in computational thinking and driven by human curiosity, imagination, and purpose.

Computer science continues to evolve as new technologies, applications, and societal challenges emerge. Foundational ideas such as algorithms, programming, and systems remain central while also enabling rapidly growing areas of computing such as artificial intelligen e, data science, cybersecurity, and physical computing. These developments increasingly raise questions about how computing shapes society, including issues of ethics, equity, and sustainability.

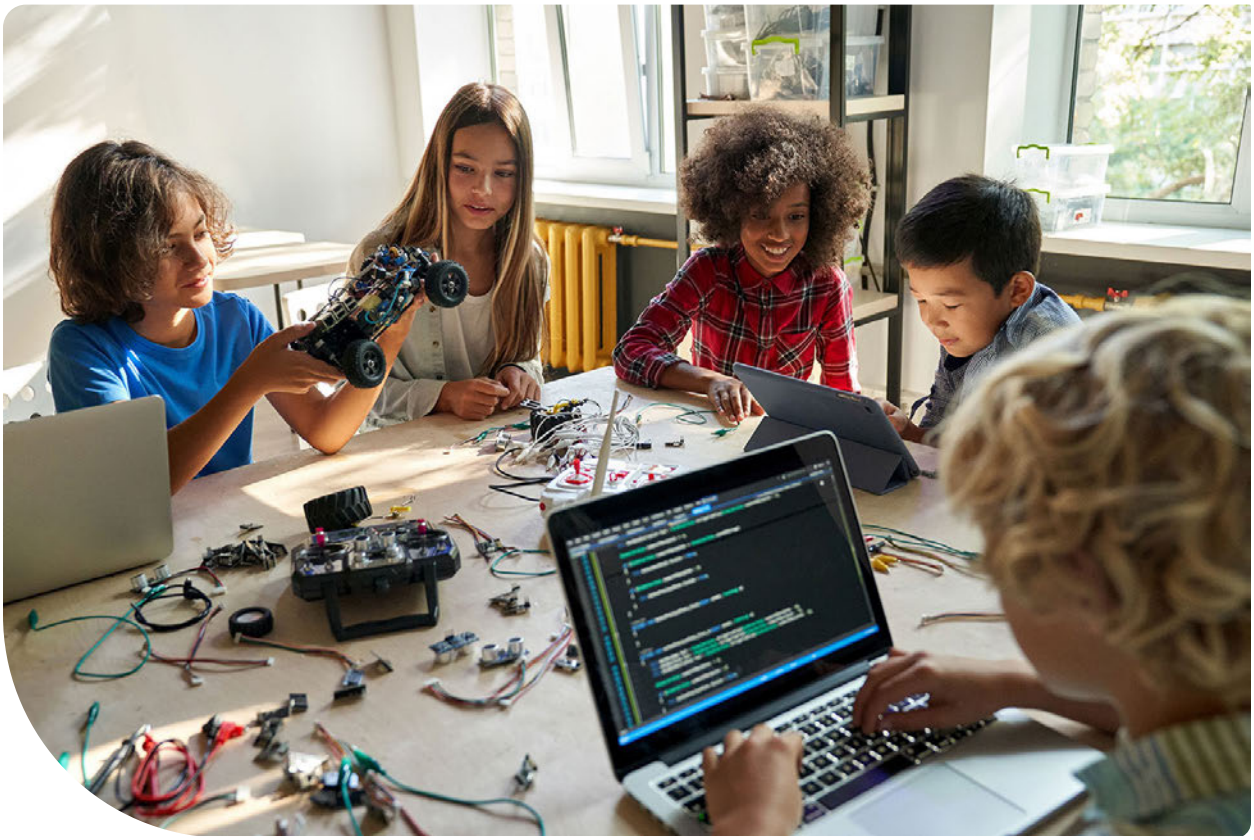
Computer Science Is:	Computer Science Is Not:
• **a scientific and creative discipline** focused on understanding and designing algorithms, data, and computing systems. • **the application of computational thinking** to develop both rules-based and data-driven solutions across a variety of disciplines and contexts. • **a collaborative discipline** in which learners plan, communicate, test, and refine ideas t design solutions that serve diverse people and communities. • **an ethical, responsible, and human-centered practice**, in which it is critical to examine impacts, identify potential harms and benefits, and desig responsibly.	• **using technology tools** like word processors, slide decks, or generative AI tools. These appear in CS and in every other subject too. • **limited to coding or programming**. Programming is an essential part of CS, but CS is much broader. • **tied to one career path.** CS has applications across every field an industry. • **being replaced by AI.** CS is what powers AI.

The CSTA PK–12 Computer Science Standards make this vision concrete by organizing learning around core **concepts** supported by crosscutting **practices**. Together, these elements emphasize that computer science is not only about understanding how technologies work but also about how people design, build, evaluate, and use computing technologies to solve problems, make discoveries, and express ideas across a variety of disciplines and contexts.

Concepts	Practices	Dispositions
• Algorithms & Design • Programming • Data & Analysis • Systems & Security • Computing & Society	• Ethics & Social Responsibility • Inclusive Collaboration • Computational Thinking • Human-Centered Design	• Creativity • Critical thinking • Curiosity • Persistence • Reflectivenes • Resourcefulness • Sense of belonging

When these concepts and practices are intentionally integrated, high-quality CS instruction fosters **dispositions** (habits of mind) that support lifelong learning and creative work with computing. Through these experiences, students come to see computer science not only as a powerful medium for expression and innovation, but also as a shared responsibility for shaping communities. **In a world powered by computing, CS education empowers students to become critical consumers, responsible creators, and informed participants in society.**

The following standards translate this vision into specific learning out omes for all students across all grade levels and contexts.



About the CSTA PK–12 Standards

The CSTA PK–12 Computer Science Standards (the Standards) define the essential knowledge, skills, and dispositions to prepare all students for a world powered by computing. Specificall , the Standards delineate coherent progressions of student learning outcomes from pre-kindergarten to grade 12 (PK–12). Together, they form the strong foundation for a rigorous and comprehensive computer science (CS) curriculum that is driven by research and informed by teacher practice. The Standards describe what students should know and be able to do in CS, but they do not prescribe specific curriculum, instructional m terials, or assessments.

Written *by* teachers *for* teachers, the Standards offer flexibility and guidan e to support state and local adaptation while also promoting instructional coherence across the United States and globally. Grounded in research and equity, the Standards emphasize creativity, ethics, data, and human-centered design. They integrate emerging technologies like artificial intelligen e while reinforcing foundational computing concepts and inclusive practices that make computer science relevant for every learner. The Standards are a primary resource for state and local education agencies when determining what PK–12 students need to know and be able to do in CS. Widespread adoption impacts millions of students by promoting consistency in state policy, curriculum development, teacher certific tion, teacher preparation, and professional development across the United States and beyond.

Intended Uses

The 2026 CSTA PK–12 Standards are designed for broad adoption and implementation:

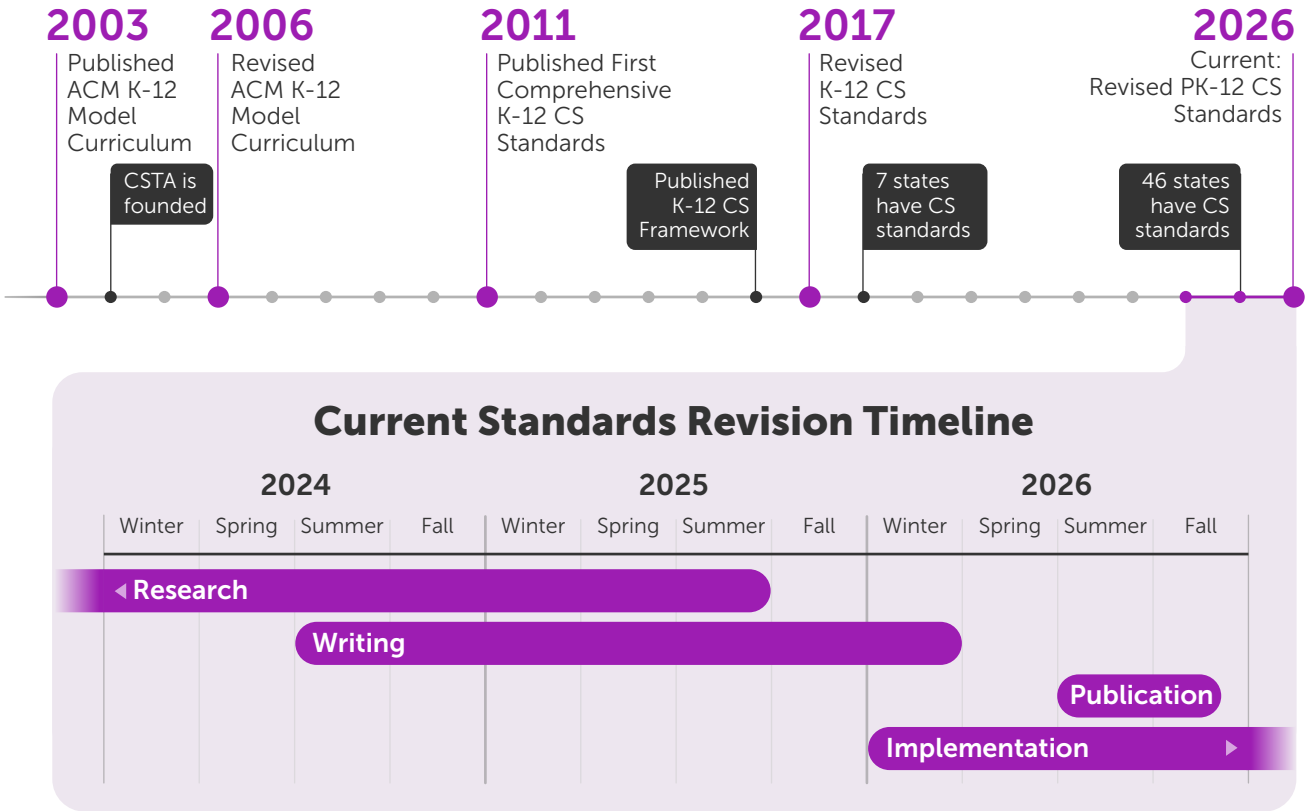
- **PK–12 CS teachers** use the Standards to design rigorous and relevant learning experiences for all students.
- **Other PK–12 teachers** use the Standards to plan how they can integrate CS across subject areas.
- **PK–12 administrators** use the Standards to establish and support district- and school-level policies that enable implementation and increase participation.
- **Curriculum providers** use the Standards to develop new, or refine existing, curriculum an associated tools that guide student learning.
- **Professional development (PD) providers** use the Standards to design in-service professional learning programs that prepare teachers to effectively teach CS across a variety of contexts.
- **Schools of education** use the Standards to ensure pre-service teachers have appropriate knowledge and skills to teach foundational CS to their future students.
- **State leaders** use the Standards to adopt and revise state CS standards, teacher certific tion, and other CS education policy decisions.
- **Researchers** use the Standards as a framework to guide study design, define constructs, and investigate the teaching and learning of computer science across diverse contexts.
- **Parents and community members** use the Standards to understand what PK–12 CS education includes and advocate for high-quality CS education in their communities.

Revision Overview

The CSTA PK–12 Computer Science Standards build on more than two decades of work by the computer science education community. Early model curricula were published by ACM in 2003 (Tucker, 2003) and revised in 2006 (Tucker et al., 2006), laying the groundwork for the first comprehensive CSTA K–12 Computer Science Standards released in 2011 (Seehorn et al., 2011). A revised set of Standards was published in 2017 (Seehorn et al., 2017) to align with the K–12 Computer Science Framework (2016), further strengthening coherence across CS education efforts.

The Standards required an update as both the field of computer science and the landscape of CS education have evolved significantly. Since the previous revision, CS implementation in schools has expanded dramatically, the body of CS education research has grown, and rapid advances in areas such as artificial intelligence and cybersecurity have raised new questions about what students should learn.

Figure 2. CS Standards Past & Present



Values

CSTA defined the following values to guide the revision process and provide a lens for reflecting on and defining project outputs:

	**Equity-centered:** Promotes broad and equitable access, participation, and experiences in CS education among all students.
	**Community-generated:** Meets the needs of the community, including PK–12 educators, postsecondary institutions, students, parents, and industry.
	**Future-oriented:** Anticipates future needs of current learners, and prepares them for a future that is increasingly reliant on computing.
	**Grounded in research:** Reflects the evolving body of knowledge of how students learn CS.
	**Flexible in implementation:** Considers multiple pathways for meeting individual needs of learners, including regional, cultural, ability, social, and economic factors.

People

The standards revision process involved the coordinated efforts of three main groups.

- **Standards Writing Team:** Included 25 individuals representing 18 states that serve in a variety of roles, including CS teacher (44%), state CS specialist (16%), teacher preparation (32%), and researcher (8%). The writing team collectively had over 400 years of teaching experience, more than 250 of those years specifically teaching C . Writers took the lead on drafting standards and made key decisions throughout the writing process.
- **Advisory Board:** Composed of approximately 70 accomplished individuals from across the CS education community, including state CS supervisors, district leaders, curriculum and PD providers, nonprofit leaders, global partners, esearchers, school of education faculty, and industry partners. Advisors provided additional insight and perspective that helped to ensure a useful, viable, and accessible final product.
- **Asynchronous Reviewers:** Approximately 450 asynchronous reviewers validated directionality and provided feedback at key intervals to inform the writing process.

Process

The Standards revision process was organized around three distinct and overlapping phases: Research, Writing, and Implementation.

Research

The research phase began with the *Reimagining CS Pathways* project (CSTA et al., 2024), which developed community definitions f foundational CS content for all high school graduates and pathways for continued learning. This project laid the foundation for identifying priorities for the Standards.

Additional research efforts informed the structure and content of the Standards, including:

- Comparing state and global CS education standards (CSTA & IACE, 2024, 2025)
- Conducting literature reviews of core concepts (CSTA, 2026a, 2026d, 2026f, 2026h)
- Comparing the 2017 CSTA Standards to other standards and frameworks (CSTA, 2026c)
- Analyzing the alignment of curricula to the 2017 CSTA Standards (CSTA, 2026i)
- AI Learning Priorities for All K-12 Students (CSTA & AI4K12, 2025)
- Amplifying Social Impacts of Computing Standards (Santo et al., 2025)

See Appendix A for a detailed summary of how research informed the revision process.

The development of the Standards reflects the ollaborative nature of the computer science education community. This revision builds upon a growing body of work developed by many organizations that have contributed research, frameworks, and instructional resources that inform how CS is taught and learned. Examples include the K–12 CS Framework (2016), AI4K12 Guidelines (2022), K–12 Data Science Learning Progressions (Data Science 4 Everyone, 2025), K–12 Cybersecurity Standards (CYBER.ORG, 2021), QISE K–12 Framework (National Q–12 Education Partnership, 2023), Justice-Centered Computing Education Framework (Scott et al., 2023), and CS2023 (Kumar et al., 2024).

Writing

The writing phase began in September 2024 and included six in-person convenings and regular virtual meetings with the writing team. Writers primarily worked in grade-band teams during drafting sprints, and concept teams and cross-grade groups collaborated to ensure logical progressions across grade levels. Advisors, researchers, and technical writers provided regular feedback throughout the process to support key decisions and refine dr fts. Three public drafts of the Standards were released during this phase, allowing educators and community members to provide feedback at key points in the revision process. A high-level timeline of the writing process is shown on the next page.

Fall 2024	Established the organizational **structure** of the standards.
Winter 2024–2025	Released **Draft 1.0** of the standards and collected community feedback on the organizational structure and writing approach.
Spring 2025	Completed draft progressions across all five concepts.
Summer 2025	Released **Draft 2.0** for feedback on coherence and clarity of progressions.
Fall 2025	Developed clarifications for each standard, including boundary statements and practice alignments.
Winter 2025–2026	Released **Draft 3.0** for feedback on clarity, granularity, and placement of individual standards.
Spring 2026	Finalized the standards and developed supplementary resources.
Summer 2026	**Published** the 2026 CSTA PK–12 Computer Science Standards.

Implementation

The CSTA PK–12 Computer Science Standards are designed not only to define what students should learn, but also to help the CS education community translate that vision into classroom practice. Educators, curriculum developers, professional learning providers, and state and district leaders all play a role in bringing the Standards to life through rigorous learning experiences that prepare students to understand and shape a world powered by computing.

View guidance and resources to support implementation at https://csteachers.org/pk12standards/resources

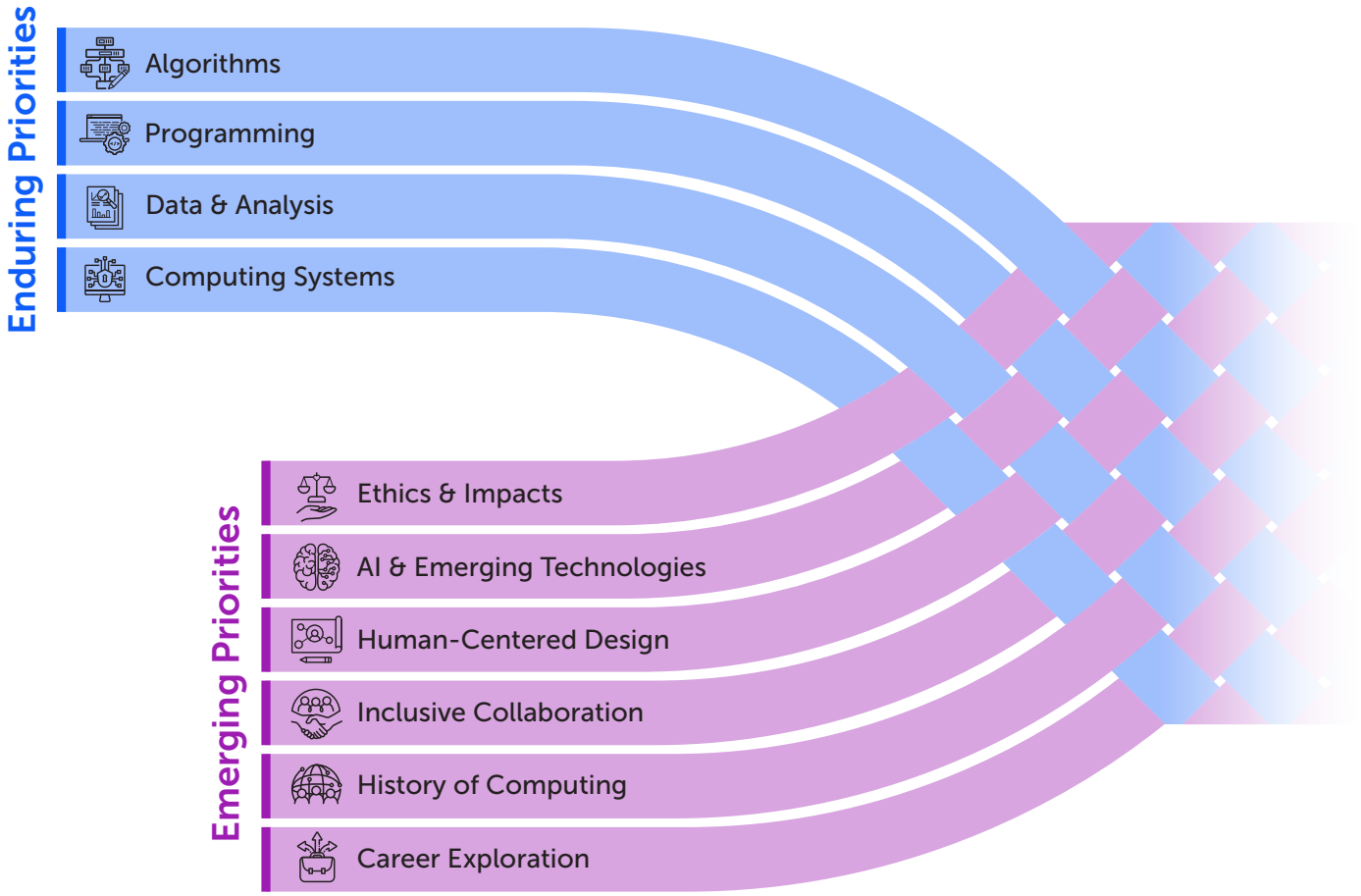
The Standards were written with practical implementation in mind. They assume that students experience a coherent progression of computer science learning across grade bands. In elementary school (grades PK–5), students engage in approximately 20–40 hours of CS learning per year. In middle school (grades 6–8) and high school (grades 9–12), students experience the equivalent of a yearlong course at each grade band. These learning opportunities may occur through dedicated computer science courses or through thoughtful integration across subject areas. Because schools and districts vary widely in their schedules, resources, and instructional models, the Standards are designed to support flexible implementation across diverse school contexts while maintaining coherent learning progressions for students.

Implementing the Standards will require sustained collaboration across the CS education community. As curriculum, professional learning, and policy increasingly align with the Standards, schools will be able to offer more coherent and accessible computer science learning opportunities. Over time, these efforts will help ensure that all students develop the knowledge, skills, and dispositions needed to participate in and shape a world powered by computing.

Priorities for the 2026 CSTA PK–12 Standards

The 2026 CSTA PK–12 Computer Science Standards maintain core disciplinary foundations while responding to advances in computing, research on CS learning, and lessons from scaled implementation. The priorities below highlight areas that remain central to computer science education as well as areas that have been newly emphasized or expanded.

Figure 3. Enduring and Emerging Priorities



Enduring Priorities

Core disciplinary concepts that remain central in the updated Standards include Algorithms, Programming, Data & Analysis, and Computing Systems.

Algorithms

Algorithms receive increased emphasis in the 2026 CSTA PK–12 Standards. To reflect this shift, Algorithms & Design is defined as a distinct oncept rather than being combined with Programming as in previous versions. This emphasis is reinforced through the designation of Computational Thinking as a category of practices woven throughout instruction.

Programming

Programming is still prioritized, and the value of learning to program remains a core aspect of foundational CS. However, the updated Standards represent a shift from focusing primarily on writing code toward a more balanced approach that also emphasizes reading, evaluating, modifying, and debugging programs.

Data & Analysis

Content related to data and its analysis is also a priority, reflecting the g owing role of data in daily life and the central role data plays in technologies such as AI. This trend also acknowledges data science as a burgeoning and increasingly important field with st ong foundations in CS.

Computing Systems

The 2017 CSTA K–12 Standards included Computing Systems and Networks & the Internet as two distinct concepts. In the updated Standards, these areas are combined into a single concept, Systems & Security, to reflect their inte connected nature and highlight the increasing importance of security in the responsible use of computing technologies.

Emerging Priorities

The following areas represent new or expanded priorities in the 2026 Standards as compared to previous versions.

Ethics & Impacts

Ethical practices and the societal impacts of computing are central priorities in the updated Standards. The Ethics & Social Responsibility practices emphasize responsible computing behaviors such as using computing for social good and respecting creators. Impact-related content appears throughout the Standards, including Impacts of Algorithms & Design, Impacts of Computing Systems, Impacts of Data Science, and Computing & Society. Together, these elements aim to develop learners who are responsible creators and critical consumers of technology, aware of its impact on their lives and communities.

Artificial Intelligence and Other Emerging Technologies

To ensure the Standards remain relevant over the next decade, they address foundational ideas related to artificial intelligen e and other emerging technologies. AI-related learning is integrated across all concepts, and a distinct Emerging Technologies progression within the Computing & Society concept creates space for students to explore both current innovations and future developments in computing.

Human-Centered Design

Human-Centered Design practices are crucial for the ethical development of computing technologies. The level to which a variety of potential user needs, abilities, and contexts are considered in the design process can either remedy or exacerbate equity issues in implementation and profoundly impact the benefits and harms experien ed by users. Human-Centered Design practices are woven throughout all CS concepts.

Inclusive Collaboration

The prioritization of Inclusive Collaboration as a category of practices highlights equity as a core value. It emphasizes key skills like effective communication, collaborative problem-solving, and navigating computing projects. The practices that define Inclusive Collabor tion are integrated across all CS instruction.

History of Computing

To fully grasp their sociotechnical world, students must explore the evolution of computing technologies, from early developments to modern innovations, and recognize the key contributors. This priority area is integrated within the Computing & Society concept.

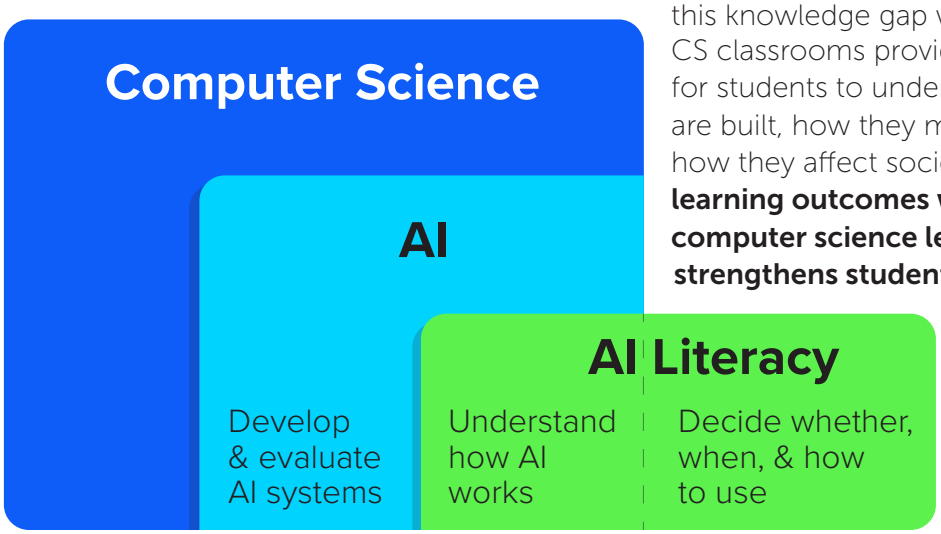
Career Exploration

Computing is foundational to nearly every industry and field f study. As such, it is critical for students to connect computing to their personal interests and career goals. Career-related standards are included in the Computing & Society concept.

The following sections describe how the updated Standards address two areas of particular importance in computer science today: artificial intelligence and the ethics and societal impacts of computing.

Artificial Intelligence as Part of Computer Science

AI increasingly shapes students' daily lives, yet many cannot explain how these systems work, why they fail, or how data influen es their behavior. Without intentional learning opportunities, this knowledge gap will continue to widen. CS classrooms provide a natural context for students to understand how AI systems are built, how they make decisions, and how they affect society. **Including AI learning outcomes within foundational computer science learning experiences strengthens students' ability to become critical consumers, responsible creators, and informed participants in society.**



Critical Consumers

From the media they consume to the jobs they pursue to their doctor's visits, students encounter AI systems throughout their daily routines. They must recognize when AI is being used, question whether it is appropriate for a task, evaluate if outputs are trustworthy, and assess whether decisions are fair.

Responsible Creators

Regardless of career path, most students will use AI tools in some professional capacity. They need to understand when human judgment should override AI, recognize limitations and harms, and apply ethical frameworks to make decisions.

Informed Participants in Society

Students need to consider the real benefits and risks f AI in their communities, when and how to leverage outputs from AI tools in daily tasks, and what the future might look like as AI systems continue to permeate society. They need to understand how bias in training data creates biased outcomes, why AI makes mistakes, who bears responsibility, and the environmental costs of AI systems.

Without knowledge of how AI works, students cannot fully and meaningfully participate in evaluating consequential technologies nor shape their future iterations.

When students learn that AI systems are created by humans, trained on human-collected data, and shaped by human decisions, their understanding of computing changes:

- Mistakes are recognized as predictable consequences of gaps in training data
- Bias is understood as the result of human choices about data and design
- Students see themselves as potential shapers of AI's future, rather than passive users

AI in the CSTA PK–12 Standards

Because AI technologies are built upon foundational computer science principles, the writing team chose to integrate AI learning throughout the Standards rather than define AI as a separate concept. AI content in the standards serves two primary purposes: (1) to help students understand how AI works and (2) to encourage students to grapple with real impacts and ethical issues related to AI technologies, their creation, and their deployment.

Specificall , AI content is represented **across the standards** in the following ways:

 **Algorithms & Design**	Rule-based vs. data-driven approaches, machine learning, and societal impacts (e.g., bias)
 **Programming**	Reading and evaluating AI-generated code
 **Data & Analysis**	Data fluency and societal impacts (e.g., la ge-scale data collection)
 **Systems & Security**	Security implications and environmental impacts
 **Computing & Society**	History of computing and AI, emerging technologies, the relationship between humans and AI, and how people decide when and how to use these technologies responsibly

To clarify where AI-related learning appears across the Standards, relevant standards are tagged as "AI-related" when they meet one or more of the following criteria:

- The standard or the boundary statement explicitly references AI.
- The standard closely aligns with a learning outcome from the *AI Priorities for All K–12 Students* project (CSTA & AI4K12, 2025).
- The standard includes content that is foundational to AI literacy.

The writing team also defined AI as a high school specialty a ea to inform the development of pathways for students who wish to continue learning about AI after they complete a foundational CS learning experience.

PK–12 students deserve to understand the systems shaping their futures. Learning experiences aligned with the AI content in the 2026 CSTA PK–12 Standards help students understand how AI systems work and prepare them to examine their broader impacts on society.

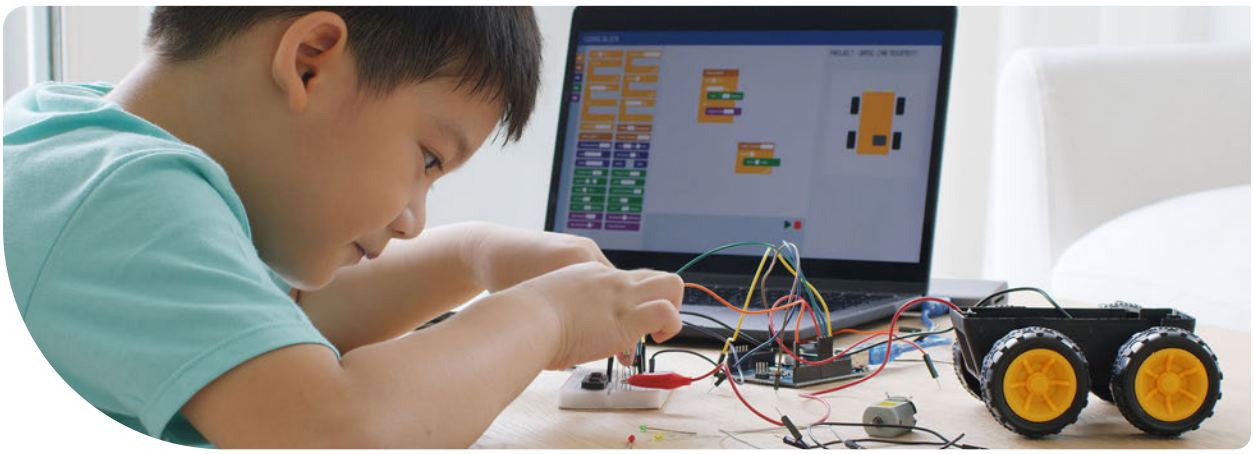
A Sociotechnical Approach to CS Education

Computing is woven into every aspect of our lives, from the tools we use to communicate to the technologies that shape our opportunities. Students need more than technical skills. They need to understand how computing and society shape one another. This understanding lies at the heart of the revised CSTA PK–12 Computer Science Standards, which aim to **prepare students not only to use and build technology, but also to question, imagine, and influence its role in the world.**

Moving Beyond Techno-Myths

Technology is often described as neutral, inevitable, or outside of human control. These *techno-myths* frame computing systems as if they evolve independently of human decisions. In reality, every technology reflects decisions made by designers, companies, and communities, and carries the values and power of those choices.

Computer science education can help students replace techno-myths with sociotechnical understanding—the recognition that technology and society are inseparable. Students learn that computing systems are not just technical artifacts but products of cultural, political, and economic choices. This prepares them to ask deeper questions: *Who benefits from this technology? Whose needs are ignored? What values are embedded in its design?*



Understanding and Addressing Computing's Harms

Discussions of ethics in computing often focus on isolated examples of bad actors or unintended consequences. A sociotechnical approach encourages students to see patterns and systems—how inequities, biases, or environmental impacts arise from both the technology itself and the contexts in which it is built, used, and discarded.

Students explore real-world issues such as biased algorithms in hiring, surveillance in schools, or environmental effects of data centers. They examine **types of harm** (e.g., discrimination or misinformation), **mechanisms** (e.g., how biased data or business incentives drive those harms), and **mitigation strategies** (e.g., responsible design, policy, and advocacy). In doing so, students develop agency and recognize that harms are not inevitable, and that human choices can shape better outcomes.

Practicing Ethical and Responsible Computing

Understanding is only half the goal; the other half is practice. Ethical, responsible, and critical computing becomes meaningful when students *do* it. This includes engaging in a range of practices:

- **Ethical design:** creating technologies that reflect values such as fairness, privac , and accessibility
- **Critical inquiry:** investigating how technologies affect people and the environment, and questioning whether certain technologies should exist at all
- **Hopeful reimagining:** envisioning technological futures that are more just and considering how they might be achieved
- **Advocacy:** recognizing when to question, challenge, or avoid technologies that cause harm
- **Responsible use:** making thoughtful decisions about personal and collective use of technology

These practices move computing education beyond coding for its own sake, helping students connect computing to real-world issues they care about.

Ethics as Dialogue, Not Doctrine

There is no single "right" answer to what is ethical in computing. Instead of prescribing moral rules, educators can help students navigate competing perspectives and values. Some classrooms might focus on identifying harms and responsibilities; others may examine the values built into technologies or explore what happens when you look at issues through multiple, sometimes competing, ethical frameworks. Across these approaches, students learn to reason, debate, and deliberate. They recognize that ethical computing is a process of ongoing reflection and dialogue. The examples below illustr te a few ways these ideas might appear in classroom practice.

Examples of Implementing Ethics in PK-12 Classrooms

Elementary (Grades PK–5)	Middle School (Grades 6–8)	High School (Grades 9–12)
• **Exploring accessibility:** Students examine digital tools used in their classroom and discuss for whom they are designed, who might have difficulty using them, and h they could be redesigned to work for more people. • **Understanding personal data:** Students investigate what types of personal data apps collect and discuss when and why people might choose to share or protect their information.	• **Investigating AI:** Students explore examples of bias in facial recognition or other AI systems, experimenting with tools that help them understand how these systems work and who they work for. • **Questioning techno-myths:** Students examine common beliefs such as "technology is neutral" or "technology is inevitable" and analyze how human decisions shape technological systems.	• **Designing with values:** Students begin software or app development projects by identifying values such as fairness, privacy, accessibility, and safety and consider how those values shape design decisions. • **Algorithmic audits:** Students systematically test AI systems to identify patterns of bias or unexpected outcomes.

Why This Matters

Reimagining computer science education through this sociotechnical lens makes CS more rigorous, relevant, engaging, and empowering. It enables students to see computing not as a distant or predetermined field, but as something they and their ommunities have the power to shape. They learn that coding and critical thinking go hand in hand, and that the most powerful computing education is one that connects knowledge to responsibility, creativity, and community.

This approach invites educators across disciplines to help prepare students to participate fully in our world powered by computing—as designers, users, and changemakers. It helps students see that *although AI may dominate headlines, human choices and collective responsibility shape the story.*

Navigating the Standards

Foundational Standards

The Foundational PK–12 Computer Science Standards are designed to provide coherent learning progressions across the PK–12 continuum. They begin with a combined pre-kindergarten and kindergarten (PK/K) band, followed by grade-level expectations for grades 1–5. Middle school is organized as a single grade band, leading into a final band th t represents foundational high school learning.

The Foundational Standards are organized around three components: concepts, practices, and dispositions. **Concepts** define the *knowledge and content* students learn and serve as the primary organizational structure. The five oncepts are: (1) Algorithms & Design, (2) Programming, (3) Data & Analysis, (4) Systems & Security, and (5) Computing & Society. Artificial intelligen e was identified as a priority during the standa ds revision process. AI-related content is distributed across the five oncepts instead of being a discrete concept.

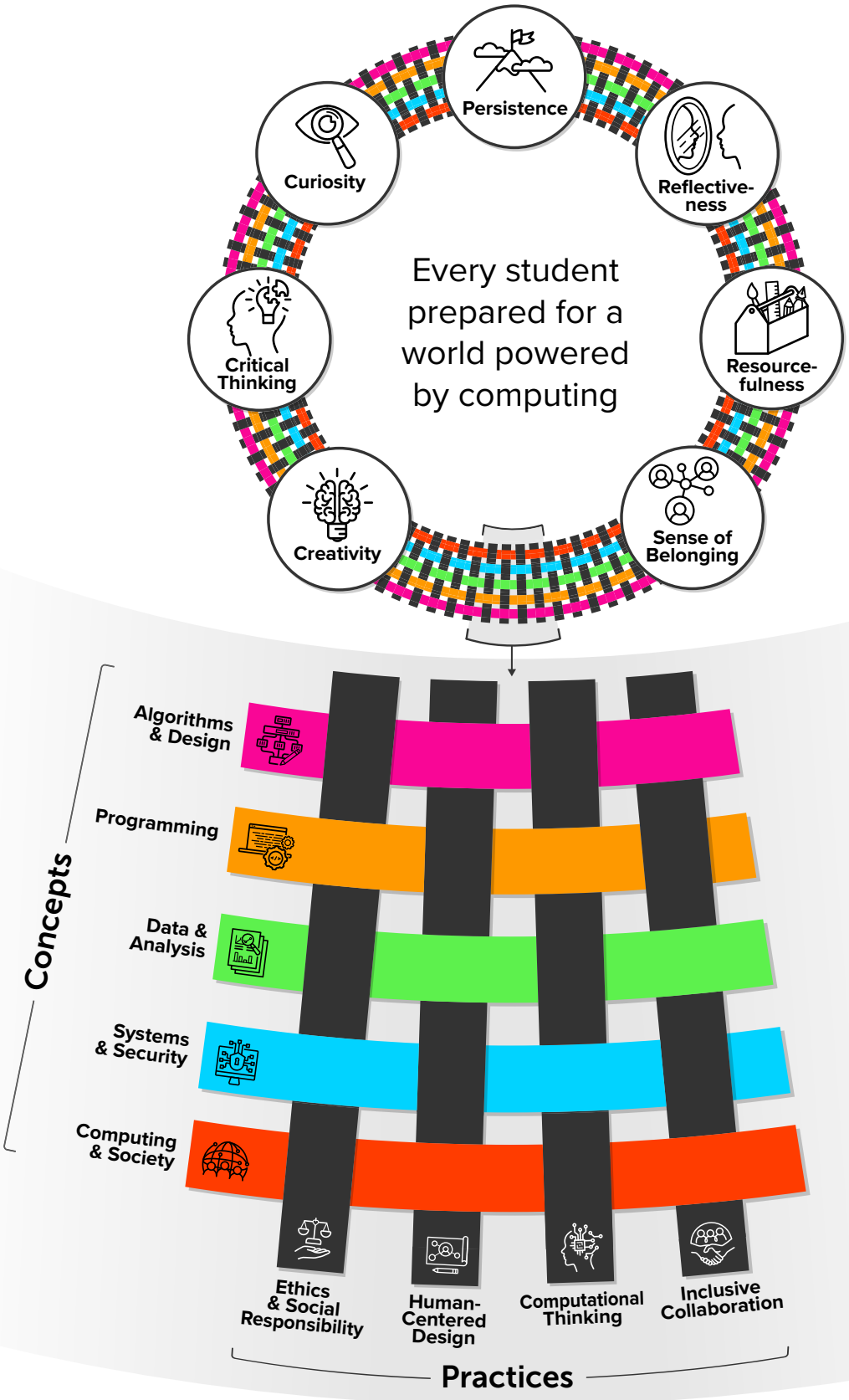
Practices define th *skills and behaviors* students develop as they learn CS. They cut across all concepts and describe *how* students engage in learning (i.e., what students *do*). Twelve practices are organized into four categories: (1) Ethics & Social Responsibility, (2) Inclusive Collaboration, (3) Computational Thinking, and (4) Human-Centered Design.

Dispositions define the *habits of mind* students develop through learning computer science. They shape how students approach learning, problem solving, and collaboration in computer science. Prioritized dispositions include creativity, critical thinking, curiosity, persistence, reflectiveness, esourcefulness, and sense of belonging.

Concepts	Practices	Dispositions
The **content** (what students **know**)	The **skills and behaviors** (what students **do**)	The **habits of mind** (**how** students approach learning)
• Algorithms & Design • Programming • Data & Analysis • Systems & Security • Computing & Society	• Ethics & Social Responsibility • Inclusive Collaboration • Computational Thinking • Human-Centered Design	• Creativity • Critical thinking • Curiosity • Persistence • Reflectiveness • Resourcefulness • Sense of belonging

Together, these components define the oundational computer science learning experience for all students. This structure builds on prior research and community work, including the *Reimagining CS Pathways* project (CSTA et al., 2024), which established a community definition f foundational CS learning and pathways beyond it. The following graphic illustrates the relationship among concepts, practices, and dispositions.

Figure 4. Foundational PK–12 CS Education



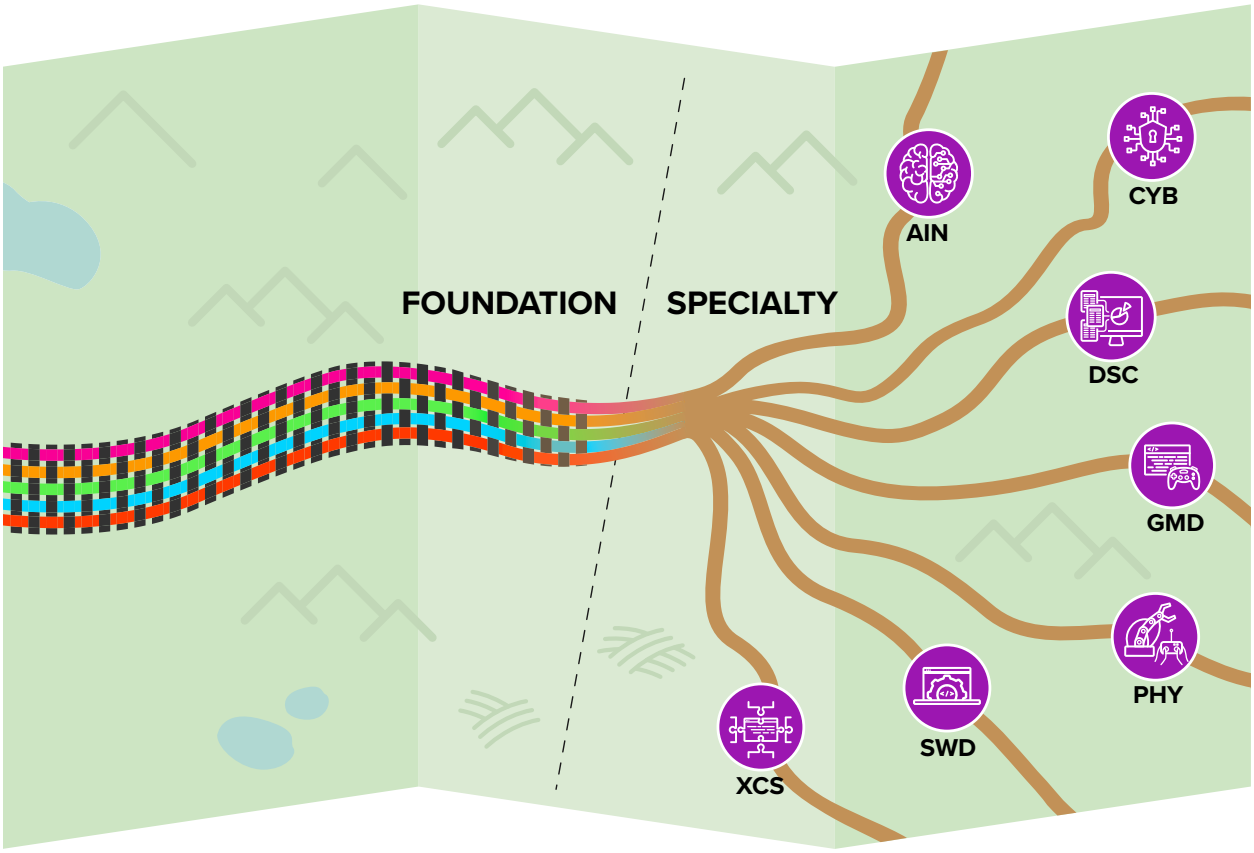
After establishing a strong foundation in computer science, students may pursue specialized learning. To support this progression, Specialty Standards articulate learning objectives for high school students who choose to deepen their knowledge in specific omputing domains. These standards extend foundational learning and support postsecondary readiness across pathways to further education, careers, and military service.

Specialty Standards

Specialty Standards extend foundational PK–12 computer science learning by defining **advanced, domain-specific learning** in key areas of computing. These standards are organized into two tiers (Specialty I and Specialty II) across six high school specialty areas identified th ough the *Reimagining CS Pathways* project:

- Artificial Intelligen e (AIN)
- Cybersecurity (CYB)
- Data Science (DSC)
- Game Development (GMD)
- Physical Computing (PHY)
- Software Development (SWD)

Figure 5. Specialty Areas in CS Education



Specialty I standards introduce the knowledge and skills essential within a chosen area, serving as a student's first ocused learning experience in that domain. **Specialty II** standards describe more advanced study, preparing students for postsecondary coursework, industry certific tions, or continued specializations.

Additionally, **X + CS Standards** (XCS) support interdisciplinary learning by integrating foundational high school CS content into other subject areas, such as journalism, biology, and the arts.

Clarifying the Standards

In addition to each standard and its identifie , a set of clarifying components provides guidance on what is included and how the standard may be interpreted and implemented. Brief descriptions of clarifying components are found in the table below. Boundary statements, practice alignment, and disposition alignment are included in the final print version f standards. Additional components are accessible through the interactive web display found at https://csteachers.org/pk12standards/view.

Component	Description
Boundary Statement*	Elaborates on the standard by clarifying key nuances, including what is in and out of scope. Ensures consistent interpretation and appropriate grade-level expectations.
Practice Alignment*	Identifies one or two omputer science practices associated with the standard. Highlights how students engage with the content.
Disposition Alignment*	Indicates one or two dispositions (habits of mind) that are developed through instruction aligned to the standard.
AI-related	Indicates whether the standard includes explicit AI-related content or supports AI-focused learning (i.e., when the standard directly addresses AI concepts, systems, or their impacts—not simply when AI could be used as a tool during implementation).
Progressions	Shows how the standard connects to prior and future learning across grade levels, clarifying what students are expected to know prior to the standard and what they are building toward.
Implementation Examples	Provides illustrative examples of how the standard might be addressed in classroom practice, including unplugged and computer-based approaches.
Interdisciplinary Connections	Highlights ways the standard can be applied in other subject areas, with an emphasis on science, mathematics, and English language arts (ELA).
Resources	Includes vetted instructional resources to support teaching and assessment.
Academic Vocabulary	Identifies ey terms referenced in the standard.

**Included in full in the print version of the Standards.*

Example of a Standard with Clarifying Components

The following figu e provides an example of a standard with its associated clarifying components. Although only a subset of these components appears in the print version, the full set is available in the interactive web display.

Figure 6. Standard MS-SYS-SE-32 with Clarifying Components

MS-SYS-SE-32

Explain the effects of not using the CIA triad* when working with data.

BOUNDARIES, PRACTICES, & DISPOSITIONS

Students should recognize and explain what happens when each element of the CIA (confidentialit , integrity, and availability) triad is not maintained. When confidentiality is ompromised, sensitive data is exposed (e.g., leaked passwords, shared personal information). When integrity is compromised, data becomes corrupted or altered (e.g., incorrect grades in a school database). When availability is compromised, systems become inaccessible (e.g., a denial-of-service attack blocking a website). Students should apply the CIA triad to age-appropriate, relatable contexts (e.g., school accounts, social media, online gaming), demonstrating understanding of how failures connect to real-world impacts.

Students are not expected to conduct professional-level security audits or develop deep cryptography knowledge.

PRACTICES:

Inclusive Collaboration
3. Communicate effectively about computing.

Human-Centered Design
12. Design computing technologies that empower and inform users.

DISPOSITIONS:

Critical Thinking

Reflectiveness

IMPLEMENTATION EXAMPLES

Unplugged: In small groups students receive scenario cards (e.g., password leaks, altered grades, denial-of-service) and identify which CIA element failed, describe the consequences for users and systems, and propose age-appropriate prevention strategies. Groups present findings and ompare cases across contexts (e.g., school accounts, social media, online gaming). They respond to multiple teacher prompts (e.g., "Which CIA element failed?" "What were the consequences?" "How could this be prevented?")

Teacher choice: Students work individually or in pairs to identify the specific failu es in teacher-run simulations that intentionally demonstrate failures (e.g., overly broad sharing settings to show confidentiality loss, silent modifi tion of a document to show integrity loss, temporary account lockouts to show availability loss). They explain the effect of the failure on tasks and users and recommend safeguards. In a wrap-up discussion, students talk about how those safeguards would change behavior or system settings.

INTERDISCIPLINARY CONNECTIONS

MATH

6.RP.A.3: Use ratio and rate reasoning to solve real-world and mathematical problems, e.g., by reasoning about tables of equivalent ratios, tape diagrams, double number line diagrams, or equations.

Students calculate the percentage of students in class who reuse passwords across accounts and analyze risks.

7.SP.C.7: Develop a probability model and use it to find probabilities of events. Compare probabilities from a model to observed frequencies; if the agreement is not good, explain possible sources of the discrepancy.

Students model probability of data breaches under weak versus strong password policies.

ELA

SL.7.1: Engage effectively in a range of collaborative discussions (one-on-one, in groups, and teacher-led) with diverse partners on grade 7 topics, texts, and issues, building on others' ideas and expressing their own clearly.

Students role-play as security analysts discussing which CIA failures led to different case study incidents.

W.8.1: Write arguments to support claims with clear reasons and relevant evidence.

Students write a persuasive essay arguing which element of the CIA triad is most critical to protect in schools.

SCIENCE

MS-ETS1-2: Evaluate competing design solutions using a systematic process to determine how well they meet the criteria and constraints of the problem.

Students evaluate different school-level security measures (e.g., two-factor authentication, password resets, backups) to determine which best prevents CIA failures.

MS-ETS1-3: Analyze data from tests to determine similarities and differences among several design solutions to identify the best characteristics of each that can be combined into a new solution to better meet the criteria for success.

Students review case studies of school Wi-Fi outages, identify availability issues, and propose system improvements.

OTHER

Social Studies - D4.6.6-8: Draw on multiple disciplinary lenses to analyze how a specific problem can manifest itself at local, regional, and global levels over time, identifying its characteristics and causes, and the challenges and opportunities faced by those trying to address the problem.

Students debate whether schools should prioritize confidentiality (p otecting student data) over availability (keeping systems always online).

Arts - VA:Cr1.1.6a: Combine concepts collaboratively to generate innovative ideas for creating art.

Students collaboratively design posters or infographics that illustrate the consequences of neglecting confidentialit , integrity, or availability in real-world scenarios.

PROGRESSIONS

Grade 5	Middle School	High School
E5-SYS-SE-13: Describe the concepts of the CIA triad and how each component is important in protecting information.	**MS-SYS-SE-32:** **Explain the effects of not using the CIA triad when working with data.**	HS-SYS-SE-31: Identify different types of cybersecurity and physical security measures and the trade-offs or users, data, and devices.
	MS-SYS-SE-33: Evaluate common types of cyber attacks and preventions.	HS-SYS-SE-32: Classify the causes and impacts of security breaches and social engineering attacks for individuals, industries, communities, and governments.
		HS-SYS-SE-33: Formulate a solution to a security flaw in a given system.

*Academic Vocabulary (in the interactive web display, underlined terms are hyperlinked to a popup modal):
CIA Triad: A core information security model based on three principles: keeping information private (Confidentiality), ensuring it is trustworthy, accurate and unchanged (Integrity), and making sure authorized users can access it when needed (Availability).

Key Terms and Definitions

Key terms used throughout the Standards are defined in an online glossary available t https://csteachers.org/glossary.

Definitions a e also embedded directly within the interactive standards web display. Glossary terms appear as clickable text that expands a modal window to display their meaning in context.

Interdisciplinary Standards

Each foundational CSTA standard includes specific inte disciplinary connections to established standards from other subject areas, including:

Arts: National Core Arts Standards
(National Coalition for Core Arts Standards, n.d.)

Career Exploration: American School Counselor Association Student Standards (American School Counselor Association, 2021), National Career Development Guidelines (National Career Development Association, 2024)

Early Childhood Education: Head Start Early Learning Outcomes Framework (Office of Head Start, 2015)

ELA: Common Core State Standards for English Language Arts
(NGA Center & CCSSO, 2010a)

Health and Physical Education: National Health Education Standards (SHAPE America, 2024a), National Physical Education Standards (SHAPE America, 2024b)

Library: National School Library Standards for Learners
(American Association of School Librarians, 2018)

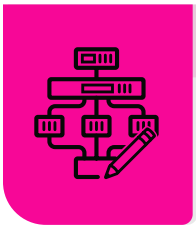
Math: Common Core State Standards for Mathematics
(NGA Center & CCSSO, 2010b)

Science: Next Generation Science Standards
(NGSS State Leads, 2013)

Social Studies: National Curriculum Standards for Social Studies
(National Council for the Social Studies, 2010)

Concepts

Concepts define the ore content areas of computer science and serve as the primary organizational structure of the Standards. Together, they represent the foundational knowledge students develop as they progress across grade levels.



Algorithms & Design

Algorithms & Design focuses on how problems can be represented and solved through precise, step-by-step processes. Students design, analyze, and refine algorithms th t transform inputs into desired outputs using both rule-based approaches and data-driven approaches informed by patterns in data. They consider the efficien , correctness, and limitations of algorithms, as well as how design choices influen e outcomes, developing the ability to create and evaluate computing technologies in real-world contexts.

The Algorithms & Design standards and the Programming standards are complementary and should be considered in tandem. Algorithms & Design standards focus more on program planning and evaluation, whereas Programming standards focus more on program implementation.

Subconcept	Overview
Algorithmic Problem Solving	Designing algorithms, or step-by-step solutions to tasks, is an essential practice in computer science. In early grades, students identify and carry out algorithms in everyday activities and begin creating their own. As they progress, students design and model algorithms, verify correctness, and consider different approaches to solving problems. By high school, students design and optimize algorithms, evaluate them for efficiency and orrectness, and distinguish between deterministic and probabilistic approaches. Students also use AI tools to support problem solving and evaluate the accuracy and impacts of algorithmic outputs.
Machine Learning	Machine learning is a subfield f artificial intelligen e in which systems identify patterns in data to make predictions or decisions without being explicitly programmed to do so. This subconcept includes content that is key for understanding foundational components of artificial intelligen e. In early grades, students recognize patterns used by people and machines for decision-making. As they advance, students explore how machine learning models evolve with new training data, train models for classific tion or prediction, and analyze the relationship between training data properties and model output. By high school, students justify the selection of machine learning algorithms, evaluate training data for quality and bias, and develop models for specific tasks using app opriate data and tools.
Impacts of Algorithms & Design	Algorithms can have positive and negative effects on people and society. It is important to evaluate not only how well an algorithm works, but also whom it benefits, whom it may unintentionally harm, and wh . In early grades, students begin by exploring how algorithms can lead to different results for themselves and others. As students progress, they learn to identify possible consequences of algorithmic decisions and they apply human-centered design principles to develop and refine computational algorithms. In later grades, students critically analyze the societal impacts of algorithms, including issues of fairness, equity, accessibility, and bias, and consider how algorithmic systems can shape real-world experiences.

Programming

Programming controls all computing systems, empowering people to communicate with the world in new ways and solve compelling problems. Developing meaningful and efficient ograms involves selecting and organizing information, breaking problems into manageable parts, recombining existing solutions, and evaluating alternative approaches.

Programming is not limited to text-based code. Especially in early grades, students may use block-based or tangible programming systems to create and explore programs.

The Algorithms & Design standards and the Programming standards are complementary and should be considered in tandem. Algorithms & Design standards focus more on program planning and evaluation, whereas Programming standards focus more on program implementation.

Subconcept	Overview
Program Development	Program development focuses on creating, improving, and collaborating on programs to solve problems or express ideas. In early grades, students translate algorithms into code and build simple programs. As they progress, students refine programs through feedback, compare solutions, and modify existing code to create new programs. In middle school, students structure programs, use documentation, and collaborate with others. By high school, students develop modular programs using advanced tools and collaborate using defined workflow
Variables & Data Storage	Variables and data storage describe how programs represent, store, and manipulate information. In early grades, students identify and label data in everyday contexts, recognize values that can change over time, and identify and trace data in programs. In middle school, students use variables of multiple data types to manage data within programs. By high school, students design programs that use appropriate data structures to store, access, and manipulate data. Note: This subconcept focuses on data used within programs, whereas the Data & Analysis concept emphasizes collecting, organizing, and analyzing data using computing.
Reading & Documenting	In a world where AI assistants can generate code based on a prompt, the ability to read and interpret code is increasingly important. In early grades, students describe how code completes tasks and explain program steps. As they progress, students document programs and explain how different parts contribute to overall behavior. In middle school, students analyze how key programming constructs work and examine AI-generated code. By high school, students analyze more complex code and evaluate AI-generated code for accuracy, reliability, and alignment with program requirements.
Testing & Refining	Testing and refining p ograms involves identifying errors and improving programs to meet their intended purpose. In early grades, students identify and debug errors in simple programs. As they progress, students use systematic strategies to test and refine their work. In middle school, students in orporate testing, documentation, and user feedback to improve programs. By high school, students evaluate how well programs meet design goals and refine them based on testing esults, user feedback, and intended impact.

Data & Analysis

Data & Analysis focuses on how data is collected, represented, analyzed, and used to understand the world and make decisions. Computing systems enable people to gather, store, and process large amounts of data, but meaningful use of data depends on the questions people ask, the ways data is organized and visualized, and how results are interpreted. As the amount of digital data generated continues to grow rapidly, the ability to work effectively with data becomes increasingly important.

Across grade levels, students learn to pose questions, work with different types of data, and use computational tools to analyze and represent information. They identify patterns, draw conclusions, and examine how data is used in real-world contexts, including issues of accuracy, bias, privacy, and impact.

Subconcept	Overview
Data Collection & Preparation	Data is generated and collected by people, often using computing technologies such as sensors and other automated systems. Metadata is "data about data." Metadata provides context about data, including its origin, structure, and purpose. In early grades, students collect and use different types of data to answer questions. As they progress, students evaluate data for accuracy and organize it using digital tools. In middle school, students consider data quality and use tools to sort, filte , and summarize data. By high school, students generate, clean, and document datasets and evaluate methods for ensuring data consistency and validity.
Data Investigation	A data investigation is a multistep process. When conducting data investigations, students pose data questions, use computational tools to collect and analyze data, create data visualizations, generate insights, and tell the story of their data. In early grades, students focus on asking simple questions that can be answered with small datasets. As students progress, they work with larger datasets, asking and answering more sophisticated questions that consider variability and relationships between multiple variables.
Impacts of Data Science	The use of data influen es decision-making and affects individuals, communities, and society. Students examine the benefits, risks, and ethical onsiderations around data use. In early grades, students discuss how using data can help them make more informed decisions in their daily lives. As students progress, they explore issues related to bias in data, data privacy, artificial intelligen e and machine learning, and large-scale societal and environmental impacts of data science applications.

Systems & Security

Systems & Security includes the broad categories of hardware and software, networks, and cybersecurity. The physical components (hardware) and instructions (software) that make up a computing system communicate and process information in digital form. Networks connect computing devices to share information and resources. Greater connectivity in the computing world has also led to an increased need for security to protect the information being transmitted.

Subconcept	Overview
Hardware & Software	Computing systems use hardware and software to communicate and process information in digital form. In early grades, students learn how systems use both hardware and software to represent and process information. As they progress, students gain a deeper understanding of the interactions between hardware and software at multiple levels within computing systems.
Security	Transmitting information securely across networks requires appropriate protection. In early grades, students learn how to protect their personal information. As they progress, students learn about information transmission across devices, network design, and how to protect networks from different types of threats.
Networks	Computing devices communicate with one another across networks to share information. In early grades, students learn that computers connect them to other people, places, and things around the world. As they progress, students gain a deeper understanding of how information is sent and received across different types of networks.
Impacts of Computing Systems	Humans created computing systems to accomplish tasks and solve problems. While there have been benefits, the e have also been harms and the creation of new problems. In early grades, students examine the impacts of computing systems on individuals. As they progress, students learn about the impacts of computing systems on global society.

Computing & Society

Computing shapes and is shaped by individuals, communities, and cultures from around the world. Computing transforms daily life, economies, governments, and global systems in ways that offer both great promise and significant challenges. The impacts f computing are complex, encompassing advances that improve lives alongside harms that deepen inequities and raise ethical concerns. Students learn to critically and responsibly navigate these social implications, considering issues of equity, access, and accountability, while also exploring computing's potential to promote social good. By examining the evolving relationship among computing, culture, and society, students are empowered to contribute thoughtfully and responsibly to a digital future.

Subconcept	Overview
History of Computing	The history of computing reflects ontributions from many societies and knowledge traditions. In early grades, students first identify how omputing is used in daily life. Then they focus on how technologies have evolved over time in response to social, scientific, and e onomic needs. In middle grades, students evaluate how historical challenges led to computing innovations and compare the roles played by individuals, communities, organizations, and governments in advancing these technologies. Students also explore the societal impacts of computing innovations. In later grades, students delve into the main eras of computing history and understand policy and legislation related to computing technologies. They also analyze the historic impacts of technologies, giving consideration to the factors that contributed to disparities across communities.
Emerging Technologies	Computing is a rapidly developing discipline. While other concepts cover what students should know about the current field f CS, this subconcept focuses on recently developed technologies that have the potential to significantly impact societ . In early grades, students learn how computing technology aids their daily life. They evaluate choices and consequences related to emerging technologies and identify problems that these advances could address. In middle grades, students evaluate how emerging technologies impact user experiences while attending to ethical design principles. They explain how emerging technologies can inspire innovation and help people accomplish tasks in new ways. In later grades, students understand the core computational principles behind emerging technologies and compare the ethical considerations of emerging technologies, using various frameworks.
Humans & Computing	Humans and computing technologies are intricately connected. Students learn that people are the fundamental creators of these technologies, and they differentiate between tasks best suited for humans versus those for computing. They investigate how humans leverage computing to solve problems and examine the motivations behind designing and building these systems. A critical focus is distinguishing between human and computational learning processes, enabling students to decide when and where computing technologies, including AI, are appropriate and helpful. As they progress, students analyze the trade-offs f using computing technologies and evaluate how human choices in designing, deploying, and regulating computing technologies influen e their risks, benefits, and long-term societal impacts
Career Exploration	Computing is foundational to nearly every career field. Understanding the ole of computing in the workplace prepares students to make informed choices about their futures. In early grades, students recognize how digital tools and technologies support everyday work across professions. In middle grades, students explore how computational thinking drives innovation across industries and examine the ethical challenges that professionals may encounter. In later grades, students connect computing to their personal interests and career goals, investigate computing-related pathways, and analyze how advancements in technology foster new opportunities for growth across diverse fields

Practices

Practices describe the skills, behaviors, and ways of thinking that students with a strong foundation in computer science use to fully engage in a world powered by computing. Organized into four categories, practices reflect ta get outcomes for students by the end of high school and, as such, are recurring across grade levels and are developed through repeated exposure. Each standard is intended to reflect ontent from a given concept, as well as one or more practices.

Ethics & Social Responsibility (ESR)

The Ethics & Social Responsibility practices develop habits that help students become responsible creators of technology. These practices are based on the Association for Computing Machinery's list of general ethical principles (ACM, 2018).

1. **Use computing for positive social impact.**
    a. Imagine and create computing technologies that solve problems, strengthen communities, and improve quality of life.
    b. Evaluate whether potential benefits f computing technologies outweigh possible harms. Ensure that harms are not concentrated on specific g oups and avoid serious environmental harm.
    c. Explain design trade-offs in omputing technologies, including the values these decisions reflect and wh t was prioritized or sacrifi ed.
2. **Respect others' rights and dignity when creating computing technologies.**
    a. Respect other creators of computing technologies. Only use others' work with permission and give appropriate attribution.
    b. Respect users' privacy and protect their data. Give users choices about how their information is collected and used. Only collect the minimum amount of information necessary. Avoid practices that manipulate or undermine users.

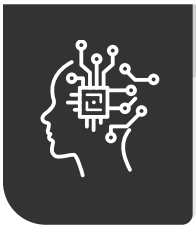
Inclusive Collaboration (IC)

Inclusive Collaboration practices help students develop productive collaborations with diverse groups of people. These practices address communication skills, project management skills, and personal conduct when working with others. They were synthesized from the Standards for Technological and Engineering Literacy (ITEEA, 2020), the Social Justice Standards (Learning for Justice, n.d.), the Framework for 21st Century Learning (Partnership for 21st Century Skills, 2009), and the K–12 CS Framework (2016).

3. **Communicate effectively about computing.**
    a. Share technical ideas and explain computing concepts clearly to different audiences.
    b. Give and actively listen to others' input and constructive feedback. Consider diverse perspectives and multiple solutions to technical challenges.

4. **Manage computing projects.**
    a. Establish shared goals, break the work into discrete tasks, and set development milestones.
    b. Document code and development processes.

5. **Act responsibly in computing collaborations.**
    a. Cultivate working relationships with individuals possessing diverse perspectives, skills, and personalities.
    b. Participate reliably in computing teamwork, share responsibility for project outcomes, and meet development deadlines.
    c. Reflect on ontributions to computing projects, including technical decisions and team interactions, to improve technical and collaboration skills.



Computational Thinking (CT)

Computational thinking is a way of thinking about problems and formulating problems and solutions so that an information-processing agent (e.g., a computer) can help to solve them. Computational Thinking practices connect students' CS learning with the engineering design process, in which students identify and define omputational problems, develop computing technologies that serve as solutions, and iteratively test, refine, and optimi e those solutions. These practices are largely based on the original Computational Thinking practices from the K–12 CS Framework (2016), but also incorporate ideas from the Next Generation Science Standards (NGSS Lead States, 2013) and *Computational Thinking 2.0* (Tedre et al., 2021).

6. **Define computational problems.**
    a. Identify real-world problems that can be solved computationally using rule-based approaches, data-driven approaches (e.g., machine learning), or a combination of these approaches.
    b. Clearly state criteria for success and identify constraints.
    c. Decompose complex problems into manageable subproblems.

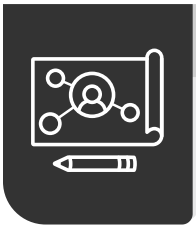
7. **Develop and use abstractions.**
    a. Extract common features and patterns from data, processes, or phenomena to create general methods and algorithms.
    b. Create reusable modules and procedures that can apply to multiple situations to reduce complexity.
    c. Model phenomena and develop simulations, using rule-based and data-driven approaches, to understand and evaluate potential outcomes.

8. **Create computing technologies.**
    a. Generate and evaluate multiple approaches to solving a problem. Determine which solution best meets the defined criteria given the onstraints.
    b. Plan the development of computational solutions.
    c. Implement solutions by developing or modifying technologies through traditional programming, model training, or a combination of these approaches.

9. Test and refine computing technologies.

a. Test, debug, and troubleshoot computing technologies systematically using appropriate methods, such as generating test cases for rule-based programs and accuracy evaluation for machine learning models.

b. Iteratively refine and optimi e technologies to meet criteria for success.



Human-Centered Design (HCD)

Human-centered design is critical to the responsible development of computing technologies. It draws on principles of human–computer interaction to understand user needs and iteratively design systems that support and empower people. The following practices are drawn from a variety of well-known sources on human-centered design, including the National Institute of Standards and Technology (NIST, 2021), the Interaction Design Foundation (IDF, n.d.), and the UX Design Institute (Vinney, 2023), as well as the K–12 CS Framework (2016).

10. Understand and involve diverse users in design decisions.

a. Learn about different people's experiences with computing technologies, including those with different abilities, backgrounds, and needs.

b. Gather input and feedback from diverse users throughout the design and development process to help create positive user experiences.

11. Use iterative design processes.

a. Start with simple prototypes and continuously test and refine omputing technologies to ensure usability and accessibility.

12. Design computing technologies that empower and inform users.

a. Respect users' autonomy. Be transparent about how computing technologies make decisions that affect users. Give users control over how they interact with technologies rather than leveraging human limitations to serve creators' interests over users' interests.

b. Consider the benefits and harms f human-like behaviors in computing technologies (e.g., conversational AI) and how they influen e user perceptions and actions.

Dispositions

Dispositions are the **habits of mind, attitudes, and approaches** that shape how students engage with computer science. They influen e how students think, persist, collaborate, and reflect beyond wh t they can code or recall. In CS, dispositions guide how students navigate challenges, debug with purpose, and build confiden e in problem solving.

Fostering these dispositions helps students develop not only technical skills, but also the capacity to learn independently, stay motivated, and persist through complexity.

Key Dispositions in CS Education

In *Reimagining CS Pathways* (CSTA et al., 2024), the CS education community identified **seven key dispositions** essential to equitable and enduring CS learning:

	**Creativity**	Generating original and meaningful computing ideas and projects.
	**Critical thinking**	Using reasoning and evidence to analyze and refine solutions
	**Curiosity**	Asking questions and exploring beyond assigned work.
	**Persistence**	Continuing effort despite frustration or setbacks.
	**Reflectiveness**	Connecting past experiences to inform future learning.
	**Resourcefulness**	Seeking and using tools, people, and information to solve problems.
	**Sense of belonging**	Feeling included, respected, and recognized in the CS community.

These dispositions align closely with *self-regulated learning*, as students plan, monitor, and evaluate their growth as learners of computer science.

Why These Dispositions Matter

Dispositions shape how students experience computer science—not just what they learn, but how they *see themselves as learners and creators*. They:

- **Advance equity and inclusion:** Belonging and creativity help all students see themselves reflected in computing.
- **Deepen understanding:** Critical thinking and reflectiveness connect conceptual learning to practice.
- **Build resilience:** Persistence and resourcefulness help students navigate challenges and turn setbacks into progress.
- **Sustain motivation:** Curiosity keeps students exploring beyond assigned tasks and supports ongoing learning.

Dispositions are the foundation of lasting CS learning. They connect technical ability to personal growth, ensuring that every student can engage meaningfully with computing. Teachers foster dispositions through intentional instructional design decisions and a consistent instructional approach that encourages their development. When classrooms intentionally emphasize these core dispositions, students develop both the confiden e and competence to thrive as problem solvers, innovators, and future leaders in CS.

Foundational Standards for PK–12

Naming Conventions for Foundational Standards

Each of the identifiers or Foundational Standards follows this naming convention:

Grade band		Concept		Subconcept		Number
XX	-	YYY	-	ZZ	-	##

There are standards for each individual elementary grade level (ages 5–11). All identifiers for elementary standards begin with "E," followed by a letter (K to indicate PK/Kindergarten standards) or number (1–5) to indicate the grade level. Standards for grades 6–8 (ages 11–14) are banded together as middle school standards. Identifiers or middle school standards begin with "MS." Standards for grades 9–12 (ages 14–18) are banded together as high school standards. Identifiers or high school standards begin with "HS."

The next two sets of characters indicate the concept and subconcept for each standard. The following table shows the abbreviations for each concept and subconcept. The last two digits of each standard reflect the standa d number. The Foundational Standards begin with 01 for each grade or grade band. The standards are numbered continuously across all concepts within each grade or grade band.

	Abb	Concept	Abb	Subconcepts
	**ALG**	Algorithms & Design	PS ML IM	Algorithmic Problem Solving Machine Learning Impacts of Algorithms & Design
	**PRO**	Programming	PD VD RD TR	Program Development Variables & Data Storage Reading & Documenting Testing & Refinin
	**DAT**	Data & Analysis	DC DI IM	Data Collection & Preparation Data Investigation Impacts of Data Science
	**SYS**	Systems & Security	HW SE NT IM	Hardware & Software Security Networks Impacts of Computing Systems
	**SOC**	Computing & Society	HI ET HU CE	History of Computing Emerging Technologies Humans & Computing Career Exploration

Algorithms & Design

Algorithmic Problem Solving

EK-ALG-PS-01: Carry out algorithms in daily activities.

Boundary Statement: Students should demonstrate simple, step-by-step activities in the correct sequence for familiar classroom or home tasks. Following classroom routines (e.g., cleaning up or lining up) or completing personal tasks (e.g., eating snacks or washing hands) are appropriate. Teachers should explicitly teach the word algorithm in context.

Students are not expected to carry out algorithms that include conditionals or iteration. Students are not required to write algorithms or use a programming language.

Practice(s): CT7	**Disposition(s):** Critical Thinking

E1-ALG-PS-01: Decompose a problem or task into individual parts to develop an algorithm.

Boundary Statement: Students should be able to break down familiar, multistep tasks into smaller, ordered steps, focusing on sequencing and logical thinking. For example, decompose classroom routines, a game in physical education, or instrument maintenance in music into distinct, sequenced steps.

Students are not expected to work with abstract problems or create algorithms involving conditional logic (if/then). Students are not required to use a programming language.

Practice(s): CT6, CT7	**Disposition(s):** Critical Thinking

E2-ALG-PS-01: Create an algorithm that includes sequence, events, and iteration to solve a problem or express an idea.

Boundary Statement: Students should create simple algorithms using a sequence of steps that respond to events (e.g., "Ready, set, go!" or tapping on a character) and include iteration (e.g., loops) to accomplish a task or express an idea. Appropriate activities include creating a storyboard for a story to be coded later, developing an algorithm to solve a math story problem, or designing an algorithm of the steps to complete a science experiment. Students may complete this work individually or with peers.

Students are not expected to use complex or nested loops. Students are not required to create algorithms with conditional branches or variables.

Practice(s): CT6, CT8	**Disposition(s):** Creativity, Critical Thinking

E3-ALG-PS-01: Create an algorithm that includes a combination of sequence, events, iteration, and selection to solve a problem or express an idea.

Boundary Statement: Students should focus on solving problems or expressing ideas that include sequences, responses to events, iteration (e.g., loops), and selection (e.g., conditionals, if/then statements). An appropriate activity is writing instructions for a game. Students may complete this work individually or with peers.

Students are not expected to learn formal conditional logic. Students are not required to create one single algorithm that includes all control structures (i.e., sequencing, events, iteration, and selection). However, all control structures should be included throughout multiple algorithms.

Practice(s): CT6, CT8	**Disposition(s):** Critical Thinking

E4-ALG-PS-01: Create a written representation of an algorithm that includes a combination of sequence, events, iteration, and selection to solve a problem or express an idea.

Boundary Statement: Students should write step-by-step representations of an algorithm using grade-appropriate, everyday language that include sequences, responses to events, iteration (e.g., loops), and selection (e.g., conditionals, if/then statements). Outlining a Choose Your Own Adventure story where readers make decisions that alter the path is appropriate. Students may complete this work individually or with peers.

Students are not expected to create visual representations, flowcharts, or ormal pseudocode. Students are not required to create a single representation that includes every program control structure (i.e., sequencing, events, iterations, and selection). However, all control structures should be included in multiple written representations.

Practice(s): CT6, CT7	**Disposition(s):** Creativity, Critical Thinking

E5-ALG-PS-01: Create a visual representation of an algorithm that includes variables and a combination of sequence, events, iteration, and selection to solve a problem or express an idea.

Boundary Statement: Students should create visual representations (e.g., physical coding blocks, mind maps, annotated diagrams) showing algorithms that use variables and include sequence, events, iteration, and selection. Illustrating an algorithm as a drawing on index cards showing the sequence of events from start to finish on a hori ontal line with branches for selection is appropriate. Students may complete this work individually or with peers.

Students are not expected to include complex data structures (e.g., arrays, lists), procedures, or more than one level of nested iteration or selection. Students are not required to use formal flowchart notation or a single comprehensive diagram showing all control structures in a single representation. However, all control structures should be included in multiple written representations.

Practice(s): CT6, CT7	**Disposition(s):** Creativity, Critical Thinking

MS-ALG-PS-01: Design an algorithm that includes variables of multiple data types to solve a problem or express an idea.

Boundary Statement: Students should design and explain an algorithm that uses multiple variables with different data types (e.g., integers, string, Boolean) to solve a problem or express an idea. The algorithm should show how variables are created, updated, and used (e.g., tracking a score, storing a name, checking a true/false condition, responding to user input, calculating a result). Representations may be verbal, visual, physical, pseudocode, or block-based and should emphasize the logic and data changes rather than producing executable code.

Students are not expected to write executable programs, manage memory, use or implement data structures (e.g., lists/arrays, dictionaries/maps, objects), or analyze efficien .

Practice(s): CT6, CT7	**Disposition(s):** Critical Thinking, Creativity

MS-ALG-PS-02: Model a given algorithm with a flowchart or pseudocode that includes a combination of control structures and procedures.

Boundary Statement: Students should model a teacher- or peer-provided algorithm using either a flowchart or clearly structu ed pseudocode that shows how the algorithm runs. The model should include a combination of control structures (i.e., sequence, iteration, and selection) and should represent at least one procedure (a named, reusable set of steps) being used within the algorithm. Models may include variables and other supporting details only as needed to accurately represent the algorithm's logic. Teacher-provided templates or agreed-upon formats are appropriate.

Students are not expected to design or optimize the algorithm, justify efficien , refactor for readability, or apply formal software engineering practices. Students are also not expected to use professional programming syntax or convert pseudocode into executable code.

Practice(s): CT7, CT8	**Disposition(s):** Reflectiveness, ersistence

MS-ALG-PS-03: Verify the accuracy of an algorithm for given inputs.

Boundary Statement: Students should test whether a given algorithm produces expected outputs for specific inputs by tracing steps or running the algorithm. For example, students ould check whether a sorting algorithm correctly orders numbers or whether a set of instructions builds a shape as intended.

Students are not expected to perform formal proofs of correctness, analyze efficien , or optimize algorithms. Students are not required to use mathematical notation or comparison of multiple algorithms for performance.

Practice(s): CT9, HCD11	**Disposition(s):** Critical Thinking, Persistence

MS-ALG-PS-04: Justify whether a problem is best solved using procedural instructions, rule-based logic, data-driven methods, or a combination of these approaches.

Boundary Statement: Students should justify whether a problem is best solved using explicit step-by-step instructions, pattern-based rules, data-driven approaches, or a combination of these. Examples may include explaining why a procedural approach fits a fixed and predictable task (e.g., calculating a total score), why rule-based logic supports responsive or conditional behavior (e.g., enforcing game rules), or why a data-driven approach is appropriate when patterns are identified f om collected data (e.g., recommending a song based on listening history). Students may also justify combining approaches when a solution requires both defined p ocedures and conditional or data-based decision-making. The emphasis is on evaluating and explaining the reasoning behind the chosen approach.

Practice(s): CT6	**Disposition(s):** Critical Thinking, Curiosity

MS-ALG-PS-05: Use an AI tool to generate outputs that assist in solving a computational problem.

Boundary Statement: Students should use age-appropriate AI tools or prebuilt models to generate outputs that support solving a computational problem. Examples include generating text to brainstorm ideas, using image recognition to classify objects, or asking a chatbot for troubleshooting help. Students should focus on interpreting and assessing AI outputs, not building or programming AI systems. Students can validate teacher generated output without generating outputs themselves if students are prohibited from using generative AI.

Students are not expected to train AI models or explain underlying algorithms.

Practice(s): CT8, HCD11	**Disposition(s):** Reflectiveness, Curiosit

HS-ALG-PS-01: Design an algorithm using appropriate data structures to solve a problem or express an idea.

Boundary Statement: Students should create a logical plan for a program that selects a specific way to organize data (e.g., using a list to store a collection of items, using an object to represent a single complex entity) to effectively solve a specific challenge. For example, a student might design a leaderboard system for a game by choosing a list to store player scores and designing a sorting algorithm to arrange them from highest to lowest.

Students are not expected to implement complex data structures (e.g., manual memory management for linked lists). The focus is on the selection and application of existing high-level structures available in modern programming languages to support the algorithm's goal.

Practice(s): CT7, CT9	**Disposition(s):** Critical Thinking, Resourcefulness

HS-ALG-PS-02: Optimize the design of an algorithm using procedural abstraction and control structures.

Boundary Statement: Students should evaluate existing algorithms to identify redundancies, inefficiencies, or unclear logic, and then efine them using p ocedural abstraction (e.g., defining functions or methods that encapsulate repeated steps). They should also apply control structures, including iteration and selection, to streamline processes and improve the algorithm's clarity, efficien , and reusability. For example, a long, repetitive script for calculating student grades could be optimized by creating a "calculateGPA" procedure and using a loop to process a list of scores.

Students are not expected to perform formal Big-O complexity analysis or optimize at the assembly/hardware level. This standard does not require the use of advanced dynamic programming or complex recursive optimizations, as the focus is on the structural improvement of the logic using fundamental programming constructs.

Practice(s): CT6, CT7	**Disposition(s):** Critical Thinking, Creativity

HS-ALG-PS-03: Evaluate algorithms for efficiency, correctness, and clarity, using metrics or test cases.

Boundary Statement: Students should analyze algorithms for efficiency (time or memory use) correctness of outputs, and clarity (human readability), using simple metrics or test cases. For example, they could compare two algorithms for finding the la gest number in a list by testing runtime when using different sized lists. Metrics may be qualitative or comparative (not formal analysis).

Students are not expected to apply formal mathematical proofs or Big O notation. Students are not required to analyze all possible inputs.

Practice(s): CT9	**Disposition(s):** Critical Thinking, Reflectivenes

HS-ALG-PS-04: Describe the differences between deterministic and probabilistic algorithms.

Boundary Statement: Students should be able to explain that deterministic algorithms always produce the same output from a specific input and ollow the same sequence of states, whereas probabilistic (randomized) algorithms use a source of randomness as part of their logic, which may result in different outputs or runtimes across different executions. For example, students could describe how a deterministic search algorithm always checks a list in a fi ed order, whereas a probabilistic approach might pick elements at random to find a ta get more quickly in certain datasets.

Students are not expected to calculate formal probability distributions or perform complex mathematical proofs.

Practice(s): CT6	**Disposition(s):** Critical Thinking, Reflectivenes

HS-ALG-PS-05: Evaluate AI-generated output to assess bias, accuracy, and potential harms.

Boundary Statement: Students should analyze and critique AI-generated content (e.g., text, images, code) for accuracy, bias, and potential harms. For example, they might examine how an image-generation tool underrepresents certain groups or produces inaccurate results due to biased training data. Students should focus on critical evaluation of outputs rather than technical implementation.

Students are not expected to build or debug AI models or understand their internal mathematical algorithms.

Practice(s): ESR1, CT9	**Disposition(s):** Critical Thinking, Sense of Belonging

Machine Learning

EK-ALG-ML-02: Recognize attributes of objects to notice patterns and make decisions.

Boundary Statement: Students should notice and identify that objects (e.g., people, animals, plants, other items) have specific characteristics (e.g., olor, size, shape, texture) that can be used to classify them. For example, students might notice that a square and a triangle have different shapes, but recognize that they share the attribute of being red so they can be grouped together based on color.

Students are not expected to recognize hidden attributes that cannot be seen or felt. Students are not required to work with more than one or two clear, observable attributes at a time to make a decision.

Practice(s): IC3, CT7	**Disposition(s):** Critical Thinking, Curiosity

E1-ALG-ML-02: Recognize that AI systems are technologies that use patterns in data to make decisions or generate new things.

Boundary Statement: Students should be able to identify that AI systems are a technology created by people that looks for repeating patterns in information to make a choice or create something new. For example, students recognize and verbally express that an app knows which book they may want to read next because it has seen a pattern of the books they have read before.

Students are not expected to understand the different types of machine learning (e.g., supervised, unsupervised).

Practice(s): IC3, CT7	**Disposition(s):** Critical Thinking, Reflectivenes

E2-ALG-ML-02: Examine how data is used to train a machine learning model.

Boundary Statement: Students should explore simple examples of how a trained machine learning model identifies p tterns in data and makes predictions or recommendations based on those patterns. Students should notice that models can make more mistakes for some kinds of examples than others, especially when the training examples were mostly of one kind. Students should focus on observing that machine learning models use patterns in data to make predictions or choices and that limited or unbalanced examples can lead to unfair or uneven results. For example, they might label images for a teacher-trained model and observe how well it classifies new pictu es or use a drawing app that predicts what object they are sketching before they finish

Students are not expected to independently train machine learning models.

Practice(s): ESR1, CT7	**Disposition(s):** Curiosity, Critical Thinking

E3-ALG-ML-02: Investigate how a machine learning model can change when new data is added to a training set.

Boundary Statement: Students should observe and describe what happens when a machine learning model receives additional training data. They should notice when adding missing or underrepresented examples helps the model make more accurate predictions. For example, students could use an image-recognition tool to identify animals, then add more pictures to see how the model's predictions change or improve. Students should focus on observing how new data affects outcomes.

Students are not expected to train their own machine learning models or understand the technical or mathematical processes behind them. The term bias may be introduced informally, but students are not expected to use it. Students are not expected to analyze complex datasets or create models from scratch.

Practice(s): CT7, CT9	**Disposition(s):** Critical Thinking, Curiosity

E4-ALG-ML-02: Analyze relationships between the properties of training data and a machine learning model's output.

Boundary Statement: Students should examine how characteristics of training data (e.g., amount, accuracy, labeling quality, representativeness) affect a machine learning model's performance. Students should connect unbalanced or poorly labeled data to inaccurate outputs and may learn the term bias to describe this pattern. Students should focus on making qualitative observations (e.g., "More data improves accuracy." "Labeling mistakes lead to incorrect predictions."). For example, students might observe that when too few images are used to train a classifie , the model often mislabels new images, leading them to conclude that data quantity and quality influen e accuracy.

Students are not expected to use statistical formulas, understand machine learning works mathematically, or define or orrect specific bias types

Practice(s): CT9, HCD11	**Disposition(s):** Reflectiveness, Critical Thinkin

E5-ALG-ML-02: Train a machine learning model to make a classification or prediction.

Boundary Statement: Students should upload a labeled dataset (e.g., text, numbers, images, sounds, poses) to train a machine learning model that classifies, p edicts, or recommends. Students should reflect on esults that do not match expectations (e.g., who is missing from the training data, how adding more diverse examples might make results more accurate). Students should focus on recognizing that training data influen es model accuracy. For example, students might create a labeled dataset of animal sounds to train a machine learning model, test its accuracy on new examples, and reflect on outputs th t differ from their expectations.

Students are not expected to understand the mathematical processes behind training a machine learning model to make predictions. Students are not required to use specific tools or specialized hardware.

Practice(s): CT8, CT9	**Disposition(s):** Critical Thinking, Reflectivenes

MS-ALG-ML-06: Hypothesize how a machine learning model generates classifications or predictions.

Boundary Statement: Students should develop a hypothesis about how a machine learning model might be processing information based on the observed inputs and outputs of the model. Students should focus on forming credible, evidence-based inferences from experimentation and observation. For example, students might test an image classific tion tool by submitting several pictures of animals and, after noticing that it consistently labels striped animals as "tiger," hypothesize that the model is prioritizing visible patterns like stripes over other features such as size or habitat.

Students are not expected to explain exactly how machine learning models work internally.

Practice(s): CT7, CT9	**Disposition(s):** Critical Thinking, Reflectivenes

MS-ALG-ML-07: Investigate ways to improve the accuracy of a machine learning model and reduce bias by refining the quality of examples and nonexamples in the training data.

Boundary Statement: Students should test a machine learning model to evaluate the outputs, identify overrepresented or missing examples in the training data, and suggest improvements to increase accuracy or reduce bias. Students should focus on testing, interpreting results, and suggesting refinements based on eviden e. For example, students might notice that a model misidentifies ertain objects and propose adding more diverse examples, then retest to see if performance improves.

Students are not expected to create machine learning models or understand the technical mechanics of training. Students are not expected to make changes to machine learning models; this can be a conceptual discussion.

Practice(s): ESR1, CT9	**Disposition(s):** Critical Thinking, Reflectivenes

MS-ALG-ML-08: Evaluate the features and limitations of a machine learning model.

Boundary Statement: Students should examine machine learning model documentation, including model cards, to determine the model's features and limitations. Students should review information about the machine model, including (a) datasets used for training, including potential sources of bias; (b) the contexts or scenarios in which the model should or should not be used; (c) the computing power or time required for operation; and (d) potential risks, ethical concerns, privacy considerations, and recommendations for fair and responsible use. Students should evaluate the observable inputs and outputs of a machine learning model for utility and reliability. Students should focus on reviewing documentation and analysis, not implementation or debugging.

Students are not expected to build a machine learning model or access proprietary training data.

Practice(s): ESR1, IC3	**Disposition(s):** Critical Thinking, Reflectivenes

HS-ALG-ML-06: Justify the selection of a type of AI algorithm to accomplish a task.

Boundary Statement: Students should support the selection of an AI algorithm for a specific task by evaluating its strengths and limitations relative to the problem requirements. This justific tion may consider data size and type, human interpretability, and desired outcomes. Students should focus on evaluating appropriateness, not technical construction. For example, students might explain that a Decision Tree is suitable for classifying spam emails because its logic is transparent and easy to explain, even if less accurate than other models.

Students are not expected to understand or implement the underlying mathematics or statistics of these algorithms.

Practice(s): IC3, CT7	**Disposition(s):** Critical Thinking, Reflectivenes

HS-ALG-ML-07: Evaluate training data by examining its source, quality, representativeness, potential biases, and privacy implications.

Boundary Statement: Students should critically evaluate datasets before use by examining their origin, purpose, accuracy, completeness, representativeness, and privacy implications. For example, when using crime statistics, students might check where the data originated (e.g., police reports, surveys), assess its quality, ensure all communities are represented, and identify potential biases or privacy concerns.

Students are not expected to conduct advanced statistical analyses to prove representativeness or to apply formal bias correction techniques.

Practice(s): ESR1, ESR2	**Disposition(s):** Reflectiveness, Critical Thinkin

HS-ALG-ML-08: Develop a machine learning model for a chosen task using appropriate data and tools.

Boundary Statement: Students should develop a machine learning model for a defined task by selecting suitable tools and data, training the model, and testing its performance against task requirements. Students should focus on applying available tools to create and evaluate functional models, not on mastering the mathematics that underlie them. For example, a student could use a block-based platform or a simple Python library to create a model that recognizes handwritten digits or classifies objects

Students are not expected to write algorithms from scratch or use advanced programming languages.

Practice(s): CT8, CT9	**Disposition(s):** Creativity, Critical Thinking

Impacts of Algorithms & Design

EK-ALG-IM-03: Describe how people create algorithms to solve problems.

Boundary Statement: Students should recognize that computers function according to algorithms created by people. They should describe, in simple terms, the human process of developing an algorithm (i.e., identifying a task, breaking it into smaller parts, and ordering steps to reach a goal). For example, students might describe how people design instructions for making a sandwich or sorting classroom supplies.

Students are not expected to describe all aspects of computational thinking. Students are not required to understand how algorithms are translated into computer code.

Practice(s): ESR1, IC3	**Disposition(s):** Critical Thinking, Reflectivenes

E1-ALG-IM-03: Explain how a change to an algorithm leads to a different outcome.

Boundary Statement: Students should express, through everyday activities or drawings, that changing the order of steps can produce different results. For example, when following two sets of instructions to build a block tower, one set might create a tall, stable tower, whereas another creates a shorter or weaker one.

Students are not expected to write or modify computer code. Students are not required to understand complex algorithms used in digital tools.

Practice(s): ESR1, IC3	**Disposition(s):** Critical Thinking, Reflectivenes

E2-ALG-IM-03: Describe how an algorithm might impact peers in varied situations.

Boundary Statement: Students should evaluate how algorithms affect peers by determining if outcomes are fair or unfair and explaining their reasoning using simple criteria. Students should recognize that the same algorithm can affect different people in different ways depending on the situation. For instance, when a teacher draws names from a bucket to choose a line leader, students might evaluate whether this is fair by considering if everyone has an equal chance, if anyone is left out, or if the same person might be chosen multiple times while others never get a turn.

Students are not expected to analyze technical details of how algorithms work, understand complex concepts like bias or discrimination, design solutions to fix u fair algorithms, or evaluate algorithms beyond determining basic fairness in concrete, familiar situations.

Practice(s): ESR1, IC3	**Disposition(s):** Critical Thinking, Reflectivenes

E3-ALG-IM-03: Compare how different algorithms may affect outcomes, situations, and people with a wide range of needs.

Boundary Statement: Students should evaluate examples of algorithms that produce different outcomes and discuss how those differences may help or harm people with diverse needs or perspectives. Students should focus on comparing impacts, not on programming or implementation. For example, students might consider how a navigation app choosing the fastest route differs from one choosing the safest route.

Students are not expected to analyze algorithms at a technical level or evaluate efficiency o performance trade-offs

Practice(s): ESR1, IC3	**Disposition(s):** Critical Thinking, Reflectivenes

E4-ALG-IM-03: Evaluate how different algorithms for solving the same problem produce outcomes that may benefit or disadvantage different groups of people.

Boundary Statement: Students should compare two or more algorithms that solve the same problem and explain how each affects people differently. Examples include different ways to sort items, plan routes, or schedule activities, noting who benefits or is disadvantaged

Students are not expected to design new algorithms or evaluate effectiveness. Students are not required to analyze algorithmic bias.

Practice(s): ESR1, IC3	**Disposition(s):** Critical Thinking, Reflectivenes

E5-ALG-IM-03: Articulate how human-centered design principles are incorporated into the development of a computing technology.

Boundary Statement: Students should identify familiar examples of computing technologies (e.g., websites, voice assistants, video games, recommendation systems) and explain how human-centered design considerations (e.g., empathy, user needs, requirements, accessibility, fairness, sustainability, accountability) can be incorporated into their development. Students should consider the impact of their values (e.g., personal, community, larger values) on the design. They should also recognize that computing technologies are created by people, which may sometimes lead to unfair or inaccurate outcomes.

Students are not expected to understand the technical details of computing technologies.

Practice(s): ESR1, IC3	**Disposition(s):** Critical Thinking, Reflectivenes

MS-ALG-IM-09: Evaluate which human-centered design principles are present or missing in an existing computing technology.

Boundary Statement: Students should analyze familiar computing technologies (e.g., websites, mobile apps, video games, school platforms, recommendation systems) to determine how human-centered design principles are reflected in their eatures. Students should identify evidence of empathy, accessibility, user needs, fairness, sustainability, and accountability and explain where these principles are effectively incorporated or overlooked. Students should consider how different users (e.g., multilingual users, users with disabilities, users with limited internet access) might experience the technology differently and evaluate the potential impacts of design choices. Students may suggest improvements to the computing technologies examined.

Students are not expected to design or implement full computing technologies. Students are not required to conduct formal user research, perform usability testing, or apply detailed accessibility standards.

Practice(s): ESR1, HCD10	**Disposition(s):** Creativity, Sense of Belonging

MS-ALG-IM-10: Examine evidence of beneficial and harmful impacts, ethical issues, and biases of algorithms encountered in daily life.

Boundary Statement: Students should identify and describe common ethical issues and biases in AI systems using familiar examples (e.g., photo apps misidentifying people, voice assistants misunderstanding accents, recommendation systems reinforcing prior interests). They should evaluate whether the outcomes of these systems seem fair and inclusive and explain why biases may occur in simple, concrete terms.

Students are not expected to program or modify AI tools, understand neural networks, or study global AI policy.

Practice(s): ESR1, HCD10	**Disposition(s):** Critical Thinking, Reflectivenes

MS-ALG-IM-11: Modify an algorithm to address a specific societal impact, ethical issue, or bias.

Boundary Statement: Students should design modific tions to an existing algorithm, verbally or through diagrams or flowcharts, to ma e its outcomes fairer or more inclusive. Students should focus on reducing bias and improving equity through simple, conceptual modific tions. For example, they might suggest adding a language option to a chatbot or adjusting a music playlist generator to include more diverse artists. Students should explain how their modific tion reduces bias or improves equity.

Students are not expected to design complex algorithms from scratch or apply statistical fairness techniques.

Practice(s): ESR1, HCD12	**Disposition(s):** Creativity, Critical Thinking

HS-ALG-IM-09: Design a computing technology using human-centered design principles.

Boundary Statement: Students should design computing technologies that are inclusive and accessible by following a human-centered design process. This includes researching user needs, exploring design options, and anticipating the social and ethical impacts of their design. For example, when creating a website, students might ensure sufficient olor contrast, keyboard navigation, and clear language for users with differing abilities. Students should also identify laws that set requirements for accessibility (e.g., ADA, IDEA, Section 508).

Students are not expected to conduct large-scale user studies or meet every criterion of formal accessibility standards (e.g., Web Content Accessibility Guidelines, WCAG).

Practice(s): CT8, HCD10	**Disposition(s):** Resourcefulness, Critical Thinking

HS-ALG-IM-10: Evaluate the ethical implications, societal impacts, and potential biases of rule-based and data-driven algorithms.

Boundary Statement: Students should evaluate how rule-based and data-driven algorithms can produce biased or inequitable outcomes. They should identify potential sources of bias, such as explicit decision rules embedded in a system (e.g., filtering candid tes by ZIP code) or unrepresentative or historically biased training data in data-driven models. For example, students might analyze a hiring algorithm that excludes candidates living in certain ZIP codes or examine a résumé screening model trained on past hiring data that disadvantages applicants with names from underrepresented cultural or linguistic backgrounds. Students should provide recommendations on how algorithms can be refined. Students' analyses should emain conceptual, focusing on observable impacts and ethical reasoning.

Students are not expected to audit proprietary systems or use advanced statistical methods.

Practice(s): ESR1, CT9	**Disposition(s):** Critical Thinking, Reflectivenes

HS-ALG-IM-11: Articulate the values embedded in the design of an algorithmic system.

Boundary Statement: Students should identify and explain the values reflected in the design f algorithmic systems, recognizing that every design choice involves prioritizing certain outcomes or perspectives. Students should focus on identifying underlying priorities (e.g., efficien , privacy, fairness, transparency) that shape algorithmic behavior. For example, students might examine a social media platform's recommendation system and discuss how its design favors engagement and attention over accuracy or civility.

Students are not expected to deconstruct proprietary systems or analyze full business models.

Practice(s): ESR1, HCD10	**Disposition(s):** Critical Thinking, Reflectivenes

Programming

Program Development

EK-PRO-PD-04: Create a sequence of commands to solve a problem or express an idea.

Boundary Statement: Students should arrange a small set of commands in order (e.g., to help a character reach an object, tell a short story, follow a classroom routine). Using manipulatives, arrows, or simple digital programs is appropriate. Unplugged activities are appropriate.

Students are not expected to write code or use control structures beyond sequencing. Students are not required to use computers.

Practice(s): CT7, CT8	**Disposition(s):** Creativity, Curiosity

E1-PRO-PD-04: Create code from an algorithm that includes sequence to solve a problem or express an idea.

Boundary Statement: Students should translate algorithms with sequential steps into block-based code. Tasks such as translating an algorithm to move a character or navigate a maze are appropriate.

Students are not expected to design the algorithm themselves or use selection or iteration.

Practice(s): CT7, CT8	**Disposition(s):** Creativity, Persistence

E2-PRO-PD-04: Create code from an algorithm that includes sequence, events, and iteration to solve a problem or express an idea.

Boundary Statement: Students should translate a provided algorithm that begins with an event and includes sequential steps and iteration (e.g., loops) into a working program. Writing an animation in a block-based language from a storyboard or coding a robot in collaboration with a partner to dance from an algorithm of arrows are appropriate.

Students are not expected to work with selection (e.g., conditionals [if/then statements]), variables, or nested loops. Teachers may provide the original algorithm, if desired.

Practice(s): CT7, CT8	**Disposition(s):** Persistence, Resourcefulness

E3-PRO-PD-04: Develop code from a student-created algorithm that includes a combination of sequence, events, iteration, and selection to solve a problem or express an idea.

Boundary Statement: Students should implement their own or a classmate's algorithm in a block-based environment using events, iteration (e.g., loops) for efficien , and basic decisions (e.g., conditionals [if/then statements]). Students should also give credit to the algorithm author and implementer. For example, turning a storyboard for a game that includes a game character moving forward, jumping over obstacles repeatedly, and choosing different paths when encountering a fork in the road into a program using a block-based programming language is appropriate.

Students are not expected to use text-based languages or design multilevel projects. Students are not required to implement a single program that includes every control structure. However, all control structures should be included throughout multiple programs.

Practice(s): ESR2, CT8	**Disposition(s):** Persistence, Resourcefulness

E3-PRO-PD-05: Use constructive feedback to improve a program.

Boundary Statement: Students should give and apply specific, helpful suggestions to imp ove the function or design of their programs. Incorporating feedback from peers, teachers, or users into a program is appropriate. Teacher-provided templates for giving constructive feedback are appropriate.

Students are not expected to interpret complex error reports, use advanced tools, or overhaul program structure or design. Students are not required to defend against all critiques or implement all feedback.

Practice(s): IC5, CT9	**Disposition(s):** Critical Thinking, Persistence

E4-PRO-PD-04: Compare different programming solutions to the same problem based on correctness and clarity.

Boundary Statement: Students should compare multiple short programs or code segments meant to solve the same task, noting which ones produce the correct result and which are easier to understand or follow. Examples that use short block-based or text-based code segments are appropriate.

Students are not expected to create multiple versions themselves, evaluate efficien , or conduct formal code reviews.

Practice(s): IC3, CT9	**Disposition(s):** Critical Thinking, Curiosity

E4-PRO-PD-05: Collaborate with a team by offering a meaningful contribution to creating a program.

Boundary Statement: Students should work in small groups to plan and create a simple program, contribute ideas or code segments, or test functionality while practicing respectful communication, active listening, and shared tasking. Adding a feature, fixing an er or, improving a design element, or helping organize the code are appropriate.

Students are not expected to use professional collaboration tools, manage large projects, or apply formal methodologies.

Practice(s): IC5, CT8	**Disposition(s):** Sense of Belonging, Creativity

E5-PRO-PD-04: Create a novel program by modifying or combining elements of existing programs.

Boundary Statement: Students should make purposeful changes to a program by incorporating teacher-created, peer-created, or other code segments to create a unique variation of the original program, giving appropriate credit for code segments used. Remixing or adapting existing code to add new features, levels, or outcomes, including credit for code reuse through comments in code or in external documentation, is appropriate.

Students are not expected to write entirely original, complex programs from scratch or manage large-scale integrations of multiple projects. Students are not required to incorporate AI-generated code.

Practice(s): ESR2, CT8	**Disposition(s):** Resourcefulness, Critical Thinking

E5-PRO-PD-05: Construct individual components of a program that are collaboratively assembled into a programming project.

Boundary Statement: Students should individually build small code components and integrate them with teammates into a complete functional program. For example, in a block-based environment, one student might construct the code component for a scoring system using variables and selection, while another constructs the player movement component. They then integrate and test these pieces to ensure the final p oject works.

Students are not expected to use version control or design full system architecture.

Practice(s): IC4, CT8	**Disposition(s):** Creativity, Sense of Belonging

MS-PRO-PD-12: Use a procedure to structure code for clarity and reusability.

Boundary Statement: Students should use procedures to make code easier to read and reuse by grouping a set of steps under a clear, descriptive name and calling the procedure in one or more places in the program. Students should recognize an opportunity for a procedure (e.g., repeated or logically grouped steps), create the procedure, replace repeated code with procedure calls, and describe how this improves clarity and reusability (e.g., reduces duplication, makes updates easier, helps organize the program). Procedures may be teacher-provided, student-created, or developed from starter code.

Students are not expected to write procedures with parameters or return values, to design large modular systems, or to manage many interacting procedures, modules, or objects.

Practice(s): CT7, CT8	**Disposition(s):** Creativity, Critical Thinking

MS-PRO-PD-13: Use reference documentation in program development.

Boundary Statement: Students should use reliable documentation that helps write, debug, and refine their programs. They should use teacher-provided or self-selected documentation to understand functions or blocks and search for common error messages to identify causes and solutions.

Students are not expected to read professional-level documentation or apply coding concepts beyond their grade level.

Practice(s): IC4, CT9	**Disposition(s):** Resourcefulness, Curiosity

MS-PRO-PD-14: Justify the importance of attribution and intellectual property when developing computing technologies.

Boundary Statement: Students should define intellectual p operty and explain why giving credit to original creators matters. They should recognize that programming and computing technologies make use of multiple artifacts (e.g., code, images, music, text), and understand the difference between learning from others' code and plagiarizing it. Identifying whether code is open source, free to use, or proprietary and providing basic attribution in their own work are appropriate. Students should focus on ethical awareness and responsible use of others' work.

Students are not expected to study copyright law, fair use, or licensing in depth or to license their own code.

Practice(s): ESR2, IC3	**Disposition(s):** Critical Thinking, Reflectivenes

MS-PRO-PD-15: Develop a program utilizing inclusive collaboration practices.

Boundary Statement: Students should actively share ideas during planning, provide and receive constructive feedback in all stages of program development, and collaborate effectively on shared digital tools. Students should focus on productive teamwork and inclusive communication. Practicing equal participation, clear communication, and respect for others' contributions is appropriate.

Students are not expected to use formal collaboration methods, like Agile or Scrum, or professional project management software.

Practice(s): IC4, CT8	**Disposition(s):** Sense of Belonging

HS-PRO-PD-12: Create a modular program that uses procedures, external libraries, or objects to improve reusability and readability.

Boundary Statement: Students should decompose a complex problem into smaller, manageable parts and implement each part as a distinct procedure, external library, or object. Students should focus on creating clear, reusable, and well-organized code. For example, a game program might separate player movement and scoring.

Students are not expected to use advanced software design patterns or deep inheritance structures.

Practice(s): CT7, CT8	**Disposition(s):** Critical Thinking, Creativity

HS-PRO-PD-13: Use documentation, libraries, application programming interfaces (APIs), and other tools in program development.

Boundary Statement: Students should locate and use external resources to support program development, including built-in documentation, common libraries, simple APIs, and features of an Integrated Development Environment (IDE) (e.g., debuggers, syntax highlighting). Students should focus on effectively using available resources to enhance programs. For example, using a math library instead of writing a trigonometric function from scratch is appropriate.

Students are not expected to create APIs or advanced development tools.

Practice(s): ESR2, CT9	**Disposition(s):** Resourcefulness, Persistence

HS-PRO-PD-14: Apply appropriate attribution of intellectual property when developing a computing technology.

Boundary Statement: Students should apply intellectual property principles by giving accurate credit for any code or digital media created by others. They should demonstrate ethical practice by documenting sources according to their terms of use or license. For example, a student creating a game might include a "Credits" section listing each asset's creator, source, and license.

Students are not expected to master copyright law.

Practice(s): ESR2, IC3	**Disposition(s):** Reflectiveness, Sense of Belonging

HS-PRO-PD-15: Collaborate on a programming project using a defined workflow that includes design documentation and clear task roles.

Boundary Statement: Students should use collaborative workflows to o ganize group projects (e.g., assigning roles, maintaining shared design documents, using basic version control tools to manage changes). Students should focus on learning structured teamwork practices with accessible, classroom-ready tools. For example, a group of students might use a project management tool to assign tasks or a collaborative document for design specific tions.

Students are not expected to use professional-grade tools or manage large-scale projects.

Practice(s): IC4, IC5	**Disposition(s):** Sense of Belonging, Persistence

Variables & Data Storage

EK-PRO-VD-05: Identify everyday gestures and symbols that represent information people use to make choices.

Boundary Statement: Students should recognize and explain common symbols, signs, and gestures they see in familiar environments such as home, school, or playgrounds. Students should be able to state what each symbol means or what action it signals. Symbols with clear, immediate meanings are appropriate (e.g., stop signs, traffic lights, a teacher's raised hand or quiet, thumbs up or down, pictures on classroom labels).

Students are not expected to interpret abstract or unfamiliar symbols, read written words on signs, or analyze why specific symbols we e chosen. Students are not required to understand cultural variations, complex decision-making, or digital icons requiring prior background knowledge.

Practice(s): IC3, CT7	**Disposition(s):** Critical Thinking, Reflectivenes

E1-PRO-VD-05: Identify terms that refer to data values that change over time in everyday life.

Boundary Statement: Students should name real-world examples of information that changes, such as temperature, time, score, or age. For example, students can identify that the score of a game changes during the game, that the temperature outside changes throughout the day and night, and that their age changes each year. Students should recognize that some information changes while other information stays the same.

Students are not expected to use the formal term variable or understand the programming concept of assignment.

Practice(s): IC3, CT7	**Disposition(s):** Critical Thinking, Reflectivenes

E2-PRO-VD-05: Label different representations of information with a name and whether its value is constant or changes.

Boundary Statement: Students should give logical names to pieces of information they encounter in real life and determine whether the value of each piece of information changes or stays constant. Students should focus on describing information in meaningful, age-appropriate terms. Examples from digital contexts (e.g., score in a game) and nondigital contexts (e.g., temperature in an oven, day of the week) are appropriate.

Students are not expected to use programming vocabulary or write code.

Practice(s): IC3, CT7	**Disposition(s):** Critical Thinking, Reflectivenes

E3-PRO-VD-06: Identify the variables being stored and manipulated in a program.

Boundary Statement: Students should read completed programs to find variables. Students should recognize that each variable has a unique name that must be spelled consistently. Identifying variables that track a game score, user input, or collected data is appropriate.

Students are not expected to assign variables to specific d ta types (e.g., string, integer, Boolean) or write code themselves.

Practice(s): IC3, CT7	**Disposition(s):** Critical Thinking, Persistence

E4-PRO-VD-06: Trace how data flows and changes variable values in a program.

Boundary Statement: Students should follow short, age-appropriate programs to observe how variable values change as a program runs and be able to explain observable changes to variable values. For example, they might trace how a score counter increases with each correct answer or how a temperature tracker updates as new readings are added.

Students are not expected to design variable-driven programs, work with multiple data types, or understand memory allocation, scope, or data structures.

Practice(s): CT7, CT9	**Disposition(s):** Critical Thinking, Persistence

E5-PRO-VD-06: Use variables to store, compare, and modify data within a program.

Boundary Statement: Students should use variables in block-based or beginner text-based environments to store, compare, and update data that affects program behavior. Examples include tracking a player's score, comparing time values to trigger an event, or updating health or points after an action.

Students are not expected to manage complex data types, arrays, or mathematical formulas involving multiple variables. Students are not required to understand variable scope.

Practice(s): CT7, CT8	**Disposition(s):** Critical Thinking, Persistence

MS-PRO-VD-16: Use variables of multiple data types to store, access, and manipulate data within a program.

Boundary Statement: Students should create programs that use variables of multiple data types (e.g., integers, strings, Booleans) to store, access, and change data during execution. Students should be able to update individual values, use selection to make decisions based on variable values, and use iteration to repeat a process over time (e.g., repeated input/updates, repeated calculations, repeated checks) without requiring collections. Students should perform basic manipulations such as incrementing and decrementing, accumulating a running total, tracking a maximum or minimum, counting occurrences, and transforming values (e.g., converting between types or combining text).

Students are not expected to use or implement data structures (e.g., lists/arrays, dictionaries/maps, objects) or use professional or industry-specific syntax and onventions beyond what their programming environment requires.

Practice(s): CT7, CT8	**Disposition(s):** Persistence, Critical Thinking

HS-PRO-VD-16: Create a program that uses appropriate data structures to store, access, and manipulate data.

Boundary Statement: Students should design and implement programs that use data structures such as arrays, lists, or dictionaries to organize, access, and modify data effectively. Students should focus on selecting and applying suitable structures to manage data efficient . For example, students might create a program that stores and retrieves information about daily tasks, chores, or student names and grades.

Students are not expected to implement advanced data structures like linked lists, stacks, queues, trees, or graphs.

Practice(s): CT7, CT8	**Disposition(s):** Critical Thinking, Persistence

Reading & Documenting

EK-PRO-RD-06: Describe how a sequence of commands completes a task.

Boundary Statement: Students should observe and describe, in their own words, what happens when simple code runs (e.g., "The robot moved three steps." "The character turned red."). Gestures or short phrases to express what happens are appropriate.

Students are not expected to read text-based code, explain why it works, use technical vocabulary, or explain advanced programming concepts.

Practice(s): IC3, CT6	**Disposition(s):** Critical Thinking, Reflectivenes

E1-PRO-RD-06: Explain the function of code that includes an event and sequence.

Boundary Statement: Students should explain what each step in a short program does and how an event triggers what happens next. Expressing what happens when a sequence of movement blocks runs or when a tap or collision event starts an animation are appropriate.

Students are not expected to analyze advanced control structures (e.g., loops, conditionals), formal flo , syntax, or efficien .

Practice(s): IC3, CT6	**Disposition(s):** Reflectiveness, Critical Thinkin

E2-PRO-RD-06: Explain the steps taken to create a program.

Boundary Statement: Students should provide a simple step-by-step description of how they built their program and the specific function f the code blocks they used. Students should acknowledge which parts of the process and code were their own ideas and which parts were inspired or assisted by a peer or teacher. Students can explain verbally or in a written document. For example, a student could verbally express that they wanted a character to spin and first used a sequen e of turn commands but they changed the code to a "forever" loop because a classmate helped them to make the spinning better. Teacher-provided graphic organizers or sentence starters are appropriate.

Students are not expected to use advanced vocabulary, formal documentation, or written citations.

Practice(s): ESR2, IC3	**Disposition(s):** Reflectiveness, Critical Thinkin

E3-PRO-RD-07: Articulate how a specific segment of code contributes to the overall purpose of a program.

Boundary Statement: Students should use age-appropriate academic language to describe what a code segment does and how it supports the program's goal. For example, students sharing their understanding via annotated screenshots, prompted explanations, or brief presentations is appropriate.

Students are not expected to conduct formal code reviews or be the original author of the code.

Practice(s): IC3, CT6	**Disposition(s):** Critical Thinking, Reflectivenes

E4-PRO-RD-07: Document a program to clarify its functionality.

Boundary Statement: Students should use simple documentation (e.g., brief comments, labeled diagrams, short descriptions) to explain what their program does and how it works. The goal is to make the program easier for others to understand or modify.

Students are not expected to produce professional design documents or complex explanations of code structure, use version control, or write industry-style technical prose.

Practice(s): IC3, CT6	**Disposition(s):** Critical Thinking, Reflectivenes

E5-PRO-RD-07: Create embedded or external documentation for a programming project.

Boundary Statement: Students should produce documentation common to their programming language environment (e.g., in-code comments, project notes) describing its purpose in detail. Comments, virtual sticky notes, program descriptions or an external document are all appropriate options.

Students are not expected to follow professional templates or produce formal documentation sets. Teachers may opt to introduce guidelines or templates, but students are not required to follow any specific guidelines or documentation.

Practice(s): IC4, CT6	**Disposition(s):** Critical Thinking, Reflectivenes

MS-PRO-RD-17: Analyze the roles of iteration, selection, variables, and procedures in a segment of code.

Boundary Statement: Students should demonstrate how a segment of code works. Tracing the execution of a code segment and the value of variables through sequences of code (e.g., step-by-step in order), iteration (e.g., repeated sections of code), selection, and procedures (e.g., separate modules of code) are appropriate.

Students are not expected to debug or test the code or analyze procedures with parameters or return values.

Practice(s): IC3, CT7	**Disposition(s):** Critical Thinking, Reflectivenes

MS-PRO-RD-18: Analyze AI-generated code for accuracy and usability in a programming project.

Boundary Statement: Students should be provided with simple AI-generated code and evaluate whether any functions used actually exist, identify flaws in logic, and describe their opinion f the level of complexity and readability of the code. Students should also make suggestions for improvements to the code.

Students are not expected to evaluate code beyond their current level of understanding or generate code with AI themselves.

Practice(s): CT9, HCD11	**Disposition(s):** Reflectiveness, Critical Thinkin

HS-PRO-RD-17: Analyze how a segment of code works, including the role of parameters, return values, and data structures.

Boundary Statement: Students should read existing code and explain how its components interact to achieve the program's purpose. This includes explaining the roles of elements learned in prior years (i.e., sequence, iteration, selection, variables, and procedures) as well as elements learned in high school (i.e., parameters, return values, and data structures) and how these elements work together within a segment of code. For example, students could analyze a segment of code for a simple guessing game and explain how a variable stores the secret number, how a while loop continues until the correct number is guessed, and how an if/else statement provides feedback on whether the guess was too high or too low.

Students are not expected to analyze complex algorithms or systems with many interconnected data structures.

Practice(s): CT6, CT7	**Disposition(s):** Critical Thinking, Persistence

HS-PRO-RD-18: Evaluate AI-generated code for accuracy, reliability, and alignment with program requirements.

Boundary Statement: Students should evaluate AI-generated code by testing it against predetermined requirements and verifying that it produces correct and reliable results. They should analyze whether the code meets its intended purpose, handles a variety of inputs appropriately, and follows any given design constraints (e.g., using a specific d ta structure or algorithm). Students should focus on evaluating the quality and reliability of the output. For example, a student using an AI tool to generate a function that calculates an average should test edge cases (e.g., empty lists, single values, mixed positive and negative numbers) and confirm th t outputs are accurate and consistent with the task description.

Students are not expected to understand the underlying machine learning models or algorithms that power the AI tools.

Practice(s): CT6, CT9	**Disposition(s):** Critical Thinking, Reflectivenes

Testing & Refining

EK-PRO-TR-07: Identify a step in a sequence of commands that does not work as expected.

Boundary Statement: Students should recognize which step in a simple sequence is not working correctly or is causing a problem. Observing and identifying a block that led to an unexpected result in a block-based coding app or a command card that was incorrect in an unplugged activity are appropriate.

Students are not expected to explain why the error occurred, analyze program logic, or use formal debugging strategies.

Practice(s): CT6, CT9	**Disposition(s):** Critical Thinking, Persistence

E1-PRO-TR-07: Debug a program that includes a sequence of commands.

Boundary Statement: Students should identify and fix simple er ors in block-based programs they have developed or were provided by the teacher. Students should focus on identifying and correcting simple mistakes. For example, if a character is supposed to move and then jump, but the code blocks are in the wrong order, the student should be able to identify the problem and reorder the blocks.

Students are not expected to find omplex logical errors or debug programs that include loops or conditionals.

Practice(s): CT6, CT9	**Disposition(s):** Critical Thinking, Persistence

E2-PRO-TR-07: Debug a program that includes sequence, events, and iteration.

Boundary Statement: Students should test a simple, block-based program that includes sequence, events, and iteration, identify one obvious bug, and correct it. Bugs may involve commands in the wrong order (sequence), an event not triggering the intended action, or a loop repeating too few or too many times (iteration). For example, if a robot does not respond when clicked, moves in the wrong order, or draws only three sides of a square instead of four, students should locate the issue and adjust the relevant block.

Students are not expected to fix multiple or omplex errors or explain problems using technical terms.

Practice(s): CT6, CT9	**Disposition(s):** Critical Thinking, Persistence

E3-PRO-TR-08: Debug a program that includes a combination of sequence, events, iteration, and selection.

Boundary Statement: Students should locate and fix ommon logical or structural errors in programs using a combination of sequence, events, loops, and conditionals. For example, students might debug a block-based program in which commands are in the wrong order (sequence), a character does not move when clicked (event), a loop runs too many times (iteration), or an "if" block triggers the wrong action (selection).

Students are not expected to debug syntax errors in text-based languages or correct complex programs involving variables, functions, or nested conditions or loops. Students are not required to debug a single program that includes all control structures (i.e., sequence, events, iteration, and selection). However, across multiple debugging experiences, students should work with programs that collectively include all of these control structures.

Practice(s): CT6, CT9	**Disposition(s):** Critical Thinking, Persistence

E4-PRO-TR-08: Debug a program incrementally and repeatedly throughout the development process.

Boundary Statement: Students should test and fix their ode as they develop programs rather than completing all coding and then testing. For example, students can add a short sequence of blocks and then run the code to confirm th t it does what is expected. If the code does not behave as expected, students can find, eproduce, and fix the er or easily in the code they added incrementally. Students can use tracing skills to identify where code does not behave as expected. Debugging programs that include a combination of sequence, events, iteration, and selection are appropriate.

Students are not expected to debug programs involving procedures or nested structures. Students are not required to debug one single program that includes all control structures (sequencing, events, iteration, and selection); however, all control structures should be included throughout programs that students debug.

Practice(s): CT9, HCD11	**Disposition(s):** Persistence, Reflectivenes

E5-PRO-TR-08: Debug a program using systematic strategies.

Boundary Statement: Students should apply more than one systematic debugging strategy. Strategies such as reproducing the problem, reading error messages, tracing code, changing one element at a time, or adding temporary outputs (e.g., sounds, text, other alerts) to locate issues are appropriate. Students should focus on using clear, repeatable methods to find and fix syntax, logic, or runtim errors.

Students are not expected to use professional debugging tools or automated testing frameworks.

Practice(s): CT9, HCD11	**Disposition(s):** Persistence, Reflectivenes

MS-PRO-TR-19: Use systematic strategies to test, refine, and document changes to a computing technology to meet the intended purpose.

Boundary Statement: Students should select and implement a variety of techniques to find and fix er ors, explain code, refine a user interfa e, and give and receive constructive feedback during code review. Appropriate techniques might include testing input and output, using watchers, testing individual procedures, or using AI to make suggestions for alignment with design specific tions.

Students are not expected to use debugging software, unit testing frameworks, or version control systems or to follow documentation standards beyond basic commenting practices.

Practice(s): IC4, CT9	**Disposition(s):** Persistence, Critical Thinking

MS-PRO-TR-20: Refine a computing technology based on user feedback to improve its usability and accessibility.

Boundary Statement: Students should discuss user feedback with peers, consider the needs of diverse users, and identify which suggestions would most improve a computing technology's usability or accessibility. Students should focus on developing awareness of others' needs and incremental improvement.

Students are not expected to perform formal audits or implement complex accessibility frameworks.

Practice(s): HCD10, HCD11	**Disposition(s):** Critical Thinking, Reflectivenes

HS-PRO-TR-19: Evaluate a computing technology's alignment with design specifications and responsible design values, including its correctness, effectiveness, and user experience.

Boundary Statement: Students should systematically evaluate a computing technology that they or someone else created to determine whether it functions correctly, aligns with original design specific tions, and reflects esponsible design values. For example, a student might test a range of inputs to verify accuracy and reliability, examine edge cases, and assess usability, accessibility, clarity, and overall user experience. Students should compare the completed project to the intended features and purpose to confirm alignment

Students are not expected to use formal verific tion methods or master industry-specific tools

Practice(s): CT9, HCD11	**Disposition(s):** Reflectiveness, Critical Thinkin

HS-PRO-TR-20: Refine a computing technology based on user feedback, testing results, and responsible design values to improve its effectiveness and impact.

Boundary Statement: Students should use testing results and user feedback to make meaningful refinements to a omputing technology's functionality, usability, accessibility, accuracy, and efficien , but not necessarily all at the same time. For example, a student might add a help command, simplify instructions, or adjust text size and contrast to improve readability and accessibility.

Students are not expected to correct every functional or user-related issue or achieve full optimization.

Practice(s): HCD10, HCD11	**Disposition(s):** Critical Thinking, Reflectivenes

Data & Analysis

Data Collection & Preparation

EK-DAT-DC-08: Use collected data to help answer questions.

Boundary Statement: Students should collect data to answer one or more questions about a grade-appropriate topic involving two to three variables. Having students wonder about the world to create questions to answer is appropriate. Collecting simple survey data or recording measurements and creating graphs as a class in order to answer a question are also appropriate.

Students are not expected to work independently. Students are not required to use computational tools or collect large sets of data.

Practice(s): IC3, CT7	**Disposition(s):** Curiosity, Resourcefulness

E1-DAT-DC-08: Use multiple methods to collect both numeric and non-numeric data to help answer questions.

Boundary Statement: Students should use different ways (e.g., observation, measurement, surveys) to gather numbers and descriptive words to help answer one or more questions. For example, students might ask their classmates about their favorite fruit (descriptive words) or measure the height of a bean plant in cubes or with a ruler (numeric data).

Students are not expected to analyze or organize data into complex charts. Students are not required to use computational tools or formal statistical methods.

Practice(s): IC3, CT7	**Disposition(s):** Curiosity, Resourcefulness

E2-DAT-DC-08: Compare numeric and non-numeric types of data in terms of how they are collected and what information they provide.

Boundary Statement: Students should compare how numeric (quantitative) data can be used for math and measurement and non-numeric (categorical) data can be used to sort, group, and describe. Having students identify the kinds of pets the class has (non-numeric data), count the total number of pets in each category (numeric data), and make simple comparisons (e.g., more students have cats vs dogs) is appropriate.

Students are not expected to perform more sophisticated data transformations or analyses, nor are students expected to identify finer-grained d ta types (e.g., integers, characters). Students are not required to use technology to collect or compare different types of data.

Practice(s): IC3, CT7	**Disposition(s):** Reflectivenes

E3-DAT-DC-09: Evaluate numeric and non-numeric data for accuracy and completeness.

Boundary Statement: Students should focus on checking for missing data and recognizing data that seems unreasonable (e.g., values that are much too large or much too small, data that is the wrong type). Students may use a survey to collect data that they record on a chart. Students may also use data supplied by the teacher.

Students are not expected to create complex graphs or work with large datasets. Students are not required to use technology for collection, recording, or evaluation of data.

Practice(s): IC3, CT7	**Disposition(s):** Critical Thinking, Curiosity

E4-DAT-DC-09: Organize collected data into a table using a computational tool, with rows representing records and columns representing attributes.

Boundary Statement: Students should practice organizing small sets of collected or provided data into simple tables, ensuring that each row represents one record (e.g., a survey respondent, a pet) and each column represents an attribute (e.g., age, favorite color, type of animal). Students may use paper-based tables and templates prior to using computational tools for this work.

Students are not expected to design complex databases, manage large datasets, or use advanced spreadsheet functions. Students do not need to understand relational databases, formulas, or data visualization techniques.

Practice(s): CT8	**Disposition(s):** Critical Thinking, Resourcefulness

E5-DAT-DC-09: Use computational tools to collect and organize different types of data.

Boundary Statement: Students should demonstrate their ability to use computational tools to collect (e.g., surveys, sensors) and organize (e.g., spreadsheets) data. Students should collect and distinguish between quantitative (numeric) and qualitative (descriptive and non-numeric) data. When organizing the data, students should use a computational tool to efficiently sort and oup similar data (e.g., sort from highest to lowest temperature, sort non-numeric responses alphabetically).

Students are not expected to create their own data collection tools from scratch. Students are not required to create data visualizations or to analyze or interpret the data collected.

Practice(s): CT8, CT9	**Disposition(s):** Critical Thinking, Persistence

MS-DAT-DC-21: Evaluate how different levels of precision and granularity in data collection affect accuracy, storage, and analysis.

Boundary Statement: Students should explain the difference between precision and granularity in data collection and evaluate how these choices affect data accuracy, storage requirements, and interpretation. Students should identify and explain situations in which higher precision or granularity might be more important. For example, students might compare tracking daily step counts (lower granularity, less storage) versus tracking steps every minute (higher granularity, more storage) or recording ages in years (lower precision) versus years and months (higher precision). Comparing medical monitoring data (requiring high precision and granularity for patient safety) with casual fitness tracking data (where lower precision and granularity are sufficient) is ap opriate.

Students are not expected to collect data at varying precision or granularity levels. Students are not expected to understand the technical details of data collection processes.

Practice(s): IC3, CT7	**Disposition(s):** Critical Thinking

MS-DAT-DC-22: Explain how data and its associated metadata can be used to answer questions.

Boundary Statement: Students should be able to explain how data and metadata serve different but complementary purposes when answering questions. Students should understand that metadata is data about data (i.e., it provides context that helps interpret the primary data content). For example, a photo's image data shows what was captured, whereas its metadata (e.g., timestamp, location, camera settings) provides context about when, where, and how the photo was taken. Students should be able to use computational tools to view metadata (e.g., viewing file p operties, examining web page source code). Students should recognize that metadata can be collected automatically without their explicit knowledge (e.g., their phone adding location and time data to photos).

Students are not expected to understand technical standards for metadata formats or structures. Students are not required to focus on any particular metadata tool or to manually create or edit metadata fields. Students a e not expected to provide exhaustive lists of all possible metadata types for different file ormats.

Practice(s): ESR2, IC3	**Disposition(s):** Critical Thinking, Reflectivenes

MS-DAT-DC-23: Use a computational tool to sort, filter, group, and summarize structured data.

Boundary Statement: Students should use a spreadsheet or other age-appropriate computational tool to apply basic operations such as sorting, filtering, g ouping, and calculating simple summaries (e.g., counts, averages) on a structured dataset. Students can work with a pre-existing data set of a grade-level-appropriate size. Students should explain why they are using each operation and how it helps answer their data question (e.g., "I am filtering the d ta to show only results from 2023 so I can analyze recent trends," "I am grouping by product type and calculating total sales to find our best selling category.")

Students do not need to collect data to fulfill this standa d. Students are not expected to clean messy datasets, correct inconsistencies, restructure text fields, or p epare unstructured data for analysis. Students do not need to master any computational programs or software to transform data.

Practice(s): IC3, CT7	**Disposition(s):** Critical Thinking

MS-DAT-DC-24: Analyze options to address a data quality issue.

Boundary Statement: Students should be able to identify common data quality issues, such as missing data and incorrect formatting, as well as investigate their causes. Students should also be able to develop multiple approaches for addressing data quality issues, such as deleting records with a lot of missing attributes, reformatting data that doesn't conform to expected values, or leaving the data as is. Students should evaluate the outcomes of each approach to addressing data quality and consider how removing data can introduce bias into a data set. Students should be focused on the human oversight of errors in data to prepare them to understand how decisions made by humans in the development of automation tools can impact others. Students can be provided with a data set to analyze.

Students are not expected to choose a "correct" approach for addressing data quality. Students should be focused on analyzing and evaluating solutions. Students are not expected to work with large, complex datasets, nor do they need to generate their own data with errors to analyze. Students are not expected to use automation tools to identify errors.

Practice(s): CT9, HCD11	**Disposition(s):** Critical Thinking, Curiosity

HS-DAT-DC-21: Use a computational tool to generate simulated data that fits certain parameters for use in a simulation.

Boundary Statement: Students should be able to generate simulated data using a computational tool. Students should be able to write a simple program or use a spreadsheet function to create a dataset that meets specific criteria (e.g., gener te a set number of data points within a specific range f values, create a certain distribution). For example, students could generate random initial velocities and launch angles within specified ranges to serve as inputs or a projectile motion simulation. Students should focus on creating data that can be used by a pre-existing simulation or model.

Students are not expected to build a simulation. Students are not expected to use advanced statistical methods or write complex algorithms to generate the data.

Practice(s): CT8, HCD12	**Disposition(s):** Critical Thinking, Persistence

HS-DAT-DC-22: Create a data dictionary that describes the name, type, and allowable values for each attribute and the logical relationships between variables in a dataset.

Boundary Statement: Students should be able to create a data dictionary for a dataset, including the names of all attributes (variables), their data types, and the range of acceptable values for each. The data dictionary should also describe the logical connections between variables, such as how a particular response on a survey question might lead to missing responses on other questions due to survey skip logic. For example, students could analyze a dataset of anonymized survey responses from a school and then create a data dictionary that lists each question as an attribute, defines the esponse format (e.g., integer, text string), and specifies the valid range f responses (e.g., a rating scale might allow the numbers 1–5 and "No Response" if the question was skipped).

Students are not expected to create the dataset or handle extremely large, complex datasets that require specialized tools or databases.

Practice(s): CT7, CT8	**Disposition(s):** Critical Thinking, Persistence

HS-DAT-DC-23: Use a computational tool to clean and organize text-based data.

Boundary Statement: Students should be able to use a computational tool to prepare text-based data for analysis. This includes handling inconsistencies (e.g., variations in capitalization, inconsistencies in spacing), correcting errors (e.g., misspellings, invalid entries), and structuring the data appropriately (e.g., separating combined fields, standa dizing formats). Students should identify when regular expressions can be used for pattern matching in text. Using a spreadsheet program or a simple programming script to convert inconsistent date formats, remove duplicate entries, or separate a single text column into multiple columns (e.g., first name and last name) within a d taset of survey responses is appropriate.

Students are not expected to build cleaning tools from scratch. Students are not required to handle large-scale, unstructured data like social media feeds.

Practice(s): CT9, HCD11	**Disposition(s):** Persistence, Critical Thinking

HS-DAT-DC-24: Evaluate different approaches to verifying consistency and compliance with expected data types, values, and ranges.

Boundary Statement: Students should be able to identify and evaluate different methods for ensuring that data is clean, accurate, and ready for use. This includes examining data for expected formats, values, and ranges. For example, students could evaluate different approaches for validating a dataset of customer birthdates to ensure all entries are in the correct format (e.g., MM/DD/YYYY) and fall within a reasonable range of years.

Students are not expected to write or implement scripts or use statistical models to perform data validation.

Practice(s): CT6, CT7	**Disposition(s):** Critical Thinking, Persistence

Data Investigation

EK-DAT-DI-09: Investigate a question that can be answered by collecting data in students' everyday environments.

Boundary Statement: Students should pose a simple, concrete question that can be answered by gathering data in familiar contexts. For example, "How many students ride the bus?" or "What is the most common favorite fruit in our class?" Then they should hold morning meetings to collect answers.

Students are not required to use computational tools. Students are not expected to design experiments or other types of research studies, use statistics, or analyze large datasets.

Practice(s): IC3, CT6	**Disposition(s):** Curiosity

E1-DAT-DI-09: Compare a question that can be answered through a data investigation and a question that can be answered through other means.

Boundary Statement: Students should distinguish between questions that require evidence gathered through counting or observing (data) and questions that are based on feelings, guesses, or creative ideas. For example, students should recognize that "How many students are wearing blue shirts today?" is a data question because we can count them and that "What is the best color in the world?" is a question of opinion that cannot be answered by counting.

Students are not expected to come up with complicated questions or do math that is beyond first grade expectations. Students are not required to analyze qualitative data versus quantitative data. The focus is on identifying whether a question needs proof from the real world to be answered.

Practice(s): IC3, CT6	**Disposition(s):** Critical Thinking, Curiosity

E2-DAT-DI-09: Develop a question that can be answered with data.

Boundary Statement: Students should brainstorm and refine a question th t can be answered with data. Students must understand that the best way to answer some questions is not through data collection. For example, a student might start with a vague question like "What colors do kids like?" and refine it to "Wh t is the most common favorite color in our class?" Students can develop questions related to counting and frequency, comparisons, interpretation of tally charts, bar graphs, pictographs, and making simple predictions and inferences.

Students are not required to use computational tools for data collection or creation of data representations. Students are not required to use statistical concepts beyond grade level (e.g., mean, median, mode, probability). Students are not required to create or interpret graphs with complex scales or work with large datasets.

Practice(s): IC3, CT6	**Disposition(s):** Critical Thinking, Curiosity

E3-DAT-DI-10: Investigate a data question involving relationships between multiple attributes.

Boundary Statement: Students should investigate how two or more attributes in a dataset relate to each other. Investigating the relationship between sunlight and water on the height of a plant, where students collect data on all three attributes and examine how height changes with different combinations of sunlight and water, is appropriate.

Students are not expected to perform statistical analysis. Students are not expected to work with more than two or three attributes. Students are not required to use digital equipment to collect or analyze data.

Practice(s): IC3, CT7	**Disposition(s):** Critical Thinking, Curiosity

E4-DAT-DI-10: Create an explanation that includes at least one data visualization to report the process and results of a data investigation.

Boundary Statement: Students should use age-appropriate, accessible computational tools (e.g., spreadsheets, presentation software, data visualization applications) to create a short written report, slide presentation, or other digital document that includes a data visualization to describe a data investigation and its results. The explanation should include a brief narrative that describes how they collected data and what the data shows, paired with an appropriate chart or graph (e.g., bar chart, line graph, pictograph). For example, students could investigate whether the class is meeting recycling goals and create a short presentation that includes how they weighed the recycling bins, a bar graph showing the weight of the bins over time, and a conclusion on whether the class is meeting its recycling goals based on the visual evidence.

Students are not expected to produce statistical analyses, interactive dashboards, or complex data visualizations. Students are not required to have knowledge of data visualization design principles beyond choosing a graph type that is appropriate for the data being presented.

Practice(s): IC3, CT8	**Disposition(s):** Creativity, Reflectivenes

E5-DAT-DI-10: Analyze a dataset to identify the nature and possible sources of variability in the data.

Boundary Statement: Students should identify patterns and variability in a dataset and distinguish between typical values and outliers. Students should look for trends or regularities in data and identify relationships between different variables. Understanding that data can vary from one observation to another is appropriate. For example, students could create a dataset using a timer to record multiple instances of how long it takes a marble to travel two meters and recognize that the times are not all the same, attributing the differences to possible error in clicking Start and Stop, differences in how the marble traveled (e.g., marble bumped the side of the track), and possible recording errors.

Students are not required to calculate statistical measures (e.g., mean, median, standard deviation) or perform statistical tests.

Practice(s): IC3, CT7	**Disposition(s):** Critical Thinking, Reflectivenes

MS-DAT-DI-25: Use a computational tool to identify relationships among variables in a dataset and make a classification or prediction.

Boundary Statement: Students should use spreadsheets or programming tools to identify relationships between different variables in a large dataset. Once a relationship is identified, it can be used to make a prediction. Students should critically analyze their predictions to see if they make sense for unknown conditions or input. Students should recognize that many artificial intelligen e systems work by identifying relationships and making predictions.

Students are not required to understand statistics beyond what is covered in sixth grade math standards.

Practice(s): IC4, CT7	**Disposition(s):** Reflectiveness, Critical Thinkin

MS-DAT-DI-26: Create data visualizations to show how different design choices can impact the interpretation of the same data.

Boundary Statement: Students should generate multiple visualizations for the same data to highlight a key insight they discovered. Students should critically analyze whether the different visualizations are equally clear and accurate, visually appealing, and accessible to all people, including those with visual or auditory disabilities. Students should also critically analyze the visualizations to see if they convey the same message despite different design choices. Students can use representations other than visual to communicate insights from data. Students can work with pre-existing data.

Students are not required to use computational tools, though they may be helpful.

Practice(s): HCD10, HCD11	**Disposition(s):** Creativity, Curiosity

MS-DAT-DI-27: Summarize a data investigation process, including potential biases, limitations, and supporting evidence.

Boundary Statement: Students should describe a data investigation, including the original question, how the data was collected and analyzed, and the key decisions made during the process. They can describe an investigation they completed individually or with a team or they can interview someone else and report on that person's process. Students should explain the motivation for the investigation, evaluate whether the results were expected or satisfactory, and identify potential sources of bias and limitations. They should clearly connect their conclusions to the evidence gathered during the investigation.

Students are not expected to collect actual data, but rather reflect on a p ocess that is already completed. Students are not required to use computational tools, but they may find them helpful

Practice(s): IC5, CT6	**Disposition(s):** Critical Thinking, Reflectivenes

HS-DAT-DI-25: Create a data visualization of a multivariate dataset to answer a question or make a classification or prediction.

Boundary Statement: Students should be able to use computational tools (e.g., spreadsheets, data analysis and visualization tools) to create data visualizations (e.g., scatter plots, line graphs, stacked bar charts, heat maps) that represent the results of their data investigations. For example, to represent the school's recycling habits, students could create a stacked bar chart showing the distribution of recycling material type (paper, plastic, metal) by school location (cafeteria, gym, classrooms) or time of day.

Students are not expected to use text-based programming or implement an algorithm or model to generate their visualization or make predictions. Students are also not required to conduct statistical analyses.

Practice(s): IC4, CT8	**Disposition(s):** Critical Thinking, Resourcefulness

HS-DAT-DI-26: Evaluate a data simulation or visualization to answer a data question, inform decision-making, and identify potential limitations.

Boundary Statement: Students should be able to evaluate the results presented in data visualizations and data simulations to determine their validity and usefulness in answering a question. Students should be able to identify potential biases, misleading representations, and limitations in the data or the model that could lead to different conclusions. For example, students could inspect a line graph showing daily air quality index values and identify whether the Y-axis starts at 0 or at a higher value, which could exaggerate changes in air quality. Students should focus on evaluating existing data products rather than creating them.

Students are not expected to create their own complex data simulations. Students are not required to rewrite or debug the code that generated the data visualizations or simulations.

Practice(s): ESR1	**Disposition(s):** Critical Thinking, Reflectivenes

Impacts of Data Science

EK-DAT-IM-10: Investigate how data can help a person make informed decisions in everyday life.

Boundary Statement: Students should explore everyday questions involving things they can count, observe, sort, and touch. Examples include household objects, classroom materials, and weather attributes. Students should compare quantities (e.g., less than, more than, equal to) and use patterns to make simple decisions based on data.

Students are not expected to work with large datasets. Students are not required to use statistical concepts. Students should not make decisions that are subjective and not based on data.

Practice(s): ESR1, CT7	**Disposition(s):** Reflectiveness, Curiosit

E1-DAT-IM-10: Examine a variety of data questions that address the needs of a person or community.

Boundary Statement: Students should examine data questions that are relevant to their lives (e.g., "How many students bring lunch versus buy lunch?" "What are the most common ways students get to school?"). Students should consider how a given data question might be relevant to or impact different people and communities and how minor changes to wording can affect how we interpret and answer the question (e.g., "How many first graders ead at home?" vs. "How many first graders have books at home?").

Students are not expected to generate data questions or answer the questions being examined. Students are not required to collect or analyze data.

Practice(s): ESR1, HCD10	**Disposition(s):** Critical Thinking, Curiosity

E2-DAT-IM-10: Distinguish between data collection approaches, including those that may lead to inaccurate or biased data.

Boundary Statement: Students should articulate how different ways of gathering data could affect different people or communities they are familiar with (e.g., classroom, family, school, neighborhood). Students should identify which communities might be unfairly represented or left out by specific data collection methods. Students should compare similar approaches to describe how they might produce different results. Students should connect these differences to real-world situations where biased data collection could lead to unfair decisions for their community.

Students are not expected to analyze data at this stage. Students are not required to collect data.

Practice(s): ESR1, HCD10	**Disposition(s):** Critical Thinking, Curiosity

E3-DAT-IM-11: Design a data collection process that addresses the needs of people from different backgrounds or groups.

Boundary Statement: Students should consider different data collection plans and how they would affect accuracy and representation. Students' plans should include what data to gather, which tools to use, and which sources to access for an investigation. Tools like class surveys or tally charts marking observations are appropriate. Students should identify different groups and describe how they would be affected by different data collection strategies. Students can obtain consent by including an opt-out box for the survey a student is designing. Students should learn that they have a right to tell authority figu es (e.g., teachers, website) that they don't want to share their information.

Students are not required to use advanced data collection techniques or statistics. Students are not required to consider the ethics associated with data collection or design a process for more than two or three variables.

Practice(s): ESR1, CT6	**Disposition(s):** Critical Thinking, Resourcefulness

E4-DAT-IM-11: Investigate how data collected about people may affect individuals and groups.

Boundary Statement: Students should investigate simple scenarios to identify who is affected by data collection and what the effect is, including concerns about privacy and fairness. Students should ask clarifying questions (e.g., "Who is collecting this information?" "What information are they collecting?" "Why do they need it?"). Students should identify who is affected, distinguishing between individuals and groups (e.g., the whole class). Students should describe what the effect is for the individuals and groups using simple, descriptive terms (e.g., "helpful," "harmful," "unfair"). Gathering facts and identifying clear effects rather than weighing pros and cons is appropriate.

Students are not expected to conduct formal research projects or make evaluative judgments about whether the data collection is overall good or bad. Students are not expected to understand the technical aspects of how data is collected online.

Practice(s): ESR1, ESR2	**Disposition(s):** Critical Thinking, Curiosity

E5-DAT-IM-11: Analyze the benefits and risks of a computing technology that uses collected data.

Boundary Statement: Students should examine everyday technologies that rely on collected data (e.g., weather apps, fitness trac ers, AI tools like voice assistants or recommendation systems) and identify both benefits (e.g., onvenience, personalization, improved predictions) and risks (e.g., privacy concerns, inaccurate results, or bias).

Students are not expected to understand technical details of algorithms, machine learning model training, or data storage infrastructure. Students are not required to evaluate statistical methods, encryption, or policy frameworks.

Practice(s): ESR1, ESR2	**Disposition(s):** Critical Thinking, Reflectivenes

MS-DAT-IM-28: Explain the benefits and risks of allowing personal data and metadata to be collected and used in datasets, including issues of data ownership, privacy, and sovereignty.

Boundary Statement: Students should be able to explain the benefits and risks f allowing personal data and metadata to be collected by websites, apps, and other computing technologies. Students should understand that benefits may include personali ed recommendations, improved services based on user data, or free access to platforms in exchange for data collection. Students should recognize risks including loss of privacy (e.g., location tracking, browsing history collection), unauthorized sale or sharing of personal data, and reduced control over how information is used (data ownership). Students should also understand basic concepts of data sovereignty, such as how data collected in one country may be stored or governed under another country's laws.

Students are not expected to understand technical data protection methods (e.g., encryption, secure protocols) or analyze specific legal frameworks (e.g., General D ta Protection Regulation [GDPR], the Children's Online Privacy Protection Act [COPPA]). Students are not required to create technical explanations of cybersecurity systems or evaluate the technical adequacy of privacy policies

Practice(s): IC3, IC5	**Disposition(s):** Critical Thinking

MS-DAT-IM-29: Analyze how decisions made at different stages of working with data can lead to biased data, misleading conclusions, and compromised AI models.

Boundary Statement: Students should be able to recognize that decisions made during data collection, data processing, data analysis, and data presentation all involve human choices and that those choices can, often unintentionally, introduce bias or distort results. Students should examine simple examples to identify how limitations or assumptions at different stages can affect interpretations and conclusions and how those same limitations may influen e the performance of AI models trained on the data. Students should be able to explain how these limitations shape conclusions and outcomes.

Students are not required to learn advanced statistical methods, the technical details of AI training algorithms, or programming-level solutions to bias. Students are not required to have deep knowledge of machine learning mathematics or data science processes.

Practice(s): ESR1, HCD12	**Disposition(s):** Critical Thinking

HS-DAT-IM-27: Evaluate the societal, environmental, and ethical implications of large-scale data collection and processing, including within AI applications.

Boundary Statement: Students should be able to analyze and articulate the societal, environmental, and ethical trade-offs associ ted with large-scale data collection and processing. This includes evaluating issues such as individual privacy and consent, data sovereignty, algorithmic bias, the spread of misinformation, and the environmental impact of data centers. For example, students could evaluate a social media platform's data collection policy and discuss the ethical implications of using that data for targeted advertising.

Students are not expected to perform a formal legal or economic analysis of data policies or otherwise focus on the technical or legal specifics

Practice(s): ESR2, CT6	**Disposition(s):** Critical Thinking, Reflectivenes

HS-DAT-IM-28: Debate the efficacy of a policy or regulation to ensure responsible data use.

Boundary Statement: Students should be able to analyze and evaluate the effectiveness of existing data policies and regulations. This could look like students researching a specific d ta privacy issue and preparing arguments for a classroom debate on whether a new law is needed to protect consumer data. Students should include the efficacy f social approaches (e.g., social movements, political resistance, personal behavioral change) aimed at shaping the design and use of computing systems.

Students are not expected to read the text of laws.

Practice(s): ESR1, CT6	**Disposition(s):** Critical Thinking, Reflectivenes

Systems & Security

Hardware & Software

EK-SYS-HW-11: Examine the use of tools to accomplish a task or solve a problem for different users.

Boundary Statement: Students should explore and talk about how everyday tools, both digital and non-digital, help people complete activities. Students should also recognize that different tools are designed to solve specific p oblems for different people. Age-appropriate examples include using a pencil or pen to write, using a paintbrush or drawing app to create a picture, using a magnifying glass to see small things, or using an app on a tablet or physical book to read a story.

Students are not expected to explain technical details about how the tools work or compare them based on efficien . Students do not need to develop knowledge of specialized hardware, software, or design processes.

Practice(s): IC3, HCD10	**Disposition(s):** Critical Thinking, Curiosity

E1-SYS-HW-11: Describe the purpose of basic hardware components of a computing system, using accurate terminology.

Boundary Statement: Students should use correct academic language to both identify the name and explain the basic function of external computing system hardware (e.g., laptop computer, tablet device, keyboard, mouse). Expressing that a laptop or tablet is used to run apps to complete school work or relating that headphones are used to hear sounds on a device is appropriate.

Students are not expected to identify or describe the function of internal computer system hardware. Students do not need to determine whether peripherals serve as input or output for a computing system.

Practice(s): IC3	**Disposition(s):** Curiosity, Critical Thinking

E2-SYS-HW-11: Explain how the basic hardware components of a computing system work together to perform input and output (I/O) operations.

Boundary Statement: Students should use academic language to name components of a computing system, identify whether they are input or output components, and explain how the components work together to take in information from the real world and produce an action that can be seen or heard by people. Explaining input and output operations of components like keyboards, monitors/screens, microphones and sound sensors, headphones, speakers, webcams/cameras, printers, light sensors, sound sensors, and distance sensors is appropriate.

Students are not expected to explain the technical details (e.g., I/O ports and buses, data digitization) of how inputs and outputs work. Students do not need to explain processing and storage.

Practice(s): IC3	**Disposition(s):** Critical Thinking, Curiosity

E3-SYS-HW-12: Describe the role of software in a computing system to accomplish tasks or solve problems.

Boundary Statement: Students should understand that software consists of programs and applications that tell hardware what to do and that different types of software serve different purposes (e.g., word processors for writing, web browsers for accessing information, games for entertainment, educational apps for learning). Students identify common software applications and describe how they process specific user inputs (e.g., typing, clicking) into meaningful outputs (e.g., a document, a game action) to complete tasks.

Students do not need to learn about operating system structures, cloud storage administration, or version control systems.

Practice(s): IC3, CT6	**Disposition(s):** Critical Thinking, Resourcefulness

E4-SYS-HW-12: Apply a basic troubleshooting process to identify and fix common hardware and software issues.

Boundary Statement: Students should focus on following a consistent set of steps (e.g., define, plan, try, reflect) to add ess common hardware and software issues such as a device not responding, no network access, or an app crashing. Students should apply basic troubleshooting strategies such as restarting the device, checking connections, or closing and reopening applications.

Students are not expected to solve all hardware and software issues they encounter. Students do not need to open devices to diagnose or troubleshoot internal hardware problems.

Practice(s): IC3	**Disposition(s):** Critical Thinking, Resourcefulness

E5-SYS-HW-12: Explain how hardware and software components of a computing system work together to perform input and output operations, processing, and storage.

Boundary Statement: Students should focus on everyday examples of how hardware and software interact to complete a task (e.g., pressing a key on a keyboard sends input to the computer, which is stored in memory and processed with software to display a letter on the screen as output). Age-appropriate models, such as flow diagrams, ole-play activities (e.g., students acting as a computer with input, processor, memory, and output), and physical computing projects that use sensors are appropriate.

Students are not expected to design or analyze actual logic gate diagrams, understand machine-level binary representations, or study complex computer architecture. Students do not need to develop knowledge of circuitry or number base systems.

Practice(s): IC3, CT6	**Disposition(s):** Reflectiveness, Critical Thinkin

MS-SYS-HW-30: Examine differences between computing systems based on user needs, system requirements, and potential societal, environmental, and ethical impacts.

Boundary Statement: Students should compare multiple computing systems based on features and system requirements (e.g., storage capacity, connectivity, accessibility features). Students should identify user needs (e.g., portability, cost, battery life) and provide reasoning for system choices. Students should connect at least one societal, environmental, or ethical impact to system choice (e.g., digital divide due to cost, energy consumption and e-waste, privacy and accessibility concerns).

Students are not expected to explain detailed technical specific tions (e.g., processor clock speeds, low-level architecture).

Practice(s): HCD10, HCD12	**Disposition(s):** Reflectiveness, Curiosit

MS-SYS-HW-31: Describe computing devices used in various industries, their basic functions, and how they are used to accomplish tasks or solve problems.

Boundary Statement: Students should learn about computing devices used in various industries and describe their basic functions. Devices might include robots, electronic control units in vehicles, medical imaging devices, automated manufacturing equipment, and agricultural sensors. Students should identify how these devices use input, output, processing, storage, and (when applicable) mechanical interactions to accomplish specific tasks or solve p oblems within industries such as medicine, agriculture, or manufacturing. Students should have agency in selecting industries and devices to study.

Students are not expected to build or program devices. Students do not need to demonstrate advanced technical fluency in ha dware or software integration.

Practice(s): IC3, CT6	**Disposition(s):** Reflectiveness, Curiosit

HS-SYS-HW-29: Differentiate an operating system as a special type of software that manages both the hardware and other software components of a computing system, including handling memory and peripherals.

Boundary Statement: Students should be able to differentiate an operating system from other types of software by identifying its role in managing hardware and software resources. For example, students might describe a computer's operating system, explaining that the operating system manages the computer's basic functions (e.g., allocating memory, connecting to peripherals).

Students are not expected to understand the intricacies of how an operating system is coded or to perform advanced tasks like configuring system drivers

Practice(s): IC3, CT7	**Disposition(s):** Critical Thinking

HS-SYS-HW-30: Demonstrate the capabilities and limitations of a physical or simulated computing device to address a task or problem.

Boundary Statement: Students should be able to select and use a computing device to solve a practical problem. For example, a student could use a device to manage project deadlines and task assignments for a group project, using its apps and cloud synchronization features to coordinate with team members while also identifying limitations, such as battery life or screen size.

Students are not expected to build their own computing device, understand the internal architecture of the device, or write code that directly controls the device's hardware components.

Practice(s): CT6, HCD12	**Disposition(s):** Critical Thinking, Resourcefulness

Security

EK-SYS-SE-12: Differentiate between public and private information.

Boundary Statement: Students should focus on being able to describe the differences between information that can be safely made public (e.g., favorite food, favorite animal) and information that should be kept private (e.g., home address, family photos). Real-life scenarios, like deciding what is okay to share with a friend versus with a stranger, are appropriate.

Students are not expected to use technical vocabulary like "confidential." Students do not need to understand detailed internet safety rules or the technical aspects of data privacy or cybersecurity.

Practice(s): ESR2, IC3	**Disposition(s):** Critical Thinking, Reflectivenes

E1-SYS-SE-12: Describe how to keep devices and online accounts safe from unauthorized access.

Boundary Statement: Students should focus on how authentication (e.g., passwords, QR codes for signing in) and good habits like signing out/logging off help eep their information protected online. Demonstrating how they sign out of devices and accounts and keep their authentication details secret from others is appropriate if students have online accounts that require authentication.

Students are not expected to create their own passwords or other authentication methods or to describe more complex authentication methods (e.g., multifactor authentication, passkeys, biometrics). Students do not need to have online accounts that require authentication. They can learn and describe these concepts without online accounts.

Practice(s): ESR2, IC5	**Disposition(s):** Critical Thinking, Reflectivenes

E2-SYS-SE-12: Explain how online actions have real-world consequences.

Boundary Statement: Students should focus on age-appropriate examples that connect how their online actions (e.g., sharing information, posting comments, using a device) can have effects in the real world (e.g., hurting someone's feelings, compromising personal safety). Writing a story or drawing a picture to show how a specific online behavior (e.g., sharing a pictu e without asking) can lead to a consequence (e.g., making a friend angry) is appropriate. Discussing the consequences of copying someone's work online is also appropriate.

Students are not expected to identify specific laws (e.g., Child en's Internet Protection Act, COPPA) or legal frameworks like copyright or intellectual property law. Students do not need to have online accounts. They can learn and describe these concepts without online accounts.

Practice(s): ESR2, IC3	**Disposition(s):** Sense of Belonging, Reflectivenes

E3-SYS-SE-13: Evaluate how sharing information online might reveal personally identifiable information and other details.

Boundary Statement: Students should focus on developing skills to help them recognize risks in online situations and assess what information to share online and when to share it. Activities using age-appropriate real-world situations (e.g., sharing their favorite book with a classmate, providing their address in response to a pop-up on a website saying they won a prize) are appropriate. Students should practice identifying how seemingly harmless information shared in contexts like online gaming (e.g., screen names, team names, game schedules) can unintentionally reveal PII to unintended recipients.

Students are not expected to evaluate all risky online situations, including things like catfishing and data breaches. Students do not need to participate in situations where they could reveal their personally identifiable i formation.

Practice(s): ESR2, IC3	**Disposition(s):** Critical Thinking, Reflectivenes

E4-SYS-SE-13: Distinguish between authentication and authorization in protecting devices and private information.

Boundary Statement: Students should focus on authentication and authorization as two distinct but related concepts for protecting private information and devices. Using an analogy of a house is appropriate: authentication is proving you have the key to get in the door, whereas authorization is a parent telling you which rooms in the house you're allowed to enter once you're inside.

Students are not expected to learn about different types of authentication (e.g., biometrics, multifactor authentication) or technical details of how authentication and authorization work. Students do not need to use a variety of authentication methods.

Practice(s): IC3	**Disposition(s):** Critical Thinking, Reflectivenes

E5-SYS-SE-13: Describe the concepts of the CIA triad and how each component is important in protecting information.

Boundary Statement: Students should be able to explain the CIA triad at a conceptual level and describe how each element (confidentialit , integrity, and availability) helps protect information. For example, students might describe confidentiality as eeping a password secret and not sharing personally identifiable i formation online. They might describe integrity as ensuring that data is accurate and unchanged (e.g., checking whether a photo is real or AI-generated, ensuring that grades in a gradebook haven't been improperly changed). They might describe availability as ensuring that they can access their accounts, devices, apps, or gaming systems when they need them (e.g., the system isn't down or blocked).

Students are not expected to implement technical security measures such as encryption or access control lists. Students do not need to understand cybersecurity systems or networks beyond how they affect students' daily lives.

Practice(s): IC3, HCD12	**Disposition(s):** Critical Thinking, Reflectivenes

MS-SYS-SE-32: Explain the effects of not using the CIA triad when working with data.

Boundary Statement: Students should recognize and explain what happens when each element of the CIA (confidentialit , integrity, and availability) triad is not maintained. When confidentiality is compromised, sensitive data is exposed (e.g., leaked passwords, shared personal information). When integrity is compromised, data becomes corrupted or altered (e.g., incorrect grades in a school database). When availability is compromised, systems become inaccessible (e.g., a denial-of-service attack blocking a website). Students should apply the CIA Triad to age-appropriate, relatable contexts (e.g., school accounts, social media, online gaming), demonstrating understanding of how failures connect to real-world impacts.

Students are not expected to conduct professional-level security audits or develop deep cryptography knowledge.

Practice(s): IC3, HCD12	**Disposition(s):** Critical Thinking, Reflectivenes

MS-SYS-SE-33: Evaluate common types of cyber attacks and preventions.

Boundary Statement: Students should identify and describe common types of cyber attacks, such as phishing, malware, ransomware, and social engineering. They should be able to evaluate how these attacks exploit human behavior and system vulnerabilities. Students could also explore real-world examples of cyber incidents and their consequences. Students should learn to apply basic prevention strategies, such as using strong passwords, enabling multifactor authentication, and recognizing suspicious activity.

Students are not expected to investigate complex cyber attacks or use professional security tools. They do not need to break into systems, test network defenses, or analyze how malware is built. Students are also not expected to configu e advanced security settings or understand technical protocols used in industry.

Practice(s): IC3, CT6	**Disposition(s):** Critical Thinking, Resourcefulness

HS-SYS-SE-31: Identify different types of cybersecurity and physical security measures and the trade-offs for users, data, and devices.

Boundary Statement: Students should be able to identify and describe different types of cybersecurity and physical security measures, as well as their associated trade-offs. They should be able to articul te how a security measure may benefit one aspect (e.g., d ta protection) while posing a challenge to another (e.g., user convenience, cost). For example, students could identify how multifactor authentication enhances security for a user's account, but makes it less convenient for the user to access. Students should focus on conceptual understanding of security principles and their practical implementation.

Students are not expected to implement these security measures or demonstrate technical expertise.

Practice(s): ESR2, CT6	**Disposition(s):** Critical Thinking, Reflectivenes

HS-SYS-SE-32: Classify the causes and impacts of security breaches and social engineering attacks for individuals, industries, communities, and governments.

Boundary Statement: Students should be able to research and categorize the consequences of a security breach or social engineering attack, differentiating between the effects on various stakeholders. The classific tion should go beyond a simple list of impacts and demonstrate an understanding of how the same breach can have different types of consequences (e.g., financial, reputational, legal, operational) for different groups. For example, a student could classify the impacts of a data breach at a hospital, analyzing the effects on patients (individuals), the hospital (industry), the local healthcare system (community), and regulations (government). Students should focus on understanding the broad, interconnected impacts of an attack.

Students are not expected to conduct a forensic investigation of a real-world breach or to use professional cybersecurity tools for their analysis. Students do not need to focus on the technical details of the attack itself.

Practice(s): ESR2, CT6	**Disposition(s):** Critical Thinking, Reflectivenes

HS-SYS-SE-33: Formulate a solution to a security flaw in a given system.

Boundary Statement: Students should be able to analyze a simulated or described scenario involving a security flaw and p opose a high-level solution. The proposed solution should explain the identified vulnerability, the potential consequences, and the recommended steps for mitigation. For example, a student could be given a case study of a school's network being compromised by a phishing attack and be asked to propose a solution that includes both technical measures (e.g., email filtering th t detects suspicious links, multifactor authentication) and nontechnical measures (e.g., user training, security policies) to prevent future attacks.

Students are not expected to implement the solution.

Practice(s): ESR1, CT6	**Disposition(s):** Critical Thinking, Resourcefulness

Networks

Standards in this subconcept progress from PK/K–Grade 2 standards in the Security subconcept.

E3-SYS-NT-14: Explain how people access the internet to gain information and communicate with each other.

Boundary Statement: Students should be able to describe, in age-appropriate terms, how people use different devices to connect to the internet and communicate. Examples of devices that can be used to access the internet include computers, cell phones, and tablets. An example of accessing information is using a search engine to look up facts for a report. An example of communicating through technology is using video chat to talk with people across long distances.

Students are not expected to describe technical infrastructure such as IP addressing, routing protocols, domain name systems, or data packet transmission. Students do not need to consider network configur tion or errors.

Practice(s): ESR1, IC3	**Disposition(s):** Resourcefulness, Critical Thinking

E4-SYS-NT-14: Compare wired and wireless methods that computing devices use to connect to the internet.

Boundary Statement: Students should focus on comparing the advantages and disadvantages of different ways of connecting to the internet. Important factors include distance, speed, convenience, and safety considerations (e.g., connecting to trusted networks at home or school vs. unknown public networks). Tasks in which students determine when it is advantageous to have either a wired connection or a wireless connection are appropriate.

Students are not expected to explore other ways of connecting between devices to share information (e.g., Bluetooth) or the protocols used to connect devices (e.g., TCP/IP). Students do not need direct access to routers or hubs.

Practice(s): IC3	**Disposition(s):** Critical Thinking, Reflectivenes

E5-SYS-NT-14: Distinguish between the components of wired and wireless networks.

Boundary Statement: Students should explore and describe the basic components of networks, such as devices (e.g., computers, tablets, phones), connecting hardware (e.g., routers, switches, cables), and wireless connections (Wi-Fi signals).

Students are not expected to understand detailed networking protocols, IP addressing, or the technical differences between LANs, WANs, and the internet. Students do not need to understand packet structures, routing tables, or advanced network design.

Practice(s): IC3, CT7	**Disposition(s):** Reflectiveness, Curiosit

MS-SYS-NT-34: Model how information in a network is broken down into packets, transmitted between devices, and reassembled.

Boundary Statement: Students should be able to model how data is split into packets, encoded into binary, transmitted independently across a network, and reassembled at the destination. Students should also identify common issues such as packet loss, explain possible causes (e.g., congestion, interference, faulty hardware), and describe basic strategies to reduce or prevent packet loss.

Students are not expected to explore networking protocols in detail (e.g., TCP/IP) or to configu e real-world networking hardware (routers, switches, servers).

Practice(s): CT6, CT7	**Disposition(s):** Critical Thinking, Resourcefulness

MS-SYS-NT-35: Explain how the resilience of the internet depends on interconnected devices and their roles and functions within the network.

Boundary Statement: Students should be able to explain how the internet is made up of interconnected devices that each serve specific oles. For example, routers direct traffic betwee networks, servers host information, clients (e.g., laptops, phones) request and use data, and switches connect devices within a local network. Students should also describe how redundancy (multiple pathways between devices) supports resilience, allowing communication to continue even if some devices or connections fail.

Students are not expected to demonstrate technical knowledge of networking protocols (e.g., TCP/IP) or to configu e or analyze real-world enterprise-level networking systems.

Practice(s): IC3, CT6	**Disposition(s):** Curiosity, Critical Thinking

HS-SYS-NT-34: Diagram a network of computing systems, including hardware and software.

Boundary Statement: Students should be able to create a diagram of a computing systems network that includes both hardware (e.g., servers, routers, storage, user devices) and software (e.g., operating systems, applications). Students should focus on recognizing and representing the concept of a network and the role of each component in the network. Students should show how these components connect and interact to process information, accomplish tasks, or solve problems.

Students are not expected to build a network using hardware or software.

Practice(s): CT7, CT8	**Disposition(s):** Creativity, Critical Thinking

HS-SYS-NT-35: Analyze how the internet functions as a network of networks and how it differs from other types of networks.

Boundary Statement: Students should be able to conceptually analyze the internet's layered structure and its role as a global network connecting smaller, local networks. They analyze its layered structure, key components (e.g., routers), core protocols (e.g., TCP/IP), and the client/server model. For example, a student could compare a simple LAN in a classroom to the global structure of the internet in terms of structure, scalability, addressing, and data transfer. Students should focus on developing a high-level, conceptual understanding of networking principles.

Students are not expected to implement specific networks, onfigu e or troubleshoot professional-grade networking equipment, write network protocols, or perform packet analysis.

Practice(s): CT7	**Disposition(s):** Critical Thinking, Resourcefulness

Impacts of Computing Systems

EK-SYS-IM-13: Identify responsible behavior when using computing systems and digital tools.

Boundary Statement: Students should be able to identify the responsible use of hardware and software in terms of tangible, classroom-level expectations, including caring for hardware (e.g., handling devices gently, keeping food and drink away from devices), making safe choices (e.g., asking permission before using a device, using only the app specified), and being a good citi en in a shared technology environment (e.g., not using someone else's account, using headphones instead of playing sound out loud). Students should focus on understanding that computing systems in schools are tools for learning and should be used appropriately, in the same way other tools are used in the classroom.

Students are not expected to understand complex cybersecurity or digital citizenship topics.

Practice(s): ESR1, IC3	**Disposition(s):** Reflectivenes

E1-SYS-IM-13: Describe an individual's role in responsibly using computing systems and digital tools.

Boundary Statement: Students should be able to explain why rules for responsible use of computing systems and tools exist and why following them is important for themselves and their classroom community. They should describe their individual role in meeting these expectations and explain both individual and classroom consequences when expectations are not followed. For example, students could explain that careful handling of hardware (e.g., using devices gently, keeping food and drinks away) ensures devices remain available for everyone and that being a good digital citizen in a shared technology environment (e.g., not using someone else's account, using headphones instead of playing audio aloud) helps maintain a safe and respectful learning space.

Students are not expected to describe complex digital citizenship topics. Students do not need to understand intellectual property (e.g., copyright), plagiarism, how to identify misinformation, or the nuances of a digital footprint.

Practice(s): ESR1, IC3	**Disposition(s):** Reflectiveness, Sense of Belonging

E2-SYS-IM-13: Describe the benefits and harms that arise from an individual's use of computing systems and digital tools.

Boundary Statement: Students should describe benefits and harms in the ontext of their own personal experiences, feelings, and well-being. The concepts should be simple and concrete. Describing benefits such as learning new things (e.g., w tching a video about animals), having fun (e.g., playing a creative game), and connecting with family and friends (e.g., video-calling a relative) are appropriate. Describing harms such as the physical effects of too much screen time (e.g., tired eyes, feeling grumpy), the emotional effects (e.g., feeling sad or left out by something they see, accidentally seeing something confusing or scary) are also appropriate.

Students are not expected to describe complex or large-scale societal harms. Students do not need to understand issues like the digital divide, the spread of misinformation, or data privacy.

Practice(s): ESR1, IC3	**Disposition(s):** Critical Thinking, Reflectivenes

E3-SYS-IM-15: Describe how widely used computing systems may impact an individual's life and community.

Boundary Statement: Students should be able to describe both benefits and harms (e.g., e ects on safety, privacy, economy, culture) of computing systems in wide use. They should be able to reason about specific s enarios and identify which individuals or communities (e.g., themselves, their parents, children, elderly people, teachers) are impacted by a particular benefit or harm and describe the nature of the impacts.

Students are not expected to describe the benefits and harms f computing systems or the impacts on communities without first learning about these technologies and ommunities. Students do not need to memorize a list of technologies and their impacts on specific ommunities.

Practice(s): ESR1, HCD12	**Disposition(s):** Reflectiveness, Critical Thinkin

E4-SYS-IM-15: Investigate the impacts of widely used computing systems on natural resources and the environment.

Boundary Statement: Students should investigate the environmental effects of widely used computing systems (e.g., artificial intelligen e, computers, tablets, monitors). They should designate their effects as harmful or beneficial to Earth's envi onment and its natural resources. Students can consider large sources of impact (e.g., not disposing of e-waste properly, burning fossil fuels for power, mining of materials, modeling of solutions using technology, conserving energy with "smart" systems) and individual sources of impact (e.g., water and energy usage, frequent device upgrades, recycling/reuse of computing systems and components, information availability about the environment, reduction in paper usage).

Students are not expected to investigate the impacts of computing systems that are not in wide use (e.g., quantum computing). Students do not need specialized equipment or knowledge of science concepts beyond what is grade-level appropriate.

Practice(s): ESR1, ESR2	**Disposition(s):** Critical Thinking, Reflectivenes

E5-SYS-IM-15: Examine how computing systems impact culture and the ways people live and work.

Boundary Statement: Students should examine the cultural impact of existing and emerging computing systems that both help and hinder societal needs. Examples of appropriate topics include the different roles social media plays in our culture and society and how generative AI is changing the way people work.

Students are not expected to identify or reference specific ethical frameworks (e.g., utilitarianism) or sociological concepts (e.g., functionalism). Students do not need to access or use the technologies being discussed (e.g., social media, generative AI).

Practice(s): ESR1, HCD12	**Disposition(s):** Critical Thinking, Reflectivenes

MS-SYS-IM-36: Collaborate to improve the design of a computing system to meet the needs of diverse users.

Boundary Statement: Students should recognize that computing systems need to be designed with accessibility and inclusivity in mind so they can be better used by people with different needs, abilities, and ways of thinking. Examples include providing options for closed captions, adjusting font sizes or colors for readability, providing alternative input devices (like voice control or adaptive keyboards), and including features that support the needs of neurodiverse users. Students should collaborate in teams to propose and refine designs th t make a system more inclusive. Students may conduct simple peer testing to evaluate design improvements.

Students are not expected to design hardware from scratch or write complex code for accessibility features. Students do not need to perform professional-level usability testing.

Practice(s): IC5, HCD12	**Disposition(s):** Creativity, Sense of Belonging

MS-SYS-IM-37: Examine how access to computing systems can vary based on personal and social factors, such as physical ability, geographic location, socioeconomic status, and age.

Boundary Statement: Students should investigate how access to computing systems varies across groups and regions, considering factors like income, physical accessibility, community resources, and infrastructure. For example, students may compare access to broadband internet in urban versus rural areas or explore how adaptive technologies support people with disabilities. Students should analyze examples of digital divides at the personal (home access), the school (device availability), and the community (e.g., libraries, Wi-Fi hotspots) levels. Students may discuss the role of government or nonprofits in expanding a cess.

Students are not expected to perform statistical analyses of large national datasets, but they may use pre-curated datasets, maps, or local surveys. Students do not need to perform advanced policy evaluation.

Practice(s): IC3, HCD10	**Disposition(s):** Critical Thinking, Curiosity

HS-SYS-IM-36: Evaluate the rationale behind a law or policy governing the design and use of computing systems.

Boundary Statement: Students should be able to analyze and articulate the reasoning (technical, legal, and social) behind various policies related to the design and use of computing systems. Students should consider the rationale for technical approaches (e.g., responsible design practices, policies requiring two-factor authentication for sensitive data), legal approaches (e.g., COPPA) and social approaches (e.g., social movements, political resistance, personal behavioral change) to shaping the design and use of computing systems. Students should focus on understanding why these policies exist, the problems they aim to solve, and their intended and unintended consequences.

Students do not need to read and interpret the text of laws or other legal texts.

Practice(s): IC3, HCD10	**Disposition(s):** Critical Thinking, Reflectivenes

HS-SYS-IM-37: Investigate how computing systems and infrastructure impact society and the environment, identifying who is affected and why.

Boundary Statement: Students should research, analyze, and articulate the societal and environmental impacts of computing systems, including the lifecycle of physical components. This includes, for example, examining energy consumption of data centers, the ethics of electronic waste storage and disposal, the social effects of mobile devices, and the supply chains behind computing technologies. Students should identify which communities are most affected by these impacts and explain why, considering factors such as geography, labor conditions, access to resources, and environmental burden. For example, a student could investigate the supply chain of a mobile device, including its human and environmental costs, and analyze how those costs may disproportionately affect certain workers or communities.

Students are not expected to propose novel technical solutions to these large-scale problems, but rather to analyze and communicate the problems themselves.

Practice(s): ESR1, HCD10	**Disposition(s):** Critical Thinking, Reflectivenes

Computing & Society

History of Computing

EK-SOC-HI-14: Identify computing technologies used in daily life that have changed over time.

Boundary Statement: Students should recognize and recall basic visual differences of computing technologies used in school or by their own family members that have changed from older versions. Identifying that the current family phone is smaller and fl tter than an older phone a parent or grandparent used or that the family watches movies on a tablet now instead of using a TV are appropriate.

Students are not expected to understand the internal technological advancements (e.g., faster processors, memory capacity) that caused the changes. Students are not required to know specific historical dates, names of inventors, or the technical functions of the different generations of devices.

Practice(s): ESR1, IC3	**Disposition(s):** Reflectiveness, Curiosit

E1-SOC-HI-14: Compare how an everyday activity changed after a specific computing technology was introduced.

Boundary Statement: Students should compare observable differences in how people completed a familiar task before and after a computing technology was introduced. Concrete, everyday examples, such as how people communicate (e.g., writing letters vs. sending text messages) or find i formation (e.g., using library books vs. searching online) are appropriate.

Students are not expected to understand the technical details of how technologies work or their historical timelines. Students are not required to analyze societal shifts or economic impacts.

Practice(s): ESR1, IC3	**Disposition(s):** Reflectiveness, Curiosit

E2-SOC-HI-14: Analyze the ways that people from different cultures, backgrounds, and time periods have designed computing technologies to help them solve problems and express ideas.

Boundary Statement: Students should explore how computing technologies have emerged from people's efforts to meet specific needs and exp ess ideas within their geographic, cultural, and historical contexts. An analysis focused on how computing technologies have supported individuals with different abilities throughout history or changed the modes and frequency of communication is appropriate.

Students are not expected to conduct detailed history on the evolution of computing as a whole. Students are not required to compare complex cultural or societal impacts of technology across time periods.

Practice(s): ESR1, HCD10	**Disposition(s):** Sense of Belonging, Reflectivenes

E3-SOC-HI-16: Examine how computing innovations have changed the ways people live, work, or communicate over time.

Boundary Statement: Students should explore and explain how computing innovations across different time periods have progressively changed daily life, work, or communication, focusing on concrete, observable changes in human behavior. Examining how finding i formation, communicating verbally and in writing, enjoying entertainment, shopping, doing tasks at school and work have evolved over time due to computing innovations are appropriate.

Students are not expected to understand the technical details of how computing innovations work or to create timelines of all computing innovations. Students are not required to examine complex societal impacts of computing innovations.

Practice(s): ESR1, IC3	**Disposition(s):** Critical Thinking, Reflectivenes

E4-SOC-HI-16: Investigate the contributions of diverse individuals and communities in the history of computing.

Boundary Statement: Students should explore stories and achievements of individuals and communities whose contributions to computing are often underrepresented or overlooked in mainstream accounts. Students should focus on understanding the concept of diversity or its lack in computing history. Researching and explaining the contributions of women, people of color, and innovators from global communities is appropriate.

Students are not expected to explore the factors that led to historical underrepresentation or exclusion in computing. Students are not required to memorize a comprehensive list of every historical figu e.

Practice(s): ESR1, HCD10	**Disposition(s):** Sense of Belonging, Curiosity

E5-SOC-HI-16: Analyze how the inclusion or exclusion of diverse individuals and communities has shaped the design, development, and societal impact of computing technologies.

Boundary Statement: Students should analyze historical and contemporary examples of computing innovations, identifying how inclusion or exclusion of certain groups influen ed technological progress and access. Analyzing the development of early programming languages or accessibility tools is appropriate.

Students are not expected to conduct independent historical research or examine complex sociopolitical movements in depth. Students are not required to know specific d tes, laws, or named individuals beyond those directly connected to major computing milestones.

Practice(s): ESR1, HCD10	**Disposition(s):** Critical Thinking, Curiosity

MS-SOC-HI-38: Compare the roles of individuals, communities, organizations, and governments in shaping computing technologies across major eras in computing history.

Boundary Statement: Students should identify and describe examples of historical and contemporary individuals, communities, organizations, and governments that have influen ed the development of computing technologies. Students explain the motivations of these individuals and groups in advancing computing. Students also recognize how these different actors have interacted with one another. Students compare the different roles these groups have played (e.g., designing, funding, regulating, using technologies) and recognize that these actors sometimes work together or influen e one another.

Students are not expected to memorize timelines, technologies, or related facts. They are not expected to conduct deep ethical or social analyses. Students are also not expected to evaluate or rank individuals or groups by their relative importance.

Practice(s): IC3, HCD10	**Disposition(s):** Critical Thinking, Curiosity

MS-SOC-HI-39: Analyze intended and unintended impacts of a historical computing technology on society and the environment.

Boundary Statement: Students should distinguish between the intended purpose of a computing technology and its unintended consequences. Students identify and describe clear societal and environmental impacts, such as changes in communication, employment, privacy, e-waste, and energy consumption. They make simple cause-and-effect connections between computing technologies and their outcomes.

Students are not expected to conduct cost-benefit analyses or per orm complex calculations to determine environmental impacts. They are also not expected to make detailed predictions about future technologies or develop solutions to the unintended effects they identify.

Practice(s): ESR1, IC5	**Disposition(s):** Critical Thinking, Reflectivenes

HS-SOC-HI-38: Analyze the historical trajectory of a specific computing technology and how its development is linked to societal and environmental factors.

Boundary Statement: Students should be able to investigate a computing technology (e.g., the internet, a social media platform, a specific type f integrated sensor technology) from its inception to its current state, identifying key moments where nontechnical forces directly influen ed its design, adoption, or ethical challenges. They should use reliable secondary sources to draw and support their analytical conclusions. For example, high school students could research the rise of mobile computing, analyzing how shifts in economic factors and social trends drove its rapid adoption and evolution.

Students are not expected to conduct or synthesize research on the primary source documents of these technologies, nor are they required to master economic models or complex legal frameworks related to technology regulation.

Practice(s): ESR1, IC3	**Disposition(s):** Critical Thinking, Reflectivenes

HS-SOC-HI-39: Propose modifications to an existing policy or piece of legislation that encourages ethical innovation and minimizes societal risks associated with technology.

Boundary Statement: Students should be able to articulate the rationale behind policies and legislation that guide technological development. This includes explaining how specific laws, li e data privacy regulations and rules about algorithmic transparency, are designed to address societal concerns such as discrimination, a loss of privacy, or job displacement. Students should focus on understanding the reason behind technology policy rather than on the process of lawmaking. For example, a student could justify the need for legislation like the European Union's GDPR by explaining how it gives individuals more control over their personal data, thereby minimizing the risk of privacy breaches and misuse.

Students are not expected to draft legal documents, interpret complex legal jargon, or have a deep understanding of the legislative process itself.

Practice(s): ESR1, ESR2	**Disposition(s):** Critical Thinking, Reflectivenes

Emerging Technologies

Standards in this subconcept progress from PK/K–Grade 2 standards in the History of Computing subconcept.

E3-SOC-ET-17: Describe how new technologies create both benefits and risks in personal and family life.

Boundary Statement: Students should describe how new technologies (both those they interact with on a daily basis and those they may have learned about in other ways) may not have an exclusively positive or negative impact on their personal or family life. They should consider examples of both benefits and risks. Examples such as video calls with elatives, online learning, AI technologies, or using smart devices at home are appropriate.

Students are not expected to describe complex social or economic impacts nor complex new technologies (e.g., quantum computing).

Practice(s): ESR1, IC3	**Disposition(s):** Reflectiveness, Critical Thinkin

E4-SOC-ET-17: Analyze how the limitations of existing technologies can lead to emerging technologies.

Boundary Statement: Students should be able to identify a limitation in an older or existing technology (e.g., a weakness, something that doesn't work well, a problem) and logically explain how overcoming that limitation could motivate the creation of a new, emerging technology, focusing on cause-and-effect reasoning in innovation. Analyzing both past (e.g., mainframe computers to laptops, older mobile phones to smartphones) and more current emerging technologies (e.g., auto-complete to generative AI, older model cars to self-driving cars) is appropriate.

Students are not expected to research the actual motivations for the creation of technologies.

Practice(s): ESR1, HCD10	**Disposition(s):** Reflectiveness, Curiosit

E5-SOC-ET-17: Examine how people decide whether or not to use emerging technologies.

Boundary Statement: Students should analyze the factors people consider when making personal decisions about using, avoiding, or rejecting emerging technologies, including personal needs and values, benefits, oncerns, and accessibility. For example, students might examine why a person would choose to use or reject a smart watch or voice assistant at home by considering convenience, cost, privacy concerns, and accessibility for family members with varying abilities.

Students are not expected to do research studies or understand the technical aspects of how the emerging technologies work. Students are not required to make value judgments about whether people's decisions are "right" or "wrong."

Practice(s): ESR1, IC3	**Disposition(s):** Reflectiveness, Curiosit

MS-SOC-ET-40: Evaluate when it is appropriate to use AI and other emerging technologies to solve a problem based on their capabilities, limitations, and environmental impacts.

Boundary Statement: Students should be able to identify what emerging technologies (e.g., AI, robotics, virtual reality, quantum computing) can and cannot currently do and determine whether they are appropriate for addressing a specific p oblem. They should consider factors such as technical capabilities, reliability, infrastructure requirements, environmental impacts, and the maturity or readiness of the technology.

Students are not expected to understand the underlying mechanics of these technologies, predict future breakthroughs, or design functioning prototypes. They are also not expected to engage in ethical debates about regulation or human rights implications.

Practice(s): ESR1, HCD10	**Disposition(s):** Reflectiveness, Critical Thinkin

MS-SOC-ET-41: Evaluate how the decisions made while designing an emerging technology influence user experiences differently across different communities.

Boundary Statement: Students should be able to identify and evaluate examples of how design decisions in new and emerging technologies influen e user experiences across different communities. They analyze how selected design choices (e.g., accessibility, language, cost, connectivity, geographic context, device availability) can create benefits or barriers or different groups of users. Students may focus on one or more of these factors rather than addressing all dimensions simultaneously.

Students are not expected to conduct usability studies, apply design frameworks, or create prototypes.

Practice(s): ESR2, IC3	**Disposition(s):** Reflectiveness, esourcefulness

MS-SOC-ET-42: Debate ways an emerging technology impacts the social, cultural, and environmental issues in local communities.

Boundary Statement: Students should be able to assess the potential or current impact of emerging technologies to address local community needs. Focusing on a specific eme ging technology, students should be able to construct arguments with references for a debate on positive and negative impacts of that technology. Students could either debate which applications of an emerging technology have the greatest or least impact or argue for or against a particular application. Students should pay attention to who is impacted and how they are impacted as well as to costs and harms to the community and environment.

Students are not expected to understand all the technical details of emerging technologies.

Practice(s): ESR1, CT8	**Disposition(s):** Critical Thinking, Reflectivenes

HS-SOC-ET-40: Evaluate the fundamental technological differences between an emerging technology and established technologies and how those differences influence computing.

Boundary Statement: Students should be able to evaluate the technological advances that distinguish an emerging technology from existing technologies and how they enable features or address limitations in current systems. For example, quantum computing uses superposition and entanglement to represent and manipulate information more compactly than classical systems. This allows certain data structures and computations to be expressed with fewer physical resources.

The degree to which students are introduced to a technical concept is dependent on the technology and the backgrounds of the students. For example, quantum operations would be reasonable, but not necessary, to introduce, but the introduction of the quantum mechanics underlying superposition and entanglement is not expected.

Practice(s): ESR1, IC3	**Disposition(s):** Critical Thinking, Reflectivenes

HS-SOC-ET-41: Evaluate the societal and environmental impacts of an emerging technology, including those that lead to inequities in access and outcomes.

Boundary Statement: Students should be able to critically analyze an emerging technology by identifying and assessing its potential positive and negative effects on society and the environment. This includes considering how the technology might create or worsen social inequities in access and outcomes. Students should focus on developing a well-reasoned, conceptual argument about an emerging technology's broader impacts. For example, a student could evaluate the use of generative AI in education. They would identify not only the potential positive impacts, like providing personalized tutoring, but also the negative ones, such as the potential for academic dishonesty and the risk of widening the digital divide for students without access to these tools.

Students are not expected to conduct a formal, data-driven sociological or environmental study.

Practice(s): HCD10, HCD12	**Disposition(s):** Critical Thinking, Reflectivenes

HS-SOC-ET-42: Design a conceptual solution to a real-world problem using an emerging technology, analyzing its potential benefits and harms.

Boundary Statement: Students should be able to design a conceptual or tangible solution to a real-world problem by applying an emerging technology. This process must be supported by credible research on both the problem and the technology's capabilities. The solution should also be mindful of ethical implications, with students analyzing the potential benefits and harms to di erent groups of people and the environment. For example, a student could create a design document for a system that uses AI to analyze satellite imagery to help track deforestation, addressing a real-world environmental problem. Their work would include research to support their claim and a discussion of the ethical considerations, such as the potential for misuse of the data.

Students are not expected to build a fully functional, production-ready version of their solution. Students are not expected to engage in complex engineering.

Practice(s): ESR1, CT6	**Disposition(s):** Creativity, Critical Thinking

Humans & Computing

EK-SOC-HU-15: Explain that people design and develop computing technologies.

Boundary Statement: Students should recognize and express that the computing technologies they use every day (e.g., tablets, apps, programmable toys) do not appear by magic, but are created by people to help solve problems or for entertainment. For example, during a classroom discussion about a favorite game, a student might explain that a person had to think of the rules, draw the characters, and program the game before they could play it.

Students are not expected to understand the technical details of how technologies are built. Students are not required to build their own functioning electronic computing technologies.

Practice(s): IC3, HCD10	**Disposition(s):** Sense of Belonging, Curiosity

E1-SOC-HU-15: Differentiate between activities that humans do well and activities that computing technologies do well.

Boundary Statement: Students should categorize and explain simple, observable differences in what humans and computing technologies do best for various tasks. Examples of human strengths, such as creativity, understanding feelings, and solving new problems, are appropriate. Examples of computing technology strengths, such as repeating tasks without mistakes, calculating quickly, and remembering lots of information, are also appropriate.

Students are not expected to understand the cognitive science behind human intelligence or the technical details of how computing technologies perform certain tasks. Students are not required to analyze ethical or societal consequences that arise from humans or machines performing specific tasks

Practice(s): IC3, HCD10	**Disposition(s):** Critical Thinking, Curiosity

E2-SOC-HU-15: Investigate situations where humans have created computing technologies to solve problems.

Boundary Statement: Students should investigate familiar, concrete examples of how people have designed computing technologies to address everyday problems they can relate to. Technologies with clear, observable purposes are appropriate, such as traffic lights t t help cars and pedestrians move safely, barcode scanners that help stores track inventory, automatic doors that help people enter buildings, weather apps that help families plan activities, or classroom timers that help manage transitions. Students might investigate how these technologies were created to solve specific problems, like "cars were crashing at intersections" or "it was hard to keep track of library books."

Students are not expected to understand how computing technologies work internally or to explain programming, engineering, or data processes. Students are not required to analyze historical development, compare technologies, or address advanced issues such as data privacy, cybersecurity, artificial intelligen e, or automation.

Practice(s): ESR1, HCD10	**Disposition(s):** Sense of Belonging, Critical Thinking

E3-SOC-HU-18: Examine why people design and build computing technologies.

Boundary Statement: Students should explore the basic reasons computing technologies were created and communicate the needs or problems these technologies were likely intended to address, focusing on the human intent or goal. Examining how computing technologies, such as AI systems, are designed to meet needs, solve specific p oblems, make tasks easier, or provide entertainment is appropriate.

Students are not expected to understand the technical details of how computing technologies work. Students are not required to compare multiple technologies to determine which best meets a particular need.

Practice(s): ESR1, HCD12	**Disposition(s):** Critical Thinking, Reflectivenes

E4-SOC-HU-18: Distinguish between human learning and machine learning processes.

Boundary Statement: Students should compare and contrast basic characteristics of human learning and machine learning using simple, observable examples. For example, humans learn by making sense of experiences and drawing on prior knowledge, and AI systems use patterns or correlations in data to make decisions or generate new things.

Students are not expected to understand mathematical modeling, neural networks, or algorithmic training processes in technical detail nor are they expected to understand human cognition in detail. Students are not required to use or code actual AI models or apply formal data training procedures.

Practice(s): CT6, CT7	**Disposition(s):** Critical Thinking, Reflectivenes

E5-SOC-HU-18: Evaluate when it is appropriate to use or not use computing technologies to solve a problem.

Boundary Statement: Students should evaluate and justify whether computing technologies are the best tools for solving a specific p oblem by considering the context, potential benefits, and potential drawbacks. These potential benefits and drawbacks can include technical, ethical, privac , or environmental considerations. Students should focus on analyzing the feasibility of whether the solution requires computing. Deciding that AI is helpful for quickly translating a simple sentence but is not appropriate or helpful for deciding who should receive the last slice of pizza, which requires human judgment and social skills, is appropriate.

Students are not expected to understand ethical frameworks, legal implications, or technical performance metrics (e.g., error rates, training data bias) used by professional engineers to make these decisions. Students are not required to solve the problem.

Practice(s): ESR1, CT6	**Disposition(s):** Critical Thinking, Resourcefulness

MS-SOC-HU-43: Analyze how the decisions humans make when using computing technologies have ethical and social consequences.

Boundary Statement: Students should analyze how human choices in the design and use of computing technologies influen e fairness, equity, privacy, and social well-being. They should investigate how decisions such as what data to collect, which algorithms to use, or how results are interpreted can lead to different outcomes for individuals and communities. Students should explain how intentional human decisions, rather than the technologies alone, shape these ethical and social impacts and propose ways to use computing more responsibly.

Students are not expected to design or program full computing systems, perform advanced algorithmic audits, or resolve complex ethical dilemmas. Students are not required to take positions on philosophical questions about artificial intelligen e, consciousness, or rights.

Practice(s): ESR1, CT6	**Disposition(s):** Critical Thinking, Reflectivenes

HS-SOC-HU-43: Evaluate how human choices in using, designing, deploying, and regulating computing technologies have risks, benefits, and long-term impacts.

Boundary Statement: Students should evaluate the ethical and societal implications of computing technologies by examining how choices made by developers, users, and regulators shape their outcomes. Students should be able to analyze a case study to identify the human choices (e.g., data selection, model tuning, deployment criteria) that could introduce bias and discuss the resulting risks (e.g., discriminatory outcomes) and benefits (e.g., inc eased efficiency

Students are not expected to train a complex machine learning model from scratch or perform a deep mathematical analysis of different AI algorithms (e.g., backpropagation in neural networks). Students should not focus on the underlying code or mathematics.

Practice(s): ESR1, IC3	**Disposition(s):** Critical Thinking, Reflectivenes

HS-SOC-HU-44: Debate perspectives on differences between human and artificial intelligence and their implications for consciousness, ethics, and human responsibility.

Boundary Statement: Students should debate philosophical and ethical differences between human intelligence and artificial intelligen e, including questions about consciousness, agency, and responsibility. Discussions may include perspectives on whether AI systems could demonstrate human-like intelligence (e.g., artificial general intelligen e) and the ethical or societal implications that might follow. Students should analyze and articulate arguments about the roles and responsibilities of humans in designing, deploying, and governing AI systems, including issues such as accountability, bias, and decision-making authority.

Students are not expected to analyze the neurological or biological mechanisms of human consciousness or the mathematical details of complex AI models (e.g., specific neural network architectures).

Practice(s): ESR1, CT6	**Disposition(s):** Critical Thinking, Reflectivenes

Career Exploration

EK-SOC-CE-16: Identify how people use digital devices at home, at school, and at work.

Boundary Statement: Students should recognize, label, and state ways people use digital devices for daily work, learning, routines, and play. Examples such as using a digital thermometer to check temperature before going outside or parents checking work emails on a phone are appropriate. Students should participate in classroom discussions and activities where digital devices are used for communicating, learning, or helping with a task. Examples such as using manipulatives, role-play, or simple digital tools are appropriate.

Students are not expected to name all device types or explain technical details. Students are not required to operate digital devices independently or use advanced features of digital devices.

Practice(s): ESR1, IC3	**Disposition(s):** Sense of Belonging, Reflectivenes

E1-SOC-CE-16: Describe how computing is used by people at home, at school, and in the community.

Boundary Statement: Students should identify and describe simple, familiar examples of how people use computing devices every day at home, school, and in the community. Examples such as a teacher using a tablet to read a story, a parent using a phone to call family, or a sibling using a computer to do school work are appropriate.

Students are not expected to describe the internal parts of computers or the complex inner workings of the computing systems technology. Students are not required to use the computing devices themselves or to know specific technical terms

Practice(s): ESR1, IC3	**Disposition(s):** Critical Thinking, Reflectivenes

E2-SOC-CE-16: Investigate how personal interests connect to computing in different industries and careers.

Boundary Statement: Students should explore and describe, in simple terms, how their passions and hobbies connect to computing. They should also investigate how computing is used in different jobs and industries that relate to those interests. For example, a student who loves drawing could learn how artists create digital illustrations or a student who enjoys sports could explore how coaches use computers to track scores and performance.

Students are not expected to understand technical details about how computing systems work, specific job qualifi tions, or the exact tools used in various careers. Students are not required to conduct in-depth career research or simulate real-world industry work.

Practice(s): ESR1, IC3	**Disposition(s):** Sense of Belonging, Curiosity

E3-SOC-CE-19: Explain how people in different industries use computing technologies and skills to accomplish their work.

Boundary Statement: Students should describe how people in different fields (e.g., healthca e, transportation, entertainment) use computers and basic computing skills (e.g., finding i formation, storing data, communicating, creating media) to do their jobs. Examples such as a bus driver following GPS routes, an animator creating cartoons using digital tools, or a shop owner tracking inventory on a laptop are appropriate.

Students are not expected to understand how the identified technologies work or to demonstr te the referenced computing skills. Students are not required to research industries or careers in depth or understand training requirements.

Practice(s): ESR1, HCD10	**Disposition(s):** Curiosity, Reflectivenes

E4-SOC-CE-19: Investigate how the workforce adopts new computing technologies and continues to update their computing skills.

Boundary Statement: Students should examine and analyze how people in a variety of careers have learned and updated computing skills to adapt to changes in their work. Examples such as doctors learning to use digital records, teachers adopting video conferencing tools to teach remotely, farmers using GPS technology to plant and cultivate crops more precisely, or programmers applying AI tools to shorten development and debugging time are appropriate.

Students are not expected to conduct formal research or interviews with the workforce. Students are not required to demonstrate the computing skills or use the tools used by the workforce.

Practice(s): IC3, HCD10	**Disposition(s):** Curiosity, Reflectivenes

E5-SOC-CE-19: Examine how professionals collaborate while using computing technologies to solve problems.

Boundary Statement: Students should focus on describing ways different professionals (e.g., scientists, doctors, artists, engineers) work together on a task using common digital tools for communication and sharing information. Real-life scenarios, like doctors sharing digital X-rays to diagnose a patient or architects using shared design software to plan a building, are appropriate.

Students are not expected to understand the technical details of how the network or collaborative software functions. Students are not required to engage in complex collaborative projects or to use professional tools.

Practice(s): ESR1, HCD10	**Disposition(s):** Reflectiveness, Curiosit

MS-SOC-CE-44: Analyze how workers in different careers use computational thinking to solve real-world problems.

Boundary Statement: Students should investigate and describe examples of computational thinking across a range of professions. Students should be able to connect computational thinking concepts such as abstraction, decomposition, and pattern recognition to specific job functions, identifying parallels between how they think when coding and how professionals think when solving domain-specific p oblems.

Students are not expected to explore specialized programming frameworks or conduct domain-specific simul tions that require advanced mathematics, physics, or proprietary software.

Practice(s): IC4, CT6	**Disposition(s):** Sense of Belonging, Creativity

MS-SOC-CE-45: Evaluate how automation in technology can create or replace jobs and change how people work.

Boundary Statement: Students should explore how automation technologies (e.g., robotics, artificial intelligence, machine learning) have changed or are changing the nature of work across industries and analyze the social and economic trade-offs f automation. Examples where automation improves efficien , accuracy, and safety as well as cases where it disrupts or replaces jobs are appropriate. Students should also consider how automation affects job tasks and skill requirements, raises ethical and human-centered questions, and prompts workers and industries to adapt through upskilling, reskilling, or new roles.

Students are not expected to design or program automated systems, conduct in-depth economic analyses, or engage in labor-market forecasting.

Practice(s): ESR1, CT6	**Disposition(s):** Critical Thinking, Curiosity

HS-SOC-CE-45: Analyze how diverse teams of people use computational thinking and computing technologies to solve problems and express ideas.

Boundary Statement: Students should understand the real-world application of computer science, including how CS is used at work, the kinds of challenges that practitioners face, and how practitioners contribute to a more inclusive field. A oncrete example of this is having students analyze a podcast or video series featuring computer science professionals from diverse backgrounds and then identifying how they use their skills and have navigated workplace barriers.

Students are not expected to conduct formal, academic research on these topics or to interview professionals themselves, but rather to analyze existing, publicly available narratives. Students are not required to solve complex social problems, but rather to understand and articulate the issues and solutions presented in the narratives.

Practice(s): CT6, HCD10	**Disposition(s):** Critical Thinking, Reflectivenes

HS-SOC-CE-46: Connect computing knowledge and skills acquired to students' personal goals and career aspirations.

Boundary Statement: Students should evaluate how their personal interests and career aspirations connect to computing knowledge and skills. This involves critically reflecting on their own st engths and passions and researching how these can be applied in various computing-related fields

Students are not expected to make a final ca eer choice or to pursue an internship or work-based learning experience. Students are not expected to make a final ommitment to a specific ca eer path.

Practice(s): CT6, HCD10	**Disposition(s):** Reflectiveness, Sense of Belonging

High School Specialty Standards

Specialty Standards extend foundational PK–12 computer science learning by defining **advanced, domain-specific learning** in key areas of computing. These standards are organized into two tiers (Specialty I and Specialty II) across six high school specialty areas identified th ough the *Reimagining CS Pathways* project (CSTA et al., 2024):

- Artificial Intelligen e (AI)
- Cybersecurity
- Data Science
- Game Development
- Physical Computing
- Software Development

Specialty I standards introduce the knowledge and skills essential within a chosen area, serving as a student's first ocused learning experience in that domain. **Specialty II** standards describe more advanced study, preparing students for postsecondary coursework, industry certific tions, or continued specializations.

Additionally, **X+CS Standards** support interdisciplinary learning by integrating foundational high school CS content into other subject areas, such as journalism, biology, and the arts.

How do Specialty Standards Differ from Foundational Standards?

The Foundational Standards prepare *every* student for a world powered by computing, whereas the Specialty Standards extend this foundation by supporting deeper learning in specific a eas of computer science for students who choose to continue their studies, often through **courses** and **pathways**. The core differences between Foundational and Specialty Standards lie in the target audience, goal, and scope:

	Foundational Standards	Specialty Standards
Target Audience	**All** PK–12 students.	High school students who pursue continued CS learning **beyond** the foundational PK–12 standards.
Primary Goal	Develop CS knowledge, skills, and dispositions for **informed participation in society**, critical content consumption, responsible creation, and general problem-solving.	Develop **deeper, domain-specific knowledge, skills, and dispositions** that align with student interests and related pathways to college and/or career.
Scope	**Broad content** across five oncepts: Algorithms & Design, Programming, Data & Analysis, Systems & Security, and Computing & Society.	**Focused, advanced** content within a specific omputing domain (e.g., software development, artificial intelligence, data science).

How to Implement Specialty Standards

Specialty Standards can be implemented in multiple ways depending on local context, resources, and student interest. Like the Foundational Standards, they can be implemented through standalone experiences, integrated with other disciplines, or a combination.

1. **Develop Course Pathways:** Package the standards into discrete, sequential course pathways. These could focus on one specialty area or combine multiple areas.

Strategy	Course 1	Course 2
Exploratory Specialty Pathway	**Course Title**: Programming the Future **Standards**: Subsets of Specialty I standards across Software Development, Artificial Intelligence, Cybersecurity, and Data Science **Sample rationale for offering:** Many students enjoyed their foundational CS experience and want to continue their learning. They want to maintain a broad view of computer science before they dive more deeply into a singular subdomain.	**Course Title**: Information and Network Security **Standards**: Cybersecurity Specialty II (and any Cybersecurity Specialty I standards that were not covered in Programming the Future) **Sample rationale for offering:** After offering Programming the Future, student interest seemed to favor cybersecurity, so an additional course opportunity was created in this area.
Focused Specialty Pathway	**Course Title:** AI Fundamentals **Standards:** Artificial Intelligen e Specialty I **Sample rationale for offering**: The local community, including parents and employers, prioritize learning opportunities to build specialized knowledge in AI.	**Course Title:** Developing AI Applications **Standards:** Artificial Intelligen e Specialty II **Sample rationale for offering**: Students developed their skills and confirmed their inte est in AI through AI Fundamentals and wanted to continue to learn even more about how AI works and how to develop AI applications.

2. **Augment a Foundational Course to Emphasize One or More Specialty Areas:** Incorporate content from a specialty area into foundational CS learning experiences to increase early exposure to specialty areas and/or create thematic foundational experiences. This might include using AI Specialty Standards to develop an AI unit that is incorporated into a foundational CS course or using Cybersecurity Specialty Standards to dive deeper into Systems & Security concepts within a foundational experience.

3. **Guide Integrated Courses/Experiences:** Use the Specialty Standards to structure interdisciplinary courses/experiences that blend CS with other subjects (e.g., *Computational Biology* or *Computational Journalism*), ensuring the CS depth goes beyond the foundational level. This approach requires co-requisite knowledge in the non-CS discipline (X). X+CS specialty standards could be used across any discipline, however other specialty area standards can inform how CS concepts might be integrated into other disciplines as well (e.g., leverage Data Science Specialty Standards to integrate CS into math coursework). The table below shows a course progression that infuses CS and journalism:

Foundational Courses	X+CS Course
Computer Science Foundations Computer Science Foundations supports all high school students, regardless of postsecondary goals, in developing the knowledge, skills, and dispositions necessary to navigate and understand the technology-driven world in which they live. Course content is organized into five concepts: Algorithms & Design, Programming, Data & Analysis, Systems & Security, and Computing & Society. Introduction to Journalism Introduction to Journalism includes the fundamentals of gathering, writing, and reporting news stories across various media platforms. Students learn essential skills including conducting interviews, fact-checking, understanding media ethics and law, and recognizing different story formats (e.g., news articles, features, opinion pieces). The course typically emphasizes critical thinking, effective communication, and the role of journalism in a democratic society.	Computational Journalism Designed for students who have completed a foundational computing course as well as a foundational journalism course, this class exposes students to computational techniques and issues related to journalism, including: • Ethical issues, including data privacy and security • Language processing and text analysis • Reporting on technology and the technology industry • Data journalism, including data visualization • AI and its impact on the field f journalism

The Relationship Between Specialty Standards and CTE

Specialty Standards and Career and Technical Education (CTE) frameworks serve complementary purposes. Specialty Standards define advan ed CS learning, and CTE emphasizes workforce preparation and industry credentials. Specialty Standards can:

- Align with existing CTE pathways,
- Supplement CTE courses and pathways, and
- Inform the refinement f CTE standards and curricula to increase the depth and breadth of CS content.

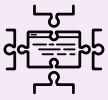
Specialty Standards are not exclusively intended to complement CTE experiences. They may also inform the development of academic courses and pathways, integrated studies, and informal or nontraditional learning experiences (e.g., out-of-school time).

Naming Conventions for Specialty Standards

Each of the identifiers or Specialty Standards follows a naming convention similar to the one for Foundational Standards:

Level		Specialty Area		Subconcept		Number
XX	-	YYY	-	ZZ	-	##

The first two characters indic te the level of the standard: Specialty I (S1) or Specialty II (S2). The next two sets of characters indicate the specialty area and subconcept for each standard. The following table shows the abbreviations for each specialty area and subconcept. The last two digits of each standard reflect the standa d number. Specialty Standards begin with 01 in each specialty area and level. The standards are numbered continuously across all subconcepts within each level of every specialty area.

	Abb	Specialty Area	Abb	Subconcepts
	**AIN**	Artificial Intelligence	DD DS HR PP	Design & Development Data Science for AI Human Responsibility Professional Practice
	**CYB**	Cybersecurity	ND NO TS CP PP	Network Theory & Design Network Operations Threats & Security Measures Cybersecurity Policies Professional Practice
	**DSC**	Data Science	CC AM MI VZ PP	Creation & Curation Analysis & Modeling Techniques Model Interpretation & Reasoning Visualization Professional Practice
	**GMD**	Game Development	DD PX AR PP	Design & Development Player Experience Architecture Professional Practice
	**PHY**	Physical Computing	HC IO DD CI PP	Hardware & Circuit Design Inputs & Outputs Design & Development Connectivity & Internet of Things Professional Practice
	**SWD**	Software Development	DD UX TR PP	Design & Development User Experience Testing & Refinin Professional Practice
	**XCS**	X+CS	XC	X+CS

Artificial Intelligence

Specialty I

S1-AIN-DD-01: Analyze AI systems to differentiate the types of problems they address.

Boundary Statement: Students should explore and compare different types of AI systems to understand how they work and what problems they solve. They should analyze generative AI (e.g., chatbots or image creators) and discriminative AI (e.g., image classifiers) to examine how each supports real-world applications (e.g., language translation, image recognition, sentiment analysis, recommendations, game playing).

Students are not expected to evaluate the performance or quality of AI systems. Students are not expected to train, build, or modify AI models.

Practice(s): CT6, CT7	**Disposition(s):** Creativity, Critical Thinking

S1-AIN-DD-02: Modify AI system training data to improve fairness and accuracy in outputs.

Boundary Statement: Students should mitigate bias and improve accuracy by modifying the data used to train an AI model. They should investigate changes in accuracy by modifying training data, making simple changes like improving data representation or balance, and keeping the changes that lead to better results. For example, students could analyze a facial-recognition model's poor performance on certain groups of users, identify the lack of diversity in the training dataset as a source of bias, and add additional examples representing the underrepresented groups to improve system performance.

Students are not expected to understand the mathematical formulas behind fairness metrics or analyze why different fairness measures cannot be satisfied t the same time.

Practice(s): ESR1, CT9	**Disposition(s):** Critical Thinking, Reflectivenes

S1-AIN-DD-03: Create an application using a prebuilt supervised learning model to make a classification or prediction.

Boundary Statement: Students should create a functional application (e.g., to classify, to predict) by selecting and integrating an appropriate, pre-existing supervised learning model. For example, students could use a pretrained image classific tion model (e.g., from a library) and integrate it into a simple web or mobile application that allows users to upload a photo and receive a prediction of the object in the image.

Students are not expected to train models from scratch using raw data, develop production-ready applications, or work with advanced architectures requiring specialized hardware (e.g., large language models, deep convolutional networks with dozens of layers).

Practice(s): CT8, CT9	**Disposition(s):** Creativity, Critical Thinking

S1-AIN-DD-04: Compare data representations and how representation choice constrains applicable algorithms.

Boundary Statement: Students should analyze the trade-offs between di erent data representations (e.g., tabular vs. relational data, graph vs. network data, image vs. pixel data, text vs. sequence data) and articulate how the chosen representation dictates the suitable class of algorithms for analysis.

Students are not expected to develop custom data representations, convert data into multiple representations, or mathematically prove why certain algorithms cannot work with specific d ta representations. Students are not required to work with specialized data formats beyond common types (e.g., tabular, graph, image, text).

Practice(s): CT6, CT7	**Disposition(s):** Critical Thinking, Resourcefulness

S1-AIN-DD-05: Evaluate whether an AI or non-AI computational solution is appropriate for a real-world problem.

Boundary Statement: Students should evaluate a real-world problem and systematically critique the trade-offs, feasibility, and appropriateness of AI-based solutions versus non-AI computational solutions. For example, students could evaluate whether to use a pre-trained image classific tion model (AI) to identify defective parts using assembly line photos or use a traditional image-processing algorithm that checks for specific pi el color ranges or geometric shapes (non-AI). Students should assess factors like data availability and interpretability, computational resources, and accuracy needed.

Students are not expected to implement AI or non-AI solutions for real-world problems. Students are not expected to produce detailed cost estimates, but recognizing that cost is a factor in choosing between approaches is appropriate.

Practice(s): CT6, CT7	**Disposition(s):** Critical Thinking, Reflectivenes

S1-AIN-DS-06: Examine how data flows through a neural network structure.

Boundary Statement: Students should examine how input data moves through a neural network's layers, how weights and activation functions modify the data at each step, and how the final output is generated. They should conceptually trace how a numerical input value flows th ough a single neuron, is modified by the weight and bias, and is then trans ormed by the activation function before moving to the next layer. Students should explain this process clearly, using appropriate vocabulary and visual representations (e.g., diagrams, annotated examples).

Students are not expected to implement neural networks from scratch, understand the mathematical formulas for backpropagation, or work with advanced architectures (recurrent neural networks, transformers, convolutional neural networks with many layers).

Practice(s): IC3, CT7	**Disposition(s):** Critical Thinking, Resourcefulness

S1-AIN-DS-07: Apply data acquisition, cleaning, and transformation techniques to prepare data for AI analysis.

Boundary Statement: Students should perform hands-on application of data preparation steps, including acquiring data from sources (e.g., comma-separated values [CSV] files, public APIs, web scraping of small-scale data), identifying and handling missing values (imputation or removal), and performing basic transformations (normalization) necessary for training an AI model. For example, students could use a programming library to load a raw dataset, clean inconsistent entries, convert non-numeric features into a machine-readable format, and split the data into training and testing sets before model application.

Students are not expected to work with datasets requiring distributed computing infrastructure or implement advanced data preprocessing algorithms from scratch.

Practice(s): CT6, HCD11	**Disposition(s):** Persistence, Critical Thinking

S1-AIN-HR-08: Plan safeguards for AI systems that protect human well-being and privacy while ensuring meaningful human involvement in decision-making.

Boundary Statement: Students should plan safeguards in AI systems focusing on practical measures to protect privacy, ensure human well-being, and maintain meaningful human involvement in critical decision-making processes. For example, students could analyze a scenario where an AI is used for medical diagnosis and formulate a human-in-the-loop workflow whe e the AI provides a recommendation, but a qualified human doctor is equired to make the final, auditable diagnosis, thereby ensuring meaningful human involvement and protecting patient well-being.

Students are not expected to develop formal methods for verifying AI safety.

Practice(s): ESR1, HCD12	**Disposition(s):** Critical Thinking, Reflectivenes

S1-AIN-HR-09: Analyze the potential biases and limitations of AI systems.

Boundary Statement: Students should employ both qualitative and quantitative methods to identify and analyze systemic issues in AI systems, investigating how social, cultural, and historical biases are encoded in the training data, model architecture, and application design. Students should analyze AI systems they created themselves as well as AI systems created by others. For example, students could perform a comparison of a facial recognition model created by others versus a text classifier c eated by the student, using metrics to show how performance varies across different demographic groups and articulating how those performance gaps may be influen ed by specific training d ta limitations.

Students are not expected to develop novel bias detection algorithms, perform complex causal inference to determine the root cause of systemic inequality, or access proprietary, nonpublic model weights and data from systems created by industry.

Practice(s): ESR1, HCD12	**Disposition(s):** Critical Thinking, Reflectivenes

S1-AIN-HR-10: Analyze the environmental impacts of widespread AI adoption.

Boundary Statement: Students should quantitatively and qualitatively analyze the environmental impacts of AI, specifically ocusing on the energy consumption required for model training and inference and the resulting carbon footprint and electronic waste. Students should consider how these environmental impacts affect different communities, recognizing that the costs of AI infrastructure (e.g., data center energy use, water consumption, e-waste disposal) are often concentrated in specific egions and populations. For example, students could compare the estimated carbon dioxide emissions generated by training a large language model versus the energy used by a local, optimized model running on an edge device.

Students are not expected to conduct complex, full life-cycle assessments for computing hardware.

Practice(s): ESR1, HCD10	**Disposition(s):** Critical Thinking, Reflectivenes

S1-AIN-PP-11: Integrate a prebuilt AI agent into an application.

Boundary Statement: Students should integrate a prebuilt AI agent into a computational application to accomplish a specific task, given i formation on available AI agents or services. Such prebuilt AI agent templates may include web search, data analysis, creative writing, or customer support bots.

Students are not expected to design or train their own agent.

Practice(s): CT6, CT8	**Disposition(s):** Resourcefulness, Persistence

S1-AIN-PP-12: Analyze how AI tools shape user experiences for people with diverse backgrounds and characteristics.

Boundary Statement: Students should analyze how AI tools produce different user experiences across diverse characteristics and backgrounds (e.g., socioeconomic status, cultural background, geographic region, language). They should examine differences in outputs and interactions and explain how training data, system design, or deployment context contribute to those differences. For example, students could analyze how a voice assistant performs for speakers with different accents or how a recommendation algorithm exposes users to different content based on demographic characteristics.

Students are not expected to design user experience studies for AI tools or implement adaptive personalization systems.

Practice(s): ESR1, HCD10	**Disposition(s):** Critical Thinking, Reflectivenes

S1-AIN-PP-13: Assess how unauthorized data collection has influenced the practice of training AI models.

Boundary Statement: Students should evaluate the tension between respecting creators' intellectual property rights and the practice of using copyrighted works to train AI models. They should also consider broader forms of unauthorized data collection, such as scraping personal images or collecting user behavior data without consent. Students could debate whether AI companies should obtain permission and provide compensation when training models on copyrighted works, considering arguments about fair use, the transformative nature of AI-generated outputs, and whether scraping publicly accessible content from the internet constitutes legitimate use or infringement.

Students are not expected to cite specific legisl tion or case law.

Practice(s): ESR1, ESR2	**Disposition(s):** Critical Thinking, Reflectivenes

S1-AIN-PP-14: Evaluate how ethical implications of AI have changed over time.

Boundary Statement: Students should conduct a critical evaluation of AI ethics, appraising how ethical considerations have changed over time. For example, students could assess the ethical concerns surrounding early conceptions of AI (Asimov's Three Laws of Robotics) and rule-based AI systems (concerns about programming errors and logic) and compare them with the ethical implications of contemporary AI systems that focus on bias, fairness, and transparency. Students should also evaluate ethical challenges that may emerge as AI capabilities advance, drawing from futuristic depictions of AI in media or their own imaginations. Students should communicate their evaluation clearly, articulating how ethical concerns have shifted and presenting a reasoned perspective on emerging challenges.

Students are not expected to conduct research on AI development trajectories or apply specific ethical frameworks. Students are not expected to rigorously define onsciousness or sentience in discussions of future AI capabilities.

Practice(s): ESR1, IC3	**Disposition(s):** Critical Thinking, Reflectivenes

Specialty II

S2-AIN-DD-01: Analyze problems that can be addressed using unsupervised learning.

Boundary Statement: Students should analyze and categorize real-world problems that can be effectively addressed using unsupervised learning approaches. They should categorize problem characteristics that make unsupervised learning appropriate (e.g., lack of labeled data, need to discover hidden patterns, exploratory data analysis) and provide examples of problem types (e.g., customer segmentation, anomaly detection, dimensionality reduction, pattern discovery in large datasets).

Students are not expected to implement unsupervised learning algorithms or prove their mathematical convergence properties.

Practice(s): CT6, CT7	**Disposition(s):** Critical Thinking, Resourcefulness

S2-AIN-DD-02: Optimize an AI algorithm's performance for a given problem.

Boundary Statement: Students should optimize the performance of a pre-existing AI algorithm or model for a given problem by modifying and tuning the parameters and structures of the algorithm or model. This involves actively altering the algorithm's control flow or hyperparameters, not simply using preset tools. For example, students might optimize an algorithm used in discriminative systems (e.g., by changing the depth of a decision tree used for an image classifier) or adjust the sampling parameters of a generative system. Students could work across different problem domains employing different optimization approaches, such as tuning weights for sentiment analysis, modifying neural network layers for image recognition, or adjusting model parameters for a recommendation system.

Students are not expected to implement algorithms from scratch, work with production-scale systems, or optimize across all possible hyperparameters using advanced techniques (e.g., Bayesian optimization, neural architecture search).

Practice(s): CT7, CT9	**Disposition(s):** Persistence, Resourcefulness

S2-AIN-DD-03: Create an application using a complex supervised learning model.

Boundary Statement: Students should design and develop an application that integrates a complex supervised learning model to solve a real-world problem. Complex supervised learning models are defined as using mo e advanced model structures (e.g., random forests, neural networks), larger or richer feature sets, and multiple layers or decision paths. The design and development process includes selecting an appropriate model type (e.g., neural networks, ensemble methods, advanced regression and classific tion techniques), preparing and preprocessing labeled data, and training the model using suitable tools or libraries. Students should incorporate the trained model into an end-to-end application (e.g., a web app, mobile app, interactive tool) that accepts user input, performs inference, and returns predictions in a usable form. For example, students could create a home price prediction application, select relevant features from housing data (e.g., square footage, location, number of bedrooms), train a supervised learning model, and evaluate its performance by testing predictions against actual prices using appropriate metrics.

Students are not expected to design or implement deep learning models.

Practice(s): IC4, CT8	**Disposition(s):** Persistence, Critical Thinking

S2-AIN-DS-04: Evaluate metadata and data pipelines to support transparency in AI model selection and deployment.

Boundary Statement: Students should critically evaluate the entire data pipeline from acquisition through cleaning for potential sources of bias, inaccuracy, and lack of transparency. Students should use this appraisal to justify or critique the selection of a specific AI model and its deployment context. For example, students could evaluate the metadata of an image dataset used to train a facial recognition model (e.g., date of collection, sensor type, demographic coverage), identify a lack of demographic diversity, and argue how this flaw ne essitates either selecting a different model less sensitive to demographic bias or changing the deployment plan (e.g., limiting its use to controlled environments).

Students are not expected to design and implement algorithmic fairness metrics.

Practice(s): ESR1, IC5	**Disposition(s):** Critical Thinking, Reflectivenes

S2-AIN-DS-05: Transform data to meet the input requirements of an AI algorithm.

Boundary Statement: Students should apply data transformation and restructuring techniques to prepare data for AI algorithms, including feature scaling, handling categorical data, aggregating data, and reshaping data. For example, students could take a raw dataset with mixed feature types and apply normalization to numerical columns and one-hot encoding to categorical columns to prepare the data for a specific algorithm (e.g., k-nea est neighbors classifier)

Students are not expected to implement dimensionality reduction techniques.

Practice(s): CT7, CT8	**Disposition(s):** Critical Thinking, Persistence

S2-AIN-HR-06: Develop a policy that promotes transparency in AI applications.

Boundary Statement: Students should develop a well-reasoned, actionable, and specific policy recommendation aimed at increasing transparency across the AI lifecycle, from raw data to deployed impact, and justify how that policy balances competing values like innovation and privacy. For example, students could draft a policy requiring that high-risk AI applications (e.g., in hiring, in criminal justice) publicly disclose the training data sources, the model type, and the results of a bias analysis before deployment.

Students are not expected to cite existing state or federal statutes or implement the technical infrastructure (e.g., secure data vaults) required to make sensitive data accessible. Students are not expected to produce policy documents that meet professional or legal standards.

Practice(s): ESR1, HCD10	**Disposition(s):** Critical Thinking, Creativity

S2-AIN-HR-07: Analyze how model optimization choices impact a model's performance, interpretability, and fairness.

Boundary Statement: Students should analyze how different optimization choices (e.g., using gradient descent, adding regularization) and accuracy measures (e.g., F1-score, mean absolute error) affect how an AI model performs and how those choices can impact fairness and interpretability. For example, students could compare how a loan approval model focused only on high recall might lead to more defaults in certain groups, whereas one focused only on high precision might unfairly reject qualified applicants

Students are not expected to write code or calculate complex metrics.

Practice(s): ESR1, CT6	**Disposition(s):** Reflectiveness, Critical Thinkin

S2-AIN-PP-08: Develop an AI agent to accomplish a defined task.

Boundary Statement: Students should design, develop, and optimize an AI agent by setting a clear goal, choosing the information it can use and defining simple steps or how it responds and takes actions. They should develop the agent using an accessible tool and optimize it by testing on a set of realistic prompts.

Students are not expected to build custom AI models for their workflo .

Practice(s): IC4, CT8	**Disposition(s):** Creativity, Resourcefulness

S2-AIN-PP-09: Develop a professional communication artifact by interpreting technical AI results for diverse audiences.

Boundary Statement: Students should develop a professional communication artifact by translating model performance metrics (e.g., F1-score, precision, recall) and ethical risks (e.g., disparate impact, bias) into language that is relevant and actionable for specific nontechnical audien es. For example, students could create a presentation on an AI-driven school admissions model where the executive summary for the school board focuses on cost-savings and scalability, and the community group briefing ocuses on fairness metrics and due process for appeals.

Students are not expected to create publishable graphics or animated media or to participate in real-world communications campaigns.

Practice(s): IC3, HCD10	**Disposition(s):** Creativity, Critical Thinking

S2-AIN-PP-10: Debate aspects of AI regulatory frameworks and legislation across countries.

Boundary Statement: Students should assess major AI regulatory frameworks from at least two regions of the world (e.g., the European Union's AI Act [EU AI Act] vs. a US state or federal approach). Students should critique the core philosophical approaches and practical impacts on key areas like data sovereignty and governance. For example, students could assess how the EU's focus on regulating "high-risk" AI systems differs from a less restrictive regulatory environment and debate the resulting impact on innovation speed versus human protections in areas like facial recognition.

Students are not expected to read the text of legislation.

Practice(s): ESR1, IC3	**Disposition(s):** Critical Thinking, Reflectivenes

Cybersecurity

Specialty I

S1-CYB-ND-01: Analyze the role of network services and protocols in secure communication and their potential vulnerabilities.

Boundary Statement: Students should analyze security trade-offs in ommon network services and protocols based on context, including when plaintext communication may be acceptable and when encryption is required. Analysis should include: (a) comparison of insecure versus secure protocol pairs (e.g., HTTP/HTTPS, Telnet/SSH), (b) examination of common vulnerabilities, and (c) contextualizing of risks to users and communities if those vulnerabilities are exploited. Students may demonstrate understanding through simulated environments, packet capture analysis, case studies, or instructor-provided demonstrations.

Students are not expected to exploit live networks, configu e enterprise infrastructure, or analyze advanced routing protocols. Hands-on Man-in-the-Middle (MitM) simulations are not required and should only be conducted in safe, isolated environments where permitted.

Practice(s): IC3, CT6	**Disposition(s):** Critical Thinking, Resourcefulness

S1-CYB-ND-02: Examine the concepts of the Open Systems Interconnection (OSI) model and their role in network communication.

Boundary Statement: Students should trace the path of a data packet through the OSI model, articulating how data is encapsulated and processed at each layer during transmission. Students should use the OSI model as a tool for reasoning, troubleshooting, and security analysis, not rote memorization. Students should identify which layers support key security functions, such as encryption, authentication, or error detection. Students should demonstrate a functional understanding of how layered systems support connectivity and security.

Students are not expected to memorize all protocol-layer mappings or perform low-level debugging. Students are not expected to recall the exact sequence of all seven layers or associate every protocol with its specific laye .

Practice(s): CT6, CT7	**Disposition(s):** Critical Thinking

S1-CYB-ND-03: Distinguish between networks based on their protocols, topologies, and addressing schemes.

Boundary Statement: Students should distinguish between common network architectures (wired, wireless, hybrid) and articulate how topologies, segmentation and addressing schemes, and protocols interact in real-world systems. Students should analyze how modern networks: (a) combine multiple topologies (e.g., hybrid, mesh), (b) use multiple protocols simultaneously, and (c) employ IPv4 and IPv6 addressing.

Students are not expected to rigidly classify networks into single categories. For example, students are not expected to classify networks into legacy addressing classes (e.g., Class A, Class B, Class C), but rather to describe how addressing is structured and used within a network. Students are not expected to design complex addressing plans or configu e production networks.

Practice(s): CT6, CT7	**Disposition(s):** Critical Thinking, Reflectivenes

S1-CYB-ND-04: Examine security risks in digital systems and corresponding mitigation strategies.

Boundary Statement: Students should analyze cybersecurity risks by identifying threats, vulnerabilities, exploits, and potential impacts using a simple risk framework that considers likelihood and impact. Students may analyze case studies or simulated scenarios to: (a) identify risks (including AI-enhanced threats, such as automated phishing), (b) propose appropriate mitigation strategies, and (c) evaluate societal impacts, including effects on trust, equity, or access to services. Hands-on investigation may occur in isolated virtual environments, such as a cyber range, or through instructor-provided artifacts.

Students are not expected to conduct penetration testing or deploy enterprise security tools. Students are not expected to access physical hardware or live networks.

Practice(s): IC5, CT6	**Disposition(s):** Critical Thinking, Reflectivenes

S1-CYB-NO-05: Apply diagnostic tools and techniques to resolve network connectivity issues.

Boundary Statement: Students should use basic network diagnostic tools to document connectivity and performance issues and to investigate whether those issues may indicate misconfigur tion, failure, or potential security concerns. At a minimum, students should: (a) use command-line or equivalent diagnostic tools (e.g., ping, traceroute, netstat, simulated equivalents), (b) connect observed behavior to relevant OSI layers to isolate where a problem may be occurring, and (c) document whether issues are due to routine network issues (e.g., latency, packet loss, routing failures) or whether they may indicate security-related events, such as denial-of-service activity or on-path interference. Students should focus on systematic troubleshooting, evidence-based reasoning, and understanding the security implications of network behavior.

Students are not expected to have access to system-level permissions or live network monitoring tools. Students are not expected to write diagnostic scripts, perform penetration testing, deploy monitoring infrastructure, or respond to live attacks. Students are not expected to perform operational remediation.

Practice(s): IC5, CT6	**Disposition(s):** Critical Thinking, Persistence

S1-CYB-NO-06: Analyze system processes and network activity using command-line tools to identify potential security concerns.

Boundary Statement: Students should analyze system or network activity to identify potential security concerns, using a defensive and ethical perspective. At a minimum, students should: (a) explain why systems generate logs (e.g., accountability, troubleshooting, security), (b) identify common indicators of concern in logs or monitoring outputs (e.g., repeated failed logins, unusual process activity, unexpected network connections), and (c) propose appropriate mitigation or response strategies based on observed behavior. Students may examine instructor-provided logs, command outputs, packet summaries, screenshots, or scripted scenarios rather than executing tools directly. When command-line tools (e.g., process listing, system status, network discovery utilities) are used, they should be limited to safe, read-only, defensive contexts within virtualized or sandboxed environments where permitted.

Students are not expected to perform vulnerability scanning, penetration testing, password attacks, exploit development, or unauthorized network scanning. Students are not expected to execute powerful tools (e.g., network scanners) on live or unauthorized networks.

Practice(s): CT6, CT7	**Disposition(s):** Resourcefulness, Persistence

S1-CYB-TS-07: Analyze how common cyber threats related to network activity exploit system vulnerabilities.

Boundary Statement: Students should analyze how common cyber threats exploit vulnerabilities in digital systems, with particular attention to network-related examples, and distinguish clearly between threats, vulnerabilities, exploits, and attacks. Students should analyze examples such as spoofing, inte ception, denial-of-service, or misuse of insecure protocols to articulate: (a) what the threat is, (b) what vulnerability is being targeted, and (c) what impact the attack could have on users or communities. Students should focus on developing conceptual understanding, classific tion, and explanation.

Students are not expected to perform attacks, exploit systems, or simulate adversarial behavior. Students are not expected to use offensive security tools or reproduce attack techniques in any environment.

Practice(s): ESR1, CT6	**Disposition(s):** Critical Thinking, Reflectivenes

S1-CYB-TS-08: Evaluate a security threat using the CIA triad, states of data, and types of control.

Boundary Statement: Students should evaluate common cybersecurity threats based on their impact on the confidentialit , integrity, and availability properties of an information system. Students should evaluate specific, ommon threat examples (e.g., ransomware, malware, denial-of-service attacks, on-path attacks) using the CIA model and assess: (a) which CIA properties are affected, (b) which data states are impacted (data at rest, in transit, or in use), and (c) which types of security controls (preventive, detective, or corrective) are most relevant. Students should focus on structured reasoning, classific tion, and explanation.

Students are not expected to design security systems, implement controls, or analyze advanced threat models. Students are not expected to select or deploy specific security p oducts or technologies.

Practice(s): CT6, CT7	**Disposition(s):** Critical Thinking, Reflectivenes

S1-CYB-TS-09: Analyze how encryption methods in network communication protect privacy and security.

Boundary Statement: Students should analyze how symmetric and asymmetric encryption methods support confidentialit , integrity, authentication, and nonrepudiation in digital systems and articulate when encryption is necessary based on context. Students should evaluate established, trusted cryptographic algorithms and libraries. Students should analyze trade-offs associ ted with encryption, including: (a) performance and computational overhead, (b) key management complexity, and (c) situations where plaintext communication may be acceptable or preferable. Students should understand at a conceptual level that encryption strength varies and that some methods require significantly mo e computational effort to break than others.

Students are not expected to implement cryptographic algorithms, break encryption, or evaluate algorithmic strength mathematically. Students are not expected to analyze the mathematical foundations of cryptographic hardness or assess how advances in computing (e.g., quantum computing) may impact specific algorithms

Practice(s): ESR1, CT6	**Disposition(s):** Critical Thinking, Reflectivenes

S1-CYB-TS-10: Evaluate security measures as part of a layered defense strategy to protect sensitive information.

Boundary Statement: Students should evaluate common security controls by categorizing them as preventive, detective, or corrective and assessing their use as part of a layered defense strategy. Students should consider security controls across multiple domains (e.g., network security such as fi ewalls and filtering, applic tion security such as input validation and authentication, device and system security such as updates and access controls, and physical security such as locks and secured locations). Students should understand that no single control is sufficient on its own and t t effective cybersecurity relies on multiple, overlapping layers of defense (defense-in-depth). Students should also evaluate trade-offs and onstraints associated with security controls, such as ease of use versus strength of protection, cost and complexity, and impact on productivity or accessibility.

Students are not expected to configu e security tools, deploy AI systems, or implement enterprise-grade defenses. Students are not expected to select or configu e specific ommercial security products.

Practice(s): ESR1, CT6	**Disposition(s):** Critical Thinking, Reflectivenes

S1-CYB-CP-11: Explain the importance of cybersecurity policies in protecting organizational assets and mitigating risks.

Boundary Statement: Students should explain the purpose of cybersecurity policies and how they translate legal, ethical, and organizational requirements into clear, actionable rules that guide user behavior and protect systems, data, and people. Students should understand the history of cybersecurity policies at a high level, including why formal security policies emerged as computing systems became networked and widely used, how early failures, breaches, and misuse highlighted the need for governance alongside technical controls, and why policy is an essential but often overlooked component of cybersecurity practice. Students should analyze common categories of organizational cybersecurity policies (e.g., acceptable use policies, data classific tion and handling policies, access control and password policies, incident reporting policies).

Students are not expected to draft comprehensive policies or interpret statutes. Students are not expected to evaluate whether specific o ganizations are in compliance with laws or regulations.

Practice(s): IC3, CT6	**Disposition(s):** Critical Thinking, Reflectivenes

S1-CYB-CP-12: Explain key components of effective security policies.

Boundary Statement: Students should explain the key components of organizational cybersecurity policies (e.g., purpose and scope, roles and responsibilities, acceptable and unacceptable behaviors, enforcement mechanisms, consequences for violations). Students should describe how these components work together to shape user behavior, organizational security, and accountability. Students should demonstrate understanding of policy structure, its social impact, and ethical trade-offs

Students are not expected to draft full organizational policies or assess legal compliance. Students are not expected to evaluate whether a specific policy meets egulatory requirements or industry standards. Students are not expected to demonstrate legal or administrative expertise.

Practice(s): ESR1, CT6	**Disposition(s):** Critical Thinking, Reflectivenes

S1-CYB-PP-13: Analyze the potential consequences of a cybersecurity decision on individuals, organizations, and society.

Boundary Statement: Students should analyze how cybersecurity decisions involve trade-offs between technical effectiveness, human judgment, and societal impact. Students should examine one or more sociotechnical decision-making scenarios in which technical solutions interact with human behavior, organizational values, and ethical considerations (e.g., automated security responses vs. human-in-the-loop decision-making, predictive threat modeling systems that prioritize certain alerts or users, monitoring systems that improve security but raise privacy or surveillance concerns). Students should demonstrate reasoned analysis, ethical evaluation, and understanding of consequences.

Students are not expected to build AI systems, deploy automated defenses, or perform real-time security operations.

Practice(s): ESR2, IC5	**Disposition(s):** Critical Thinking, Reflectivenes

S1-CYB-PP-14: Analyze social engineering techniques and how they exploit human cognitive biases and organizational weaknesses.

Boundary Statement: Students should analyze and deconstruct common social engineering techniques used to manipulate individuals into revealing information or performing unsafe actions. Students should examine concrete examples of social engineering (e.g., phishing messages, pretexting scenarios, impersonation attempts), identify persuasion techniques used (e.g., urgency, authority, trust, fear), and explain how these techniques exploit human behavior rather than technical flaws. Students should ocus on analysis, recognition, and prevention.

Students are not expected to conduct real phishing campaigns, create deceptive content, or exploit others. Students are not expected to design or deploy social engineering techniques.

Practice(s): IC5, HCD12	**Disposition(s):** Critical Thinking, Reflectivenes

S1-CYB-PP-15: Translate cybersecurity concepts, risks, and solutions clearly to both technical and nontechnical audiences.

Boundary Statement: Students should translate technical cybersecurity information into clear, accurate, and actionable communication for nontechnical audiences. Students should identify the intended audience (e.g., executives, users, community members), distill complex technical findings (e.g., vulnerabilities, risks, incidents) into plain language explanations, and highlight implications, recommendations, and next steps appropriate to the audience's role and level of technical knowledge.

Students are not expected to produce communications for real external stakeholders or organizations. Students should not rely exclusively on AI-generated text.

Practice(s): IC3, HCD10	**Disposition(s):** Critical Thinking, Resourcefulness

Specialty II

S2-CYB-ND-01: Integrate security features in networking hardware and software.

Boundary Statement: Students should integrate and configu e security features found in networking hardware (e.g., routers, fi ewalls, access points) and software (e.g., host-based fi ewalls, intrusion detection systems, server configur tions) to establish defense-in-depth. Configur tion tasks may include setting up a VPN tunnel on a router, configuring a cess control lists on a simulated switch or router, and integrating a host-based fi ewall on a server. For example, students could use virtual network labs or simulation tools to configu e fi ewall policies that allow access to a server only from a specified subnet and equire encrypted communication or authentication protocols.

Students are not expected to physically manipulate hardware, write custom firmwa e for network devices, or deploy security solutions on networks with real-time production traffi

Practice(s): CT9	**Disposition(s):** Critical Thinking, Persistence

S2-CYB-ND-02: Design a secure network, including servers, switches, routers, endpoints, and firewalls.

Boundary Statement: Students should plan security features within a network design to detect, analyze, and respond to suspicious or malicious activity, using a defense-in-depth approach. Students may demonstrate this through scaffolded design exercises, such as configuring tr ffic monitoring o logging mechanisms and articulating the indicators that might signal abnormal behavior, choosing security controls at multiple layers (e.g., network, endpoint, application), and conceptually designing automated responses (e.g., alerting, blocking traffic, iso ting compromised components). Students should configu e security features appropriately, reason about trade-offs, and explain how design choices improve security rather than pursuing exhaustive implementation.

Students are not expected to build fully automated security operations centers, tune AI models, or manage real-time threat response.

Practice(s): HCD11	**Disposition(s):** Critical Thinking, Persistence

S2-CYB-ND-03: Evaluate the security implications of different network topologies.

Boundary Statement: Students should assess how network topologies affect security, reliability, and resilience, including risks such as single points of failure, redundancy, traffic expos e, and scalability trade-offs. Students should identify potential vulnerabilities in di erent topologies and justify mitigation strategies, such as segmentation, redundancy, monitoring, and access control, distinguishing between what is inherent to the topology and what is addressed through design. Demonstrations may include diagrams, simulations, case studies, or design documentation.

Students are not expected to configu e enterprise networks, deploy live AI systems, or perform real-time network management. Students are not expected to implement or test network designs on production infrastructure.

Practice(s): ESR1, CT6	**Disposition(s):** Critical Thinking, Reflectivenes

S2-CYB-NO-04: Analyze network traffic patterns to distinguish between normal and potentially malicious behavior.

Boundary Statement: Students should analyze network traffic or log ta to distinguish normal from potentially malicious behavior using baselines, pattern recognition, and basic threat-analysis frameworks. Students should establish a baseline of typical activity (e.g., protocols, frequency, destinations), compare new data to identify anomalies, and explain how these may indicate threats such as scanning, denial-of-service, or data exfiltr tion. Students should use high-level threat-analysis frameworks that categorize attacker tactics, techniques, and behaviors to reason about attacks, without requiring deep expertise in any single framework. Students should also articulate how AI tools support analysis (e.g., anomaly detection, behavioral modeling, alert prioritization). Students may use provided datasets, simulations, visualizations, or case studies.

Students are not expected to build or train machine learning models, conduct live threat hunting, operate security operations centers, or debate regulatory or legal accountability frameworks in depth.

Practice(s): CT6, CT9	**Disposition(s):** Critical Thinking, Persistence

S2-CYB-NO-05: Analyze cybersecurity tasks to improve efficiency and reliability.

Boundary Statement: Students should analyze cybersecurity tasks to improve their efficiency an reliability, with an emphasis on secure automation practices (e.g., input validation, error handling, misuse prevention). Students may use approaches such as applying command-line tools with safe parameters to automate routine tasks (e.g., log filtering, system checks), developing simple automation scripts, or analyzing existing scripts and workflows to identify risks such as ommand injection, unsafe input handling, or excessive permissions. When scripting is used, students should demonstrate secure practices, including validating or constraining user input, handling errors safely, and avoiding hard-coded credentials or unsafe command execution.

Students are not expected to write complex programs, build production-grade automation pipelines, perform penetration testing, or assume that scripting languages themselves are "secure." Students are not expected to develop or maintain automation systems for use in operational environments.

Practice(s): ESR1, CT8	**Disposition(s):** Critical Thinking, Reflectivenes

S2-CYB-TS-06: Design access controls to protect sensitive information.

Boundary Statement: Students should design authentication and authorization mechanisms that control access to systems, services, and resources and evaluate their effectiveness in different contexts. Students should address common access control approaches (e.g., password-based authentication, multifactor authentication, role-based or attribute-based authorization). Students should consider how access control design choices affect different users, recognizing that some approaches (e.g., multifactor authentication requiring a smartphone) may create barriers for users with different resources, abilities, or contexts.

Students are not expected to build biometric systems, deploy enterprise identity platforms, or implement production-grade AI models. Students are not expected to manage real identity infrastructure or administer access control systems for operational environments.

Practice(s): CT6, HCD10	**Disposition(s):** Critical Thinking, Persistence

S2-CYB-TS-07: Evaluate security risks to determine appropriate risk management strategies.

Boundary Statement: Students should evaluate cybersecurity risks and justify appropriate risk management strategies based on likelihood, impact, and context. Students should understand the four primary risk response strategies: risk acceptance (acknowledging and tolerating risk), risk avoidance (eliminating the activity that introduces risk), risk transfer (shifting risk through mechanisms such as insurance or outsourcing), and risk mitigation (reducing likelihood or impact through controls). Students should analyze scenarios to determine which strategy or combination of strategies is most appropriate and explain the trade-offs involved. Students should onsider how risk management strategies affect users, recognizing controls that conflict with how people n turally work (e.g., overly complex password policies) may introduce new risks rather than reducing them.

Students are not expected to build machine learning models, perform quantitative risk calculations, or deploy enterprise risk platforms. Students are not expected to use formal statistical methods or professional risk assessment tools.

Practice(s): CT6, HCD12	**Disposition(s):** Critical Thinking, Resourcefulness

S2-CYB-TS-08: Examine an incident response plan for a real-world scenario.

Boundary Statement: Students should examine cybersecurity incidents and apply the phases of the incident response lifecycle (e.g., preparation, detection and analysis, containment, eradication, recovery, post-incident review). Students may demonstrate this by analyzing an existing incident response plan, examining how an organization would respond to a given incident scenario, or modifying a simplified incident esponse plan appropriate to a defined ontext.

Students are not expected to operate security operations centers, deploy automated response systems, or execute live incident handling. Students are not expected to manage real-time incident response or coordinate across organizational teams during active security events.

Practice(s): IC3, CT6	**Disposition(s):** Critical Thinking, Persistence

S2-CYB-CP-09: Debate how various regulations impact organizational security policies, procedures, and compliance.

Boundary Statement: Students should evaluate and debate how cybersecurity regulations influen e organizational policies, technical practices, and individual rights. This could include critique of different types of security regulations and standards, such as prescriptive, technically specific frameworks (e.g., those that mandate particular controls or technologies) and principles-based or rights-focused regulations that emphasize outcomes and protections rather than specific implement tions. Students should assess how organizations interpret and implement these requirements, including compliance trade-offs elated to cost, complexity, and operational impact and differences in how regulations apply across industries and data types (e.g., financial, healthca e, public sector).

Students are not expected to interpret legal text, ensure compliance, or draft regulatory guidance. Students are not expected to analyze the full text of specific laws or egulations.

Practice(s): ESR1, IC5	**Disposition(s):** Critical Thinking, Reflectivenes

S2-CYB-PP-10: Analyze the benefits, risks, and ethical implications of AI in cybersecurity.

Boundary Statement: Students should analyze the dual-use nature of artificial intelligen e in cybersecurity, recognizing that the same AI techniques may be used to enhance security defenses or enable more effective attacks. Students should examine issues such as how AI can support defensive applications (e.g., threat detection, anomaly identific tion, automated response), how AI can also be leveraged maliciously (e.g., attack automation, evasion, deception), and the responsibility of cybersecurity professionals to anticipate misuse, apply safeguards, and make ethical decisions about deployment. Students should compare trade-offs and esponsibilities associated with AI use in cybersecurity, including power versus accountability, automation versus human oversight, and innovation versus harm.

Students are not expected to build AI systems or conduct offensive cyber operations. Students are not expected to train, fine-tune, or deploy AI models or security applications.

Practice(s): ESR1, CT6	**Disposition(s):** Critical Thinking, Reflectivenes

S2-CYB-PP-11: Analyze cybersecurity vulnerabilities and incidents to inform responsible disclosure practices.

Boundary Statement: Students should examine incidents and vulnerabilities to inform disclosure practices, including coordinated disclosure processes that balance public safety, organizational responsibility, and ethical considerations. Students should examine the purpose and structure of responsible disclosure and coordinated vulnerability disclosure, the role of bug bounty programs in the cybersecurity ecosystem, and common failure modes such as premature disclosure, failure to address reported vulnerabilities, and the legal or ethical consequences of improper handling.

Students are not expected to exploit real systems, bypass safeguards, or test vulnerabilities on live infrastructure. Students are not expected to participate in actual bug bounty programs or submit vulnerability reports to real organizations.

Practice(s): ESR2, IC5	**Disposition(s):** Critical Thinking, Reflectivenes

S2-CYB-PP-12: Create documentation of cybersecurity processes and decisions that supports team coordination and accountability.

Boundary Statement: Students should create and maintain clear cybersecurity documentation that supports accountability, team coordination, and informed decision-making during security operations. Documentation may include artifacts such as incident response playbooks, security decision logs, risk assessment summaries, and change management or post-incident reports.

Students are not expected to operate live security systems, manage enterprise documentation platforms, or respond to real cyber incidents. Students are not expected to produce documentation for use in actual operational environments.

Practice(s): IC4, IC5	**Disposition(s):** Critical Thinking, Persistence

Data Science

Specialty I

S1-DSC-CC-01: Apply exploratory data analysis techniques to non-hierarchical quantitative data.

Boundary Statement: Students should apply exploratory data analysis (EDA) techniques to non-hierarchical quantitative data. Students should use computational tools to calculate summary statistics, create visualizations, and identify key characteristics of the data. Students should approach EDA as an iterative process, using initial findings to guide further explor tion. For example, students could use a programming environment with a data analysis library to load a file, then use functions to calculate the mean, median, and standard deviation of values for different variables. Students could then generate a histogram to visualize the data distribution for each variable and a scatter plot to examine the relationship between two variables.

Students are not expected to analyze or manipulate hierarchical data formats. Students are also not expected to perform statistical analysis beyond generating summary statistics. Students are not expected to account for nesting or clustering of data when generating summary statistics.

Practice(s): CT7, HCD11	**Disposition(s):** Critical Thinking, Creativity

S1-DSC-CC-02: Interpret metadata when using data collected by others.

Boundary Statement: Students should identify, interpret, and discuss the metadata associated with a dataset they did not collect themselves. This involves examining and explaining what the metadata tells them about the data's origin, collection methods, variables, and potential limitations before beginning an analysis. For example, when given a dataset, students should analyze a readme file or data dictionary to understand where the data came from, when it was last updated, what each column or variable represents, and which units of measurement were used. Students should use this information to inform their analysis. Students should also discuss potential sources of bias in the data and other factors that will influen e data interpretation.

Students are not expected to work with datasets that lack any form of metadata. Students are also not expected to perform quality assurance on the metadata and data (e.g., to compare the metadata and data to ensure they align).

Practice(s): ESR2, CT6	**Disposition(s):** Critical Thinking, Resourcefulness

S1-DSC-CC-03: Develop a program to manipulate and transform data to prepare for analysis.

Boundary Statement: Students should develop programs to perform common data manipulation and transformation tasks (e.g., filtering, g ouping, summarizing, calculating new variables, restructuring, merging datasets) to prepare a dataset for a specific analysis. For example, students ould filter a dataset to include only rows that meet specific criteria (e.g., sales f om a particular region), group data by a certain category (e.g., sales by product type), and then summarize those groups (e.g., calculate the total sales for each product type). They should also be able to calculate new variables (e.g., product prices converted to a different currency), restructure a dataset (e.g., from a long format to a wide format), and merge two or more datasets based on a common key. Students should work with datasets large enough to require processing with computational tools. They should use existing libraries designed for these purposes.

Students are not expected to build their own tools for data manipulation. Students are not expected to work with datasets that require memory or processing power beyond what is available on their personal devices. Students are not expected to optimize code for performance with very large datasets.

Practice(s): CT6, CT8	**Disposition(s):** Critical Thinking, Persistence

S1-DSC-AM-04: Apply appropriate analytic and visualization techniques for categorical and quantitative data.

Boundary Statement: Students should select and apply appropriate computational methods for analyzing and visualizing categorical data (e.g., bar charts, mosaic plots, frequency tables) and quantitative data (e.g., histograms, scatter plots, box plots, summary statistics). Students should understand how the data type influen es the choice of analytic and visualization techniques. Students should also consider who will interpret the visualization when selecting a technique, recognizing that some chart types are more accessible to general audiences than others. For example, students could use computational tools to create a bar chart showing the distribution of car types in a dataset and a scatter plot examining the relationship between engine size and fuel efficien .

Students are not expected to perform inferential statistical tests (e.g., chi-squared tests, t-tests, ANOVA) or statistical modeling techniques that require understanding of probability distributions and statistical significan e (e.g., linear regression, logistic regression). Students are not expected to work with multidimensional data requiring techniques like principal component analysis or advanced machine learning models.

Practice(s): CT6, HCD12	**Disposition(s):** Critical Thinking, Resourcefulness

S1-DSC-AM-05: Examine missing data and its impact on data analysis.

Boundary Statement: Students should examine datasets with missing values to understand the nature and extent of the missing data (e.g., identifying which variables have missing values, how much data is missing, whether there are patterns in what is missing). Students should analyze how missing data affects different analytical approaches, recognizing that some methods require complete data, whereas others can handle missing values. For example, students might explore a housing dataset where income data is partially missing and detect that some visualization techniques (e.g., scatter plots) automatically exclude incomplete cases while summary statistics can be calculated on available data. Students should understand that decisions about handling missing data (e.g., removing incomplete cases, using mean imputation) can affect results and potentially introduce bias. Students should also recognize that when data is systematically missing for particular groups or communities, analyses may fail to capture impacts on those groups, leading to decisions that overlook or disadvantage them.

Students are not expected to formally classify missing data mechanisms (i.e., missing completely at random, missing at random, missing not at random) using statistical tests, though discussing these concepts is appropriate. Students are not expected to implement advanced imputation techniques (e.g., k-nearest neighbors imputation, multiple imputation by chained equations) or to perform formal statistical comparisons of model performance with different missing data treatments.

Practice(s): ESR1, CT6	**Disposition(s):** Critical Thinking

S1-DSC-AM-06: Analyze structured categorical and/or quantitative datasets, using computational tools and libraries.

Boundary Statement: Students should perform exploratory data analysis on structured datasets using computational tools and libraries. They should write and execute code to clean, transform, and aggregate data and use visualization libraries to create charts and graphs that reveal trends, patterns, and relationships within the data. For example, students could use a library to analyze a dataset of global climate information, computing and visualizing average temperature changes over a decade for different continents or grouping temperature data by month to calculate average monthly temperatures.

Students are not expected to work with datasets that require memory or processing power beyond what is available on their personal devices. Students are not expected to use distributed computing frameworks or build data analysis tools from scratch.

Practice(s): CT6, CT8	**Disposition(s):** Critical Thinking, Persistence

S1-DSC-MI-07: Interpret the results of a data analysis to explain patterns, anomalies, and trends, and connect them back to the original problem or research question.

Boundary Statement: Students should interpret existing data analyses (e.g., visualizations, summary statistics, published findings) and explain wh t the results mean in relation to the original research question or problem. Students should identify key patterns, trends, and anomalies in the data and explain how these findings add ess the research question. Students should discuss limitations of the analysis and describe alternative explanations for the observed patterns. For example, students might interpret a visualization showing declining bee populations over time, identify the downward trend, describe how this pattern relates to a research question about pollinator health, and discuss possible explanations (e.g., pesticide use, habitat loss) while reporting limitations such as the time period covered or other factors not measured in the dataset.

Students are not expected to perform the statistical calculations or create the visualizations themselves for this standard. Students are not expected to conduct formal hypothesis testing, calculate inferential statistics, or replicate the original analysis to validate their interpretations.

Practice(s): IC3, CT6	**Disposition(s):** Critical Thinking, Reflectivenes

S1-DSC-MI-08: Explain the appropriateness of predictive models for the specific problem being addressed.

Boundary Statement: Students should differentiate the suitability of predictive models for varying problem types at a conceptual level. Students should consider the type of data, the nature of the outcome (e.g., categorical vs. continuous), and the potential real-world impact of the model's performance. Students should recognize that selecting an inappropriate model for a high-stakes problem can lead to harmful outcomes for the people affected by the model's predictions. For example, students could be given a problem to predict whether a customer will click on an ad. They should explain why a logistic regression model is more appropriate than a linear regression model for this specific p oblem, which requires predicting a categorical outcome (yes/no) rather than a continuous numerical value.

Students are not expected to build or assess deep learning models. Students are not required to statistically evaluate different models to determine which model has the best fit

Practice(s): ESR1, CT6	**Disposition(s):** Critical Thinking, Reflectivenes

S1-DSC-MI-09: Explain how dataset size affects model stability and performance.

Boundary Statement: Students should describe how adding or removing data points (rows) affects model behavior (e.g., output variance) when the model is rerun. Students should recognize that small datasets are more sensitive to changes than large datasets. Adding or removing even a small number of rows from a small dataset can cause substantial changes in model outputs, whereas adding or removing the same number of rows from a large dataset has minimal impact. Students should understand that larger datasets generally produce more stable and reliable models. Students should observe and describe the practical effects of dataset size on model behavior. For example, students might start with a model trained on 20 house sales, observe how predictions can change dramatically when adding or removing 10 sales, then differentiate this from the case of adding or removing 10 sales from a dataset of 1,000 sales where predictions barely change.

Students are not expected to calculate formal statistical measures of model stability. Students are not expected to understand the mathematical relationship between sample size and variance.

Practice(s): CT7, CT9	**Disposition(s):** Critical Thinking, Reflectivenes

S1-DSC-VZ-10: Create a data visualization that communicates key findings to diverse audiences.

Boundary Statement: Students should create data visualizations that effectively communicate specific findings to d erent audiences, demonstrating how visual elements influen e comprehension and interpretation. They should intentionally select appropriate chart types (e.g., line graphs for trends, bar charts for categories, scatter plots for relationships) and apply formatting to enhance clarity (e.g., clear titles, labels, meaningful colors). For example, students might create a bar chart showing average height by grade level, using color to distinguish school levels for a general audience. For more data-savvy audiences, they could add error bars or use density plots to show distribution and overlap. Overall, students should design intentionally with an understanding of how visual choices affect interpretation.

Students are not expected to create complex data visualizations or interactive dashboards. Students are not expected to user-test and collect feedback on their visualizations.

Practice(s): CT8, HCD10	**Disposition(s):** Critical Thinking, Reflectivenes

S1-DSC-VZ-11: Analyze how graphical conventions in data visualizations support accurate interpretation and how breaking conventions can mislead.

Boundary Statement: Students should understand established graphical conventions in data visualization and explain how deviating from them can misrepresent data and mislead audiences. This includes conventions such as Y-axes increasing from bottom to top, meaningful color associations, and appropriate chart selection for data types. For example, students might analyze a chart with an inverted Y-axis, explain how it disrupts interpretation, and create a corrected version with the standard orientation.

Students are not expected to analyze or correct 3D or interactive data visualizations, data visualizations that use logarithmic scales, or other more advanced types of data visualizations (e.g., network graphs, radar graphs, sankey diagrams).

Practice(s): ESR1, IC3	**Disposition(s):** Critical Thinking, Reflectivenes

S1-DSC-PP-12: Apply ethical principles to data collection, analysis, and communication to promote privacy, transparency, and accountability.

Boundary Statement: Students should apply ethical principles in the context of data science projects to promote privacy, transparency, and accountability. For example, in a project involving a social media dataset, students could develop a plan for data anonymization to protect user privacy and draft a public-facing statement explaining what data was used, how it was analyzed, and what conclusions were drawn, thereby promoting transparency and accountability.

Students are not expected to read or interpret data privacy laws or use statistical techniques to quantify bias or representativeness.

Practice(s): ESR2, HCD10	**Disposition(s):** Critical Thinking

S1-DSC-PP-13: Assess how data collection and use may impact marginalized and underrepresented groups.

Boundary Statement: Students should analyze and articulate the potential impacts of data collection and use on different communities, particularly those who are marginalized or underrepresented. Students should specifically examine how d ta choices can exacerbate or mitigate existing social inequalities. For example, students could analyze a facial recognition dataset to assess its representation of different skin tones and genders and propose ways to collect a more inclusive and equitable dataset.

Students are not expected to use statistical tests to quantify demographic representation or develop bias mitigation algorithms.

Practice(s): ESR1, ESR2	**Disposition(s):** Critical Thinking, Reflectivenes

S1-DSC-PP-14: Document data analysis results in a format appropriate for an audience with diverse backgrounds and perspectives.

Boundary Statement: Students should translate data analysis results into clear narratives for different audiences, using appropriate documentation and communication formats. Students should articulate the purpose, methods, findings, and ethical onsiderations of their analysis. For example, students could create a presentation for a public audience explaining findings f om a community health data analysis and write a technical report for other data scientists detailing the study design, data collection methods, and statistical models used in the analysis.

Students are not expected to create data visualizations using professional tools without structured guidance. Students are not expected to produce publication-quality writing or deliver formal public presentations.

Practice(s): IC3, HCD10	**Disposition(s):** Critical Thinking, Resourcefulness

Specialty II

S2-DSC-CC-01: Apply exploratory data analysis techniques to a hierarchical structured data source.

Boundary Statement: Students should apply exploratory data analysis techniques to hierarchical structured data sources. This involves navigating nested data, extracting relevant information from different levels of the hierarchy, and applying summary statistics and visualizations to this extracted data. For example, students could be given a JSON file ontaining nested information about an API response, such as a list of songs, each with a nested object for the artist, including their name, genre, and a list of other albums. Students could parse this data to calculate the average length of songs for a specific gen e or to create a bar chart showing the number of songs per artist.

Students are not expected to handle deeply nested or graph-structured data.

Practice(s): CT7, HCD11	**Disposition(s):** Critical Thinking, Resourcefulness

S2-DSC-CC-02: Document the origin, structure, and preparation of a dataset to support clarity and reproducibility.

Boundary Statement: Students should create clear and detailed documentation for a dataset they have prepared for analysis, ensuring that collaborators and other analysts can understand and reproduce their work. This includes creating a comprehensive record of the data's entire lifecycle, from its source to its final p epared state. Students should document the data's origin, collection methods, and all steps taken to clean, transform, or organize it for analysis. Students should treat documentation as an ongoing part of the data preparation process, maintaining records as decisions are made rather than reconstructing them after the fact. For example, students could write a report detailing that their dataset came from a specific government website's AP , was collected on a certain date, and was then cleaned by removing missing rows and normalizing a date column into a standard format, providing the code snippets used for each step.

Students are not expected to use version control systems to track every change. Students are not expected to maintain documentation across multiple simultaneous projects or coordinate documentation standards across a team.

Practice(s): IC4, IC5	**Disposition(s):** Persistence, Reflectivenes

S2-DSC-CC-03: Develop a program to collect and integrate data from multiple sources.

Boundary Statement: Students should develop programs to collect data from diverse sources (e.g., APIs, web scraping, databases) and integrate datasets that were not originally designed to be combined. Students should identify or create common keys to enable merging (e.g., matching records based on ticker symbols, dates, or other shared attributes) and verify that the merge was performed correctly by checking for expected row counts and examining merged records. This will enable students to then transform the combined data (e.g., filtering, g ouping, summarizing, calculating new variables, restructuring) to answer more complex and original questions that cannot be answered with a single, precleaned dataset. For example, students might collect real-time stock market data from an API, integrate it with a historical dataset from a file, c eate a common key based on stock ticker and date, verify the merge, and then group by industry sector and calculate summary statistics to analyze trends and make predictions.

Students are not expected to perform complex data matching algorithms (e.g., fuzzy matching for approximate string matches, probabilistic record linkage). Students are not expected to handle unstructured data (e.g., images, audio files, raw text) th t requires parsing or machine learning techniques. Students are not expected to write their own API clients or web scrapers from scratch. They may use existing libraries for data collection.

Practice(s): CT8, CT9	**Disposition(s):** Critical Thinking, Persistence

S2-DSC-CC-04: Choose appropriate data to collect for a data science project based on available tools, skills, and project goals.

Boundary Statement: Students should analyze a data science project's objectives and make informed decisions about what kind of data to collect. This involves evaluating project goals to determine what questions need to be answered. Students should consider available tools (e.g., libraries for web scraping, APIs, sensor hardware) and their own skills to choose a feasible data collection strategy. For example, if a project's goal is to analyze local air quality, students should consider whether to use a public API, purchase and use a physical sensor, or use an existing government dataset, weighing the pros and cons of each method against their resources and abilities and the nature of the research question.

Students are not expected to build data collection tools from scratch (e.g., custom web scrapers). Students are not required to work with large datasets that require memory or processing power beyond what is available on their personal devices. Students are not required to work with data from multiple, disparate sources.

Practice(s): IC4, CT6	**Disposition(s):** Critical Thinking, Reflectivenes

S2-DSC-AM-05: Build a model to explain relationships, make predictions, and evaluate the influence of different variables.

Boundary Statement: Students should apply simple computational models (e.g., linear regression, decision trees, basic simulations) to make predictions and explain relationships within a dataset. They should evaluate how different input variables influen e the model's output. For example, students could apply a simple linear regression model to predict runners' race times based on factors like training hours and diet and use the model's output to explain which factors have the most significant influen e on performance. Students could also use a decision tree to classify an email as spam or not spam and explain the sequence of decisions the model makes to arrive at its conclusion.

Students are not expected to implement these model algorithms from scratch or to understand the underlying mathematical theory in depth. Students are not expected to work with advanced machine learning techniques (e.g., neural networks, ensemble methods, deep learning).

Practice(s): CT6, CT8	**Disposition(s):** Critical Thinking, Persistence

S2-DSC-AM-06: Evaluate strategies for handling missing data.

Boundary Statement: Students should evaluate different strategies for handling missing data in a dataset (e.g., empty cells, null values, placeholder values like "NA"). Students should select and apply various strategies for handling missing data. Common strategies include removing rows or columns with missing data, imputing missing values (e.g., using mean, median, a specified value), or retaining missing values and excluding them from specific calcul tions. They should consider how their chosen approach affects analysis results when selecting and implementing a strategy. Students should recognize that when data is systematically missing for particular groups or communities, some strategies (e.g., deletion) may remove those groups from the analysis entirely, whereas others (e.g., mean imputation) may mask meaningful differences. Students should make informed decisions about handling missing data based on observed effects on their analysis. For example, students might observe that removing all rows with any missing values dramatically reduces dataset size and then try mean imputation instead, comparing how the two approaches affect summary statistics or visualizations to inform their choice of which strategy to use.

Students are not expected to implement advanced imputation algorithms (e.g., k-nearest neighbors imputation, multiple imputation). Students are not expected to use statistical tests to assess the quality of different missing data strategies.

Practice(s): ESR1, CT6	**Disposition(s):** Critical Thinking, Reflectivenes

S2-DSC-AM-07: Analyze an unstructured, mixed-type, or high-dimensional dataset using computational tools and libraries.

Boundary Statement: Students should analyze and derive insights from unstructured data (e.g., text documents) or high-dimensional data where the number of features is very large (e.g., image data where each pixel is a feature, sensor data that is sampled many times per second). They should use appropriate computational libraries and techniques to process, clean, and analyze these complex data formats. For example, students could be given a collection of text documents and use a library to perform a simple analysis (e.g., word frequency counting) to find ey themes. For a high-dimensional dataset, students could apply a dimensionality reduction technique to visualize or analyze the data more effectively.

Students are not expected to build models for complex tasks (e.g., natural language processing, image recognition, complex signal processing).

Practice(s): CT7, CT8	**Disposition(s):** Critical Thinking, Resourcefulness

S2-DSC-MI-08: Evaluate the performance of models using established metrics.

Boundary Statement: Students should evaluate and compare the performance of different models using computational tools and established metrics. Students should assess a model's fit by using metrics such as accuracy or precision for classific tion models or mean squared error or R-squared for regression models. Students should also assess computational efficiency by measuring how long i takes to train or execute a model. Students should use existing library functions to generate evaluation metrics and use those results to make informed judgments about model quality, considering trade-offs between speed and a curacy. For example, after building a model to predict house prices, students could compare its performance to another model with more predictor variables by calculating the R-squared value for each. Students could then decide which model is a better fit or the data and discuss if the inclusion of more variables is worth the extra resources needed to collect additional data.

Students are not expected to implement evaluation metrics from scratch or to understand the mathematical derivations of metrics. Students are not expected to perform theoretical algorithmic analysis (e.g., Big O notation) to assess computational complexity.

Practice(s): CT7, CT9	**Disposition(s):** Critical Thinking, Persistence

S2-DSC-MI-09: Analyze the trade-offs between interpretability, accuracy, and generalizability as they relate to model complexity.

Boundary Statement: Students should analyze how choices in model complexity create trade-offs between interpretability, accuracy, and generalizability. A more complex model may be more accurate on training data but is often more difficult to understand and inter et. Additionally, complex models risk overfitting (e.g., learning the training d ta too well, including noise and outliers), which reduces their ability to generalize to new, unseen data. Students should articulate why a simpler model that is less accurate but highly interpretable may be more suitable for certain applications. For example, students could compare two models to predict whether a student will pass or fail a course: a simple logistic regression model that uses only study hours and attendance as features and a complex neural network that uses dozens of features. The logistic regression might achieve 80% accuracy and allow educators to see that study hours and attendance are the strongest predictors of passing, with clear numeric weights showing how much each factor contributes, while the neural network might achieve 85% accuracy but function as a "black box" where teachers cannot understand why specific predictions are made. For advising students, the interpretable model may be more valuable despite lower accuracy.

Students are not expected to build or train machine learning models from scratch. Students are not expected to understand or derive the mathematical theory behind why these trade-offs exist

Practice(s): CT7, HCD12	**Disposition(s):** Critical Thinking

S2-DSC-MI-10: Analyze how adding or removing variables affects model behavior and performance.

Boundary Statement: Students should analyze how changes in a dataset (e.g., adding, removing, modifying variables) impact model predictions or performance when the model is rerun. Students should examine how adding correlated variables (e.g., number of bedrooms and square footage) can affect how the model uses each variable and may change which variables appear most important. Students should analyze how different combinations of variables affect model accuracy or error. For example, students might start with a model predicting house prices based on square footage alone, then add variables like years since remodel or median neighborhood price to examine how this affects the model's accuracy or changes the relative importance of square footage.

Students are not expected to calculate variance infl tion factors or other formal measures of multicollinearity. Students are not expected to understand the underlying mathematical reasons for how correlated variables affect model coefficient

Practice(s): CT7, CT9	**Disposition(s):** Critical Thinking, Persistence

S2-DSC-VZ-11: Create a visualization that accurately represents data and avoids misleading design choices.

Boundary Statement: Students should create data visualizations while making deliberate design choices to avoid common pitfalls that can mislead viewers. Students should use appropriate axis ranges, maintain consistent scales, select chart types that match the data and message, and avoid visual distortions. Students should refine their visualiz tions by checking that the visual representation accurately reflects the underlying d ta. Students should also consider accessibility (e.g., using colorblind-friendly palettes, providing text alternatives) and adhere to appropriate style guides and standards (WCAG). For example, students creating bar charts to visualize SAT scores (200–800 range) and ACT scores (1–36 range) should recognize that using the same axis scale (e.g., 0–800) for both tests would compress ACT score variation and make differences appear negligible, whereas using appropriate scales for each test (200–800 for SAT, 1–36 for ACT) accurately represents the variation within each test.

Students are not expected to conduct formal user testing or collect empirical feedback on visualization effectiveness. Students are not expected to create interactive dashboards or use advanced visualization libraries. Students are not expected to master all accessibility standards but should demonstrate awareness of basic considerations.

Practice(s): ESR1, HCD10	**Disposition(s):** Critical Thinking, Reflectivenes

S2-DSC-VZ-12: Critique a data visualization for misleading elements and their role in spreading misinformation.

Boundary Statement: Students should evaluate a data visualization and accompanying narrative from public sources (e.g., news articles, social media posts, advertisements, political campaigns) to identify misleading or deceptive elements. They should assess how specific design choi es (e.g., truncated or flipped a es, cherry-picked data ranges, counterintuitive color palettes) can distort viewers' understanding. Students should recognize that design choices may affect different audiences differently. For example, color associations vary across cultures (e.g., red signifies danger or loss in some Western contexts but prosperity or good fortune in many East Asian cultures), meaning the same visualization can communicate different messages to different viewers. Students should debate how misleading visualizations contribute to spreading misinformation or disinformation. For example, students might analyze a social media post showing a dramatic-looking trend line that uses a truncated Y-axis and a selective time range, evaluating how these choices exaggerate the trend, what narrative the visualization promotes, and the potential real-world consequences of its spread.

Students are not expected to verify data authenticity or trace the original source of visualizations. Students are not expected to determine the intent behind misleading visualizations (whether deliberate deception or honest mistake).

Practice(s): ESR1, HCD10	**Disposition(s):** Critical Thinking, Reflectivenes

S2-DSC-PP-13: Evaluate ethical, legal, and social considerations when working with large-scale datasets, predictive models, and emerging technologies.

Boundary Statement: Students should evaluate the ethical, legal, and social implications of data science projects, particularly those that involve large datasets and predictive models, and apply principles of fairness, accountability, and transparency in their own work. Students should actively assess how decisions at each stage, from data collection through model deployment, can cause harm and take steps to prevent it. For example, when building a predictive model, students should identify potential sources of bias in the data, assess the societal consequences of the model's predictions, and debate methods for increasing fair and transparent decisions.

Students are not expected to read or interpret specific d ta privacy laws and regulations (e.g., CCPA, GDPR). Students are not expected to use mathematical techniques to measure fairness or to conduct statistical tests to validate model performance.

Practice(s): ESR1, ESR2	**Disposition(s):** Critical Thinking, Reflectivenes

S2-DSC-PP-14: Evaluate protective measures in data collection, usage, and governance for privacy, security, and fairness.

Boundary Statement: Students should evaluate protective measures used in data collection, usage, and governance, including privacy safeguards, security protocols, consent mechanisms, and bias prevention strategies. Students should assess the effectiveness of these measures in specific ontexts and examine their interconnectedness. For example, students could evaluate a mobile application's data practices by critiquing its privacy policy and requested permissions, assessing potential privacy risks (e.g., unnecessary location tracking) and evaluating alternative approaches that would better protect user privacy and prevent bias.

Students are not expected to implement cryptographic security protocols (e.g., encryption algorithms) or to develop and deploy consent management systems (e.g., software that tracks user consent choices, enforces data usage permissions, generates compliance audit trails).

Practice(s): ESR2, HCD12	**Disposition(s):** Critical Thinking, Resourcefulness

S2-DSC-PP-15: Translate technical results for diverse audiences in professional communication artifacts.

Boundary Statement: Students should translate findings f om data analyses into clear, actionable insights tailored for different audiences with varying levels of technical expertise. Students should strategically craft messages that resonate with each audience's interests and needs. For example, students could create a presentation for a city council, emphasizing the policy implications of a traffi data analysis, while simultaneously preparing a detailed technical report and code library so other scientists can reproduce the results.

Students are not expected to produce publication-quality writing, create publication-ready infographics with custom illustrations, or deliver formal public presentations.

Practice(s): IC3, HCD10	**Disposition(s):** Critical Thinking, Resourcefulness

Game Development

Specialty I

S1-GMD-DD-01: Analyze the fundamental components of a game, including players, rules, game mechanics, and outcomes.

Boundary Statement: Students should analyze the core components of simple games, including player roles, available actions, game rules that govern play, win/loss conditions, and how game state changes based on player actions. Students should represent these components using appropriate models (e.g., state diagrams, flowcharts, rule tables). For example, students analyzing a simple ca d game could document the initial setup, the legal moves on each turn, how card plays affect game state, and the conditions that end the game.

Students are not expected to analyze complex games with extensive rule sets, probabilistic elements, or real-time mechanics that would require sophisticated implementation techniques.

Practice(s): CT6, CT7	**Disposition(s):** Critical Thinking, Reflectivenes

S1-GMD-DD-02: Create a storyboard to plan and communicate a game's narrative and interactive experience.

Boundary Statement: Students should create storyboards that visually outline the narrative flow and interactive elements of a game. Their storyboards should illustrate key scenes, player choices, decision points, and how the game responds to those choices. Storyboards should use simple sketches and annotations to clarify on-screen actions, intended player interactions, and the emotions or reactions the designer wants to evoke. Students should focus on using storyboards as a planning and communication tool. Basic, low-fidelity s etches that clearly convey the game's structure and interactivity are sufficien

Students are not expected to create polished or artistically detailed drawings or depict every possible branching path.

Practice(s): IC3, CT6	**Disposition(s):** Creativity

S1-GMD-DD-03: Develop an interactive experience to support gameplay.

Boundary Statement: Students should develop interactive functions that combine to create a playable gameplay experience, such as responding to player input, updating game state, providing feedback, or determining outcomes. At a minimum, the gameplay experience should include: (a) player input that affects the system (e.g., movement, selection, timing), (b) programmatic responses to that input (e.g., state changes, scoring, feedback), and (c) clear conditions that influen e progression or outcomes (e.g., win/lose states, level changes, time-based events). Students should design interactions that give players meaningful feedback and a sense of agency over outcomes. Students should demonstrate how individual interactive functions work together to produce gameplay, rather than existing as isolated features.

Students are not expected to create complete commercial games, advanced user interfaces, complex narratives, or polished visual assets. Students are not expected to produce production-quality output.

Practice(s): CT8, HCD12	**Disposition(s):** Creativity, Critical Thinking

S1-GMD-DD-04: Adapt rule-based logic to improve control of game agents or systems.

Boundary Statement: Students should evaluate and adapt existing rule-based logic that governs the behavior of an in-game agent or system, such as a character, an object, the environment, or a game mechanic. Adaptations should focus on increasing conditional complexity, such as adding additional decision branches, combining conditions, or refining th esholds that affect behavior. Students should validate how changes to conditional logic impact game behavior, challenge, or player experience. For example, students might adapt the logic controlling a moving obstacle, scoring system, interactive object, or nonplayer character by adding new conditions that respond to player actions, game state, or time.

Students are not expected to design artificial intelligen e systems, implement learning or adaptive algorithms, or formally evaluate algorithmic efficien .

Practice(s): CT6, CT9	**Disposition(s):** Critical Thinking, Resourcefulness

S1-GMD-PX-05: Create a user-friendly interface for a game, considering accessibility, usability, and aesthetics.

Boundary Statement: Students should design a user interface and gameplay presentation that supports usability, accessibility, and aesthetic coherence within a game or interactive experience. Design decisions should address a small, clear set of criteria, such as: (a) readability (e.g., legible text, clear visual hierarchy), (b) consistency (e.g., icons, controls, feedback behaving predictably), (c) color contrast or visual clarity, and (d) at least one accessibility consideration (e.g., adjustable text size, alternative input options, remappable controls, simplified interaction modes). Examples f accessibility features are illustrative, not exhaustive, and students may select options appropriate to their game type, audience, and toolset.

Students are not expected to implement comprehensive accessibility compliance standards, conduct formal usability audits, or design professional-grade interfaces.

Practice(s): CT8, HCD10	**Disposition(s):** Creativity, Reflectivenes

S1-GMD-PX-06: Optimize basic game usability through user testing and identifying actionable design revisions.

Boundary Statement: Students should optimize game usability by conducting basic user tests. They should collect observable usability data by watching users interact with a game or an interactive system and identifying how design choices affect user experience. At a minimum, evidence of usability testing should include: (a) observation of at least two users interacting with the system, (b) two to three key user flows or interactions (e.g., starting the game, ompleting a task, responding to feedback), and (c) a short list of observed friction points, such as confusion, errors, delays, or unintended behavior. Students should use their observations to suggest or implement simple design improvements based on identified issues

Students are not expected to conduct formal usability studies, collect large datasets, apply statistical analysis, or use professional testing tools.

Practice(s): HCD10, HCD11	**Disposition(s):** Reflectiveness, Critical Thinkin

S1-GMD-AR-07: Examine the key architectural features of computing hardware and their implications for game development.

Boundary Statement: Students should examine how major computing system components influen e game performance and player experience, including processing, graphics, input, audio, memory, and networking at a conceptual level. Students should examine how different hardware and software components contribute to gameplay outcomes, such as frame rate, responsiveness, visual quality, and input latency. Graphics processing units (GPUs) may be introduced as one example of specialized hardware that supports parallel processing for rendering, but they are not the sole focus.

Students are not expected to analyze low-level hardware specific tions, compare specific devi es, optimize code for particular platforms, or work with high-end or professional gaming hardware. Students are not expected to demonstrate detailed technical knowledge of hardware internals.

Practice(s): IC3, CT6	**Disposition(s):** Critical Thinking, Resourcefulness

S1-GMD-PP-08: Analyze the ethical implications of copyright and intellectual property in game development.

Boundary Statement: Students should analyze ethical considerations related to intellectual property, attribution, accessibility, and user well-being in game development by applying a clear, concrete framework. At a minimum, students should: (a) select and apply an appropriate license or usage model (e.g., Creative Commons, public domain, original work), (b) properly attribute assets or code used in their project, and (c) articulate how their choices respect creators' rights and player communities. Students might also examine at a conceptual level how distribution platforms, marketplaces, or publishing environments can impact creator rights, ownership, or control.

Students are not expected to provide legal analysis or interpret copyright law.

Practice(s): ESR2, IC3	**Disposition(s):** Reflectiveness, esourcefulness

S1-GMD-PP-09: Develop a game project collaboratively within a diverse team.

Boundary Statement: Students should develop game projects collaboratively. They should effectively collaborate on a game project by communicating design ideas, sharing responsibilities, and reflecting on how design choices may affect different player communities. Students should use appropriate tools and methods (e.g., shared documents, version tracking, task boards) to organize tasks and coordinate contributions across the team. Students should engage in guided discussion or reflection on how historical or contemporary game design practices can contribute to inclusion, exclusion, or harm, including the absence of underrepresented voices or the reinforcement of stereotypes.

Students are not expected to control team composition or resolve complex social issues. Consideration of diverse perspectives may be supported through discussion, analysis of examples, or reflective writing, r ther than required team structures.

Practice(s): IC3, IC4	**Disposition(s):** Sense of Belonging, Persistence

Specialty II

S2-GMD-DD-01: Justify the use of responsible design principles in developing an engaging, meaningful game experience.

Boundary Statement: Students should, in the context of a game development project, apply and justify at least two responsible design choices drawn from the following categories: accessibility (e.g., remappable controls, color-contrast adjustments, alternative input methods), inclusivity (e.g., avoiding stereotypes, representing diverse identities, using culturally responsive themes), sustainability or player well-being (e.g., discouraging addictive mechanics, promoting healthy play patterns, minimizing unnecessary data collection), and transparency or trust (e.g., clearly communicating rules, feedback, data use). Students justify each design choice by explaining how it affects player experience, behavior, or equity and demonstrate how these choices are integrated into the gameplay or interaction design rather than added as afterthoughts. For example, students might design a game for a specific audience that includes adjustable difficulty and visual cessibility settings and assess how those features support player engagement and fairness.

Students are not expected to conduct formal user research studies, implement every category of responsible design, or build professional-grade analytics, monetization, or production pipelines.

Practice(s): ESR1, HCD12	**Disposition(s):** Resourcefulness, Reflectivenes

S2-GMD-DD-02: Develop a system that programmatically controls 2D or 3D animation states based on game logic and player interactions.

Boundary Statement: Students should develop systems to programmatically control existing animation states in a 2D or 3D game environment to reflect character actions, game events, or system states. This includes triggering, switching, or blending between predefined anim tions based on conditions, input, or game state. At a minimum, students should develop and demonstrate: (a) two or more animation states (e.g., Idle, Run, Jump, Attack), and (b) explicit logic or conditions that control transitions between those states. For 2D games, animation control may involve frame-based sequences or sprite states. For 3D games, animation control may involve state machines, animation blending, or parameter-driven transitions.

Students are not expected to design original animations, apply artistic animation principles (e.g., squash and stretch, anticipation), or use professional animation tools.

Practice(s): CT7, CT8	**Disposition(s):** Creativity, Resourcefulness

S2-GMD-DD-03: Construct in-game artificial intelligence to control NPC behavior.

Boundary Statement: Students should construct rule-based in-game intelligence to control NPC behavior, using explicit logic structures rather than learning or adaptive algorithms. At a minimum, the constructed behavior should include: (a) two or more distinct states or behaviors (e.g., idle, patrol, chase, evade), and (b) at least one explicit transition condition that determines when the NPC changes from one state or behavior to another (e.g., player proximity, health level, timer, game state). Students should articulate how states, transitions, and conditions work together to produce responsive and believable NPC behavior and demonstrate how changes to logic affect gameplay or player experience.

Students are not expected to implement machine learning, neural networks, pathfinding optimiz tion, probabilistic reasoning, or adaptive systems that learn from player behavior. Students are not expected to go beyond deterministic, rule-based decision-making.

Practice(s): CT7, CT8	**Disposition(s):** Persistence, Critical Thinking

S2-GMD-PX-04: Evaluate game interactions using diverse input devices to enhance immersion and player experience.

Boundary Statement: Students should evaluate player immersion by analyzing how specific design elements influen e player attention, feedback, and engagement within a game experience. Evaluation should reference at least two concrete immersion factors, such as: (a) consistency of game rules and mechanics (e.g., predictable cause-and-effect), (b) responsiveness and feedback (e.g., timely visual, audio, score-based responses), (c) alignment between player actions and system outcomes, (d) clarity of goals, progress indicators, or challenge pacing, or (e) how different input devices (e.g., keyboard and mouse, game controller, motion controls) affect player comfort and accuracy. Students should support their evaluation using observable evidence, such as gameplay scenarios, comparisons of design alternatives, playtesting observations, or documented design revisions.

Students are not expected to conduct formal user experience research, apply psychological immersion models, or quantify engagement using analytics platforms. Students are not expected to evaluate every possible input device or build custom input hardware.

Practice(s): CT9, HCD10	**Disposition(s):** Reflectiveness, Critical Thinkin

S2-GMD-PX-05: Evaluate two versions of a game using defined metrics to determine which performs better.

Boundary Statement: Students should use comparative usability methods to evaluate and refine game designs by critiquing two or more design alternatives that differ in a specific, intentional way (e.g., control scheme, feedback style, difficulty pacing, interf e layout). Evaluation should build on prior usability testing concepts and be grounded in qualitative or simple quantitative evidence, such as: (a) observed user behavior during gameplay, (b) structured user feedback or reflection, (c) ounts of successful or unsuccessful interactions (e.g., level completion, errors, retries), or (d) documented friction points or player confusion. Students should justify which version better supports the intended player experience based on this evidence and explain how the findings i formed design revisions.

Students are not expected to conduct formal experiments, apply statistical analysis, or implement industry-specific testing techniques such as full A/B testing pipelines. Students a e not expected to use professional analytics tools or recruit participants beyond their peer group.

Practice(s): HCD10, HCD11	**Disposition(s):** Critical Thinking, Reflectivenes

S2-GMD-AR-06: Analyze game system architectures that support scalability, maintainability, and performance.

Boundary Statement: Students should analyze architectural design principles of game systems to examine issues of scalability, maintainability, and performance. This includes reasoning about modular design, separation of concerns, interfaces between systems, and dependency management within a game. Students should analyze how architectural decisions affect: (a) system complexity, (b) ease of iteration or expansion, (c) performance across platforms (e.g., desktop, mobile, web), and (d) player experience. Examples may include separating input handling from gameplay logic, organizing rendering, audio, and networking systems, or designing game components that can be reused or extended. Students may examine hardware considerations (e.g., GPUs or cloud-based execution) as they relate to architectural trade-offs

Students are not expected to implement changes to improve scalability, maintainability, or performance. Students do not need to consider engine-level optimizations, manage distributed infrastructure, or perform detailed performance benchmarking. Students are not expected to implement hardware-specific optimiz tions or treat hardware considerations as implementation requirements.

Practice(s): IC3, CT6	**Disposition(s):** Critical Thinking, Resourcefulness

S2-GMD-PP-07: Create a game utilizing ethical principles in the design and development process.

Boundary Statement: Students should create a game project or prototype using ethical design principles and building on foundational ethical awareness. Students should select a focused subset of ethical considerations (e.g., accessibility, inclusivity, bias mitigation, player well-being, data protection) and articulate how those principles informed specific design decisions. Students should integr te ethical reasoning into their design process rather than treating it as a standalone add-on. The project may be a prototype, vertical slice, or partial implementation, depending on instructional context. Students should provide written or verbal justific tion explaining how their design choices address the selected ethical principles.

Students are not expected to implement every ethical domain, build fully polished games, or conduct formal ethics audits.

Practice(s): ESR1, IC4	**Disposition(s):** Reflectiveness, C eativity

S2-GMD-PP-08: Collaborate effectively in a project team by defining and executing assigned roles, communicating clearly, and managing shared resources to deliver a game project.

Boundary Statement: Students should develop a game project in a collaborative team environment by communicating effectively, managing their assigned roles and responsibilities, and contributing to shared decision-making. Students should listen to and integrate teammates' ideas, work through disagreements constructively, and be willing to contribute outside their primary expertise when it benefits the team. For example, a student who p efers coding might help brainstorm art concepts with a teammate who is more visually inclined, demonstrating willingness to engage with different aspects of game development to support the team's goals.

Students are not expected to use formal project management methodologies or maintain perfect team dynamics throughout the project.

Practice(s): IC3, IC5	**Disposition(s):** Persistence, Sense of Belonging

Physical Computing

Specialty I

S1-PHY-HC-01: Construct an electrical circuit to power and control a physical computing device.

Boundary Statement: Students should create and troubleshoot a simple electrical circuit that powers and controls physical computing devices, with the aid of schematic diagrams. For example, students could build a circuit with a microcontroller, a resistor, and an LED, use a multimeter to check voltage and current, draw a corresponding schematic diagram, and troubleshoot if the LED doesn't light as expected by checking for loose connections or incorrect component orientation.

Students are not expected to design circuits with multiple integrated circuits or design and etch custom-printed circuit boards.

Practice(s): CT7, CT8	**Disposition(s):** Resourcefulness, Critical Thinking

S1-PHY-IO-02: Integrate sensors, actuators, and peripherals into a physical computing system.

Boundary Statement: Students should select, connect, and program sensors and outputs to enable physical computing systems to respond to real-world data. Outputs may include lights, sound, displays, wearables, environmental indicators, or simple mechanical motion.

Students are not expected to integrate large numbers of components, use industrial-grade hardware, manage complex wiring or power requirements, or build systems for mission-critical use.

Practice(s): CT6, CT8	**Disposition(s):** Persistence, Creativity

S1-PHY-DD-03: Develop software to control a physical device using an iterative design process.

Boundary Statement: Students should iteratively design and develop programs that directly control physical system behavior, such as reading sensor input, activating outputs, or changing system states. Programming may be implemented using block-based, hybrid, or text-based environments, with emphasis on the observable connection between code and physical behavior.

Students are not expected to fabricate hardware, implement complex algorithms, or use professional embedded development environments.

Practice(s): CT8, HCD11	**Disposition(s):** Creativity, Resourcefulness

S1-PHY-CI-04: Use IoT devices to collect sensor data and transmit it locally using device-to-device or device-to-gateway communication.

Boundary Statement: Students should use IoT devices to collect and transmit sensor data wirelessly using local communication methods, such as device-to-device or device-to-gateway communication. Evidence of learning should demonstrate meaningful use of transmitted data, such as visualization, logging, or influencing system behavio , not merely establishing a connection.

Students are not expected to configu e cloud services, manage servers, rely on school Wi-Fi access, or implement custom communication protocols.

Practice(s): CT8, HCD12	**Disposition(s):** Resourcefulness, Persistence

S1-PHY-CI-05: Implement IoT communication by connecting physical devices with protocols, applying security practices.

Boundary Statement: Students should deploy multiple connected devices to share data or actions within a system. Communication methods may include serial, radio, Bluetooth, Wi-Fi, or low-power wide-area networking (e.g., LoRa-type systems) where available. Students should investigate basic trade-offs, such as range, d ta size, latency, or power use.

Students are not expected to design networking infrastructure, optimize performance, or deploy large-scale IoT systems.

Practice(s): IC4, CT9	**Disposition(s):** Critical Thinking, Resourcefulness

S1-PHY-PP-06: Use a version control system to coordinate development in a physical computing project.

Boundary Statement: Students should use a version control system to manage and coordinate code development when collaborating on physical computing projects. Students should create branches for different features or components, commit changes with descriptive messages, merge contributions from team members, and resolve basic merge conflicts. Students should understand how source control enables collaborative work by tracking who made what changes and allowing team members to work on different parts of the project simultaneously. For example, a team building a weather station could use a source control system to manage separate branches for sensor data collection code and display control code, with each team member committing their work and the team merging contributions into a main branch to create the integrated system.

Students are not expected to set up and administer repository servers or hosting platforms. Students are not expected to resolve complex merge conflicts involving extensive ode changes across many files. Students a e not expected to use advanced features like rebasing, cherry-picking, or managing complex branching strategies.

Practice(s): IC3, IC5	**Disposition(s):** Reflectiveness, esourcefulness

S1-PHY-PP-07: Evaluate social, technical, and sociotechnical impacts of a physical computing system.

Boundary Statement: Students should assess ethical or social impacts (e.g., accessibility, usability, safety, equity, sustainability) of a physical computing system. Evaluation should be grounded in a specific system and should eference design decisions.

Students are not expected to conduct comprehensive global impact analyses, perform policy evaluations, or address all ethical dimensions simultaneously. Broader global or systemic impacts may be introduced as aspirational extensions, not required outcomes.

Practice(s): ESR1, HCD10	**Disposition(s):** Reflectiveness, Critical Thinkin

S1-PHY-PP-08: Evaluate the security implications of a physical computing system, including data privacy, unauthorized access, and potential vulnerabilities.

Boundary Statement: Students should evaluate security threats and vulnerabilities specific to physical computing devices. Students should assess how physical access to a device can lead to security breaches and how to protect against them. For example, students might evaluate security measures like strong authentication (e.g., using a keypad and unique PIN), data encryption for communication between device and servers, and physical safeguards to prevent tampering.

Students are not expected to perform penetration testing or conduct systematic security audits.

Practice(s): IC3, HCD10	**Disposition(s):** Critical Thinking, Reflectivenes

Specialty II

S2-PHY-HC-01: Develop an electromechanical system using a CAD tool for design and testing.

Boundary Statement: Students should design and refine omplex physical computing systems using Computer Aided Design (CAD) software. Complex systems integrate multiple inputs, outputs, and behaviors to meet defined goals. For example, students ould design and build a system that integrates mechanical components (e.g., motors, actuators, structural elements) with electrical components. They could use CAD software to model and refine their designs. Students ould design a robotic arm that picks up and moves objects, using CAD software to design and simulate the arm's mechanical structure before building it.

Students are not expected to master advanced electronics theory or industrial automation practices.

Practice(s): CT7, CT8	**Disposition(s):** Creativity, Critical Thinking

S2-PHY-IO-02: Design a closed-loop feedback control system to meet a desired outcome using varying sensor types.

Boundary Statement: Students should develop a closed-loop system selecting from various sensor options to implement system behavior meeting a desired outcome. Closed-loop feedback may be conceptual, simulated, or partially implemented, depending on instructional context. Students should assess trade-offs (a curacy, responsiveness, cost, and constraints) during design.

Students are not expected to design or deploy fully optimized control systems, implement complex control algorithms, perform exhaustive quantitative analyses across multiple sensor dimensions, or build systems requiring specialized hardware, industrial-grade components, or mission-critical reliability.

Practice(s): HCD10, HCD12	**Disposition(s):** Critical Thinking, Resourcefulness

S2-PHY-DD-03: Develop an application that extends functionality and user engagement with physical devices.

Boundary Statement: Students should develop applications that extend how users interact with physical systems, such as configur tion options, feedback displays, data-informed behavior, or app-connected interaction. Students should focus on clarity of design, user interaction, and purposeful extension of functionality.

Students are not expected to implement fully autonomous systems, advanced control algorithms, multilayer feedback architectures, or professional-grade embedded software pipelines.

Practice(s): CT8, HCD10	**Disposition(s):** Persistence, Resourcefulness

S2-PHY-CI-04: Develop an IoT system to manage the collection and transmission of data and enable remote monitoring and control of a physical system.

Boundary Statement: Students should develop IoT systems that transmit, receive, and use data across connected physical devices, using communication structures appropriate to the application. Connectivity may be local or extended, and students should justify communication choices based on range, latency, data size, power use, or reliability. Systems may be conceptual, simulated, or implemented, depending on instructional context.

Students are not expected to deploy enterprise-scale IoT platforms or manage persistent cloud infrastructure.

Practice(s): CT9, HCD10	**Disposition(s):** Critical Thinking, Resourcefulness

S2-PHY-CI-05: Develop an embedded network, evaluating trade-offs among network protocols, security measures, and scalability.

Boundary Statement: Students should coordinate data from multiple physical devices using centralized or distributed approaches. Central coordination might be implemented using a local computer, single-board device, or simple script. Students might design simple communication protocols and reason about message structure and flo . Communication might be simulated or implemented, depending on instructional context. Students should connect networking and security concepts to physical system behavior.

Students are not required to use cloud servers or web services. Students are not expected to implement advanced cryptographic protocols, perform detailed packet-level analysis, configu e enterprise security systems, or master low-level networking theory.

Practice(s): IC4, CT8	**Disposition(s):** Critical Thinking, Reflectivenes

S2-PHY-PP-06: Apply a project management methodology to collaboratively design, develop, and test a physical computing project.

Boundary Statement: Students should apply project management methodologies (e.g., agile frameworks) to collaboratively manage a physical computing project from conception through delivery. They should share responsibilities, resolve conflicts equitabl , and communicate progress and outcomes. For example, students might use a Kanban board to track tasks (e.g., prototyping, circuit design, programming the microcontroller, user testing) and hold regular stand-up meetings to discuss progress and resolve issues.

Students are not expected to use professional project management software or lead projects with multiple teams and external stakeholders.

Practice(s): IC4, IC5	**Disposition(s):** Persistence, Resourcefulness

S2-PHY-PP-07: Evaluate how physical computing technologies shape social processes, communities, power, and equity in global society.

Boundary Statement: Students should evaluate how physical computing systems shape social, technical, and human outcomes through specific design decisions in their p ojects. Students might conduct small-scale, local research (e.g., surveys, interviews, observations) to identify relevant problems. Evaluation should focus on one or two impact dimensions (e.g., accessibility, sustainability, safety) and include a clear viability criterion such as cost, usability, or maintenance.

Students are not expected to conduct formal sociological studies or global policy analysis.

Practice(s): ESR1, ESR2	**Disposition(s):** Critical Thinking, Reflectivenes

S2-PHY-PP-08: Develop a physical computing solution for diverse users using the engineering design process.

Boundary Statement: Students should develop physical computing solutions using the engineering design process to plan, build, test, and refine a physical omputing project. The process should be evidenced through design sketches, written plans, testing notes, and reflections tied di ectly to technical decisions. Students should demonstrate an awareness of the needs of diverse users and address them in the design process.

Students are not expected to follow formal project management frameworks or use professional version control systems.

Practice(s): ESR1, HCD10	**Disposition(s):** Reflectiveness, Sense of Belonging

Software Development

Specialty I

S1-SWD-DD-01: Implement linear data structures to organize and access collections of data when solving computational problems.

Boundary Statement: Students should select appropriate linear data structures to store and manage data to solve a computational problem. Students should use arrays for fi ed-size collections accessed by position, dynamic arrays (lists) for collections that grow or shrink, linked lists for efficient insertio and deletion, stacks for last-in-first-out a cess, and queues for first-in-first-out cess. For example, students could use a dynamic array to store temperature readings collected over time, a stack to implement an undo feature in an application, or a queue to manage print jobs in the order they were submitted. Students should use built-in APIs or libraries and their associated methods.

Students are not expected to implement these data structures or their operations (e.g., insert, delete, search) from scratch. Students are not expected to use dictionaries, trees, graphs, or heaps.

Practice(s): CT6, CT8	**Disposition(s):** Critical Thinking, Persistence

S1-SWD-DD-02: Design software that accounts for complexity through abstraction.

Boundary Statement: Students should apply design principles that promote maintainable and extensible software. Students should make intentional decisions about program structure, data organization, and relationships between components. They should consider how their software will handle growth (e.g., more users, more data, more features) and apply abstraction (e.g., data abstractions, procedural abstraction, recursion, object-oriented abstraction) to manage complexity. For example, students could design a social media application that organizes data efficiently t support adding more users and posts without rewriting core functionality.

Students are not expected to handle scalability challenges requiring specialized infrastructure (e.g., multiple servers, distributed databases). Students are not expected to perform algorithm complexity analysis or optimize code for speed and memory usage.

Practice(s): CT7, CT8	**Disposition(s):** Resourcefulness, Critical Thinking

S1-SWD-DD-03: Use integrated development environment (IDE) features to streamline software development, including code editing, debugging, version control, and project management.

Boundary Statement: Students should use the core features of an IDE to increase their efficienc and effectiveness in developing software. This includes utilizing the IDE's built-in tools for syntax highlighting, autocomplete, and real-time error detection while writing code. Students should use the integrated debugger to set breakpoints, step through code, and inspect variables to diagnose and fix logical errors. Students should also be able to interact with a version control system directly within the IDE to manage changes to their code, commit new versions, and collaborate on projects. Students should also use the IDE's project management features, such as organizing files and olders and managing dependencies, to maintain a structured project.

Students are not expected to deeply understand the underlying mechanisms of these tools (e.g., how the debugger works at a machine level), set up complex IDE configur tions from scratch, or configu e custom build processes.

Practice(s): IC4, CT8	**Disposition(s):** Critical Thinking, Resourcefulness

S1-SWD-UX-04: Refine a user interface design using accessibility and responsive design tools.

Boundary Statement: Students should evaluate the effectiveness of user interface design choices in a digital product based on established principles such as contrast, alignment, proximity, and readability. For example, students could analyze a mobile app's user interface using an accessibility design tool and provide a written critique of how element alignment and color contrast affect readability for users with low vision.

Students are not expected to design graphics, nor are they expected to create design systems that specify detailed guidelines for all visual and interaction elements across a product.

Practice(s): HCD10, HCD11	**Disposition(s):** Critical Thinking, Reflectivenes

S1-SWD-UX-05: Create software that serves intended users and contexts using human-centered design principles.

Boundary Statement: Students should conduct heuristic reviews (i.e., systematic evaluations against established design criteria such as usability, consistency, accessibility, and user control) to assess how well the software serves diverse users, including variations in ability, language, and cultural context. Students should use these evaluations to inform design decisions and recommendations.

Students are not expected to conduct user testing or collect feedback from actual users. Students are not expected to create accessibility solutions beyond applying established guidelines (e.g., Web Content Accessibility Guidelines [WCAG]).

Practice(s): ESR1, HCD12	**Disposition(s):** Reflectiveness, Critical Thinkin

S1-SWD-TR-06: Use AI-assisted IDE tools or features to understand unfamiliar code and identify errors during debugging.

Boundary Statement: Students should use AI-assisted IDE tools or features for code comprehension and debugging tasks. This includes generating code summaries and documentation to understand unfamiliar or legacy code and using intelligent code completion, error detection, and refactoring suggestions during debugging. For example, students could use AI assistance to generate a summary explaining an unfamiliar function's purpose, inputs, and outputs before fixing a bug. Students should apply existing, commercially available, or open-source AI IDE features.

Students are not expected to manually configu e or fine-tune the underlying machine learning models within the IDE (e.g., modifying the weights or architecture of the code-generation transformer model), integrate external non-IDE-based AI tools, or develop custom AI extensions for their IDE.

Practice(s): IC3, CT9	**Disposition(s):** Resourcefulness, Critical Thinking

S1-SWD-TR-07: Design a test plan that exercises program functionality, including edge cases, error conditions, and varied user inputs.

Boundary Statement: Students should create thorough and systematic test cases from both human and AI-generated sources. The plan should include a variety of test case types, such as unit tests and integration tests. Tests should cover valid input, edge cases, error conditions, user inputs, and other scenarios based on software use cases. Students should apply test cases systematically to a software project involving multiple functions and data structures.

Students are not expected to implement automated testing frameworks or manage quality assurance processes for large systems, nor to perform adversarial security testing (e.g., penetration testing). They are also not expected to design, train, or modify any underlying machine learning models used to generate AI test case recommendations.

Practice(s): CT7, CT9	**Disposition(s):** Critical Thinking, Reflectivenes

S1-SWD-PP-08: Collaborate to develop software that reflects diverse perspectives.

Boundary Statement: Students should work in teams on a shared software project, with team members contributing to different components. They should use version control tools to manage workflows, including branching, eviewing code, resolving merge conflicts, and integr ting work through pull requests. Teams should establish inclusive norms that actively incorporate diverse perspectives on code design, user experience, and ethical considerations, including respecting intellectual property through appropriate attribution, licensing, and responsible reuse of code and media. For example, students might develop a multi-module program using version control, where pull requests are reviewed not only for technical quality but also for usability and ethical implications, ensuring that insights drawn from varied areas of expertise (e.g., user communities, UI design, accessibility, data privacy) inform final decisions

Students are not expected to administer version control servers, manage repository security policies, or integrate version control with continuous integration/continuous deployment pipelines.

Practice(s): ESR2, IC5	**Disposition(s):** Sense of Belonging, Reflectivenes

Specialty II

S2-SWD-DD-01: Apply associative and hierarchical data structures to solve computational problems.

Boundary Statement: Students should select and apply appropriate associative and hierarchical data structures including dictionaries (hash tables), trees, graphs, and heaps to solve computational problems. Students should use built-in or library implementations and their associated methods. Students should use dictionaries for efficient ey-value lookups, trees for hierarchical data or ordered collections, graphs for modeling networks and relationships, and heaps for priority-based access. Students should explain relative efficiency trade ffs in practical terms (e.g., which oper tions are faster, which use less memory). For example, students might use a graph structure to find the shortest path in a navigation app, a dictionary to quickly look up student records by ID, or a tree to organize a file system hiera chy.

Students are not expected to implement these data structures or their operations from scratch. Students are not expected to use Big O notation to express time and space complexity.

Practice(s): CT6, CT7	**Disposition(s):** Persistence, Critical Thinking

S2-SWD-DD-02: Develop a prototype using diverse user personas and user journey maps to guide design decisions.

Boundary Statement: Students should create user personas and user journey maps to guide ideation and prototyping and justify design decisions. User personas represent different types of users and their characteristics, needs, and goals. User journey maps visualize how users interact with a product from initial contact through achieving their goals, highlighting pain points and opportunities for improvement. For example, when designing an educational app, students might create personas representing different user types (e.g., a student with a learning disability, a highly motivated student, a parent) and develop user journey maps for each persona to ensure the design addresses diverse needs.

Students are not expected to conduct market research to validate personas or produce fully functional applications.

Practice(s): HCD10, HCD12	**Disposition(s):** Creativity, Critical Thinking

S2-SWD-DD-03: Develop a project in iterative cycles, documenting changes and the rationale for each cycle.

Boundary Statement: Students should develop a software project using an iterative process. This includes breaking the project into multiple distinct cycles of planning, implementation, testing, review, and documentation that continuously refine and build p oject features. Students should document each cycle with a clear record of changes made and the reasoning behind them (e.g., bug fix, new feature, response to user feedback). For example, in one cycle a student might build a basic user login system, in the next cycle add a user profile eature, and in a third cycle implement a search function.

Students are not expected to use formal, complex project management methodologies like Agile or Scrum and their specific eremonies (e.g., daily stand-ups, sprint retrospectives).

Practice(s): IC4, HCD11	**Disposition(s):** Persistence, Reflectivenes

S2-SWD-DD-04: Modify an existing algorithm to improve efficiency, considering factors such as data structures and algorithmic paradigms.

Boundary Statement: Students should identify opportunities to improve efficiency in terms f time or space complexity in algorithms. This involves understanding how different data structures (e.g., arrays, dictionaries, trees) impact performance and selecting a more suitable one. Students should apply algorithmic paradigms (e.g., divide and conquer) to break a problem into smaller subproblems. Students should use this information to modify and optimize existing solutions, measuring performance. For example, a student might improve a linear search algorithm for a large, sorted dataset by using a more efficient ta structure and applying binary search.

Students are not expected to invent new algorithmic paradigms.

Practice(s): CT6, CT7	**Disposition(s):** Critical Thinking, Persistence

S2-SWD-UX-05: Evaluate the user experience to improve usability.

Boundary Statement: Students should conduct usability evaluations on digital interfaces by observing user interactions, collecting qualitative and quantitative feedback, and documenting findings in a structured format. For example, students could create a user survey or think-aloud protocol to test a peer's program or a real-world app and compile a report outlining specific usability issues (e.g., confusing navigation, inaccessible features).

Students are not expected to obtain research ethics approval, recruit participants from outside of their peer group, or use specialized usability testing tools (e.g., eye-tracking software). Students are not expected to perform statistical analysis beyond calculating basic descriptive statistics for survey questions (e.g., average ratings, distribution of responses to Likert-type items).

Practice(s): IC3, HCD10	**Disposition(s):** Critical Thinking, Reflectivenes

S2-SWD-UX-06: Create a user-friendly, accessible, and responsively designed interface.

Boundary Statement: Students should create user interfaces for digital products that are usable, accessible, and responsive across different devices and screen sizes by applying UI design principles and tools. For example, students could develop a web application that adapts its layout and functionality to be effective on smartphones, tablets, and desktop computers, ensuring that elements are navigable and legible for users with different needs.

Students are not expected to create applications that function on every possible device and operating system or master responsive design frameworks.

Practice(s): CT8, HCD10	**Disposition(s):** Critical Thinking, Creativity

S2-SWD-TR-07: Use an IDE to test and refine a complex software project.

Boundary Statement: Students should use IDE tools and features, including AI-assistive tools, to enhance the testing and refinement phases f a complex software project. This includes features for automated test generation (e.g., unit test boilerplate, test cases), identifying redundant or inefficien tests and receiving real-time suggestions for code optimization and performance improvements during the refinement phase. Students ould write a new class, use an AI tool to generate a basic suite of unit tests for it, review and critique the generated tests for test coverage, and then accept the IDE's AI-driven recommendation to refactor a slow loop into a more performant structure.

Students are not expected to perform deep-dive performance profiling, interp et low-level hardware performance counters, or manually implement complex statistical analysis of test results derived from large-scale, enterprise-level performance testing tools. The "complex software project" should remain within the scope of advanced high school application development (e.g., multi-module, multifile application with external libraries).

Practice(s): IC4, CT9	**Disposition(s):** Critical Thinking, Persistence

S2-SWD-TR-08: Debug a complex software project to identify, isolate, and fix errors.

Boundary Statement: Students should systematically apply formal debugging methodologies (e.g., divide and conquer, binary search, backtracking) to software projects involving multiple files, libraries, and external dependencies. This includes using advanced IDE debugger tools and features such as AI-assisted debuggers, setting conditional breakpoints, inspecting the call stack and watch window for complex data structures, and remote debugging. For example, students could debug a multi-module program with a subtle bug by reproducing the error, isolating the fault by tracing state changes across functions, and then implementing a fix. Students should ocus on logical program errors.

Students are not expected to debug at the operating system level, such as analyzing assembly code, directly manipulating memory registers, or performing post-mortem memory dump analysis for errors in production operating systems or hardware drivers.

Practice(s): IC3, CT9	**Disposition(s):** Persistence, Critical Thinking

S2-SWD-PP-09: Apply an industry-standard software development process to plan and deliver a project while prioritizing equity and justice.

Boundary Statement: Students should select and apply an industry-standard software development process (e.g., Agile Scrum, Kanban, Waterfall) to a medium- to large-scale project. Students should effectively use project-management tools to manage tasks, track progress, and communicate with team members. Students should demonstrate that equity and justice are non-negotiable priorities that override technical or budgetary pressures and articulate and justify project modific tions based on ethical or societal impacts. For example, a student team could adjust a project timeline and scope to prioritize features that improve accessibility for users with disabilities or mitigate algorithmic bias, even if it delays the release or requires refactoring complex code. Students should engage in development projects of appropriate complexity for high school, with a focus on applying sound processes and prioritizing ethical considerations.

Students are not expected to manage the budget or financial esources of the project, use advanced proprietary or enterprise-level project management software, or manage external stakeholders beyond their classroom or school environment.

Practice(s): ESR1, IC4	**Disposition(s):** Critical Thinking, Reflectivenes

X+CS

S1-XCS-XC-01: Explain connections between CS concepts and practices and those from a non-CS discipline (X).

Boundary Statement: Students should identify and explain, at a conceptual level, how foundational computer science principles, such as data representation, algorithmic thinking, or abstraction, support understanding or problem-solving within a non-CS field (X) such as e onomics, linguistics, or art history. For example, students in a literature class could explain how word frequency analysis relies on data representation to convert text into numerical features and algorithmic processing to identify patterns across a large corpus, enabling insights about authorial style or thematic emphasis that would be difficult to detect t ough close reading alone.

Students are not expected to develop models that computationally solve advanced domain-specific p oblems.

Practice(s): IC3, CT6	**Disposition(s):** Critical Thinking, Creativity

S1-XCS-XC-02: Formulate a computational solution to a problem in a non-CS discipline (X) using computational thinking.

Boundary Statement: Students should identify a problem in another discipline (X) that can be addressed using computational thinking. They decompose the problem into subproblems, identify patterns or important variables (abstraction), and develop step-by-step procedures (algorithms) that describe a possible solution. For example, biology students studying population dynamics could model how a prey population changes over time by identifying key variables (e.g., birth rate, predation rate, available resources) and designing an algorithm that describes population changes across generations. Students should focus on documenting the design process using tools such as pseudocode or flowcharts

Students are not expected to create, test, and refine a full omputational solution.

Practice(s): CT7, CT8	**Disposition(s):** Critical Thinking, Resourcefulness

S1-XCS-XC-03: Analyze a computer science technology or approach used in a non-CS discipline (X).

Boundary Statement: Students should investigate a specific omputer science technology or approach, detailing the foundational computing concepts involved, how it was developed, and its real-world impact on the chosen non-CS discipline. For example, students could analyze how machine learning algorithms are used in medical imaging to detect tumors, detailing the training process using labeled image datasets, the pattern recognition methods that identify abnormalities, and how this approach has improved diagnostic accuracy and speed.

Students are not expected to reproduce the innovation, implement it from scratch, or conduct comprehensive analyses of its societal, economic, or policy implications beyond describing its direct impact on the discipline.

Practice(s): ESR1, CT6	**Disposition(s):** Critical Thinking, Resourcefulness

S1-XCS-XC-04: Model relationships within a non-CS discipline using data, visualizations, and computational methods.

Boundary Statement: Students should select and apply computational methods (e.g., sorting and filtering, g ouping and aggregating, calculating new variables) to structure, analyze, and visualize a real-world dataset from a non-CS discipline (X), revealing patterns and relationships within that discipline. For example, social science students could organize raw voting records into a structured dataset, calculate voter turnout by district, and create visualizations to reveal geographic patterns in political participation.

Students are not expected to develop novel computational methods or algorithms. Students are not expected to work with datasets that require memory or processing power beyond what is available on their personal devices.

Practice(s): CT6, CT7	**Disposition(s):** Resourcefulness, Critical Thinking

S1-XCS-XC-05: Assess how well an algorithm solves problems within a non-CS discipline (X) by evaluating its accuracy, efficiency, and relevance to the intended goal.

Boundary Statement: Students should evaluate and select appropriate metrics for assessing accuracy, efficien , and relevance based on the algorithm being analyzed. Students should also explain trade-offs between a curacy and efficiency and justify the suitability f the algorithm in the context of the discipline (X). For example, students in a geography class could assess a routing algorithm used in a mapping application by evaluating whether it finds the shortest p th accurately, how quickly it processes routes as the number of intersections increases, and whether it accounts for factors that matter in the local context (e.g., road closures, pedestrian-only zones, accessibility for wheelchair users).

Students are not expected to formally derive algorithmic complexity using Big O notation.

Practice(s): ESR1, CT6	**Disposition(s):** Critical Thinking, Reflectivenes

References

American Association of School Librarians. (2018). *National school library standards for learners, school librarians, and school libraries*. ALA Editions.

AI4K12. (2022). *AI4K12 guidelines*. https://ai4k12.org/gradeband-progression-charts

American School Counselor Association. (2021). *ASCA student standards: Mindsets & behaviors for student success*. https://www.schoolcounselor.org/Standards-Positions/Standards/ASCA-Mindsets-Behaviors-for-Student-Success

Association for Computing Machinery (ACM). (2018). *ACM code of ethics and professional conduct*. https://www.acm.org/code-of-ethics

Bargagliotti, A., Franklin, C., Arnold, P., Gould, R., Johnson, S., Perez, L., & Spangler, D. A. (2020). *Pre-K–12 guidelines for assessment and instruction in statistics education II (GAISE II): A framework for statistics and data science education*. American Statistical Association. https://www.amstat.org/asa/files/pdfs/GAISE/GAISEIIP_eK-12_Full.pdf

Caspersen, M. E., Diethelm, I., Gal-Ezer, J., McGettrick, A., Nardelli, E., Passey, D., Rovan, B. & Webb, M. (2022). *Informatics reference framework for school. Informatics for All.* https://www.informaticsforall.org/the-informatics-reference-framework-for-school-release-february-2022/

Computer Science Teachers Association. (2026a). *Algorithms and programming: Literature review*. https://csteachers.org/k12standards/research

Computer Science Teachers Association. (2026b). *Artificial intelligence: Design brief for standards writers*. https://csteachers.org/k12standards/research

Computer Science Teachers Association. (2026c). *Comparison of CSTA Standards to other standards and frameworks*. https://csteachers.org/k12standards/research

Computer Science Teachers Association. (2026d). *Computing and society: Literature review*. https://csteachers.org/k12standards/research

Computer Science Teachers Association. (2026e). *Computing systems and security: Design brief for standards writers*. https://csteachers.org/k12standards/research

Computer Science Teachers Association. (2026f). *Computing systems and security: Literature review*. https://csteachers.org/k12standards/research

Computer Science Teachers Association. (2026g). *Data and analysis: Design brief for standards writers*. https://csteachers.org/k12standards/research

Computer Science Teachers Association. (2026h). *Data and analysis: Literature review*. https://csteachers.org/k12standards/research

Computer Science Teachers Association. (2026i). *Standards alignment review*. https://csteachers.org/k12standards/research

Computer Science Teachers Association. (2026j). *Structure: Design brief for standards writers.* https://csteachers.org/k12standards/research

CSTA & AI4K12. (2025). *AI learning priorities for all K–12 students*. Computer Science Teachers Association. https://csteachers.org/ai-priorities

CSTA & IACE. (2024). *K–12 Computer Science Standards comparison report: Examining the similarities and differences in state-adopted K–12 Computer Science Standards and the CSTA K–12 Standards, revised 2017*. Association for Computing Machinery.

CSTA & IACE. (2025). *International Computer Science Standards comparison report: Examining the similarities and differences in K–12 Computer Science Standards across seven global locations*. Association for Computing Machinery.

CSTA, IACE, ACM, Code.org, College Board, CSforAll, & ECEP Alliance. (2024). *Reimagining CS pathways: Every student prepared for a world powered by computing*. Association for Computing Machinery. https://doi.org/10.1145/3678016

CYBER.ORG. (2021). *K–12 cybersecurity learning standards*. https://cyber.org/standards

Data Science 4 Everyone. (2025). *K–12 data literacy and data science learning progressions*. https://datasciencelearning.org

Interaction Design Foundation (IDF). (n.d.). *Human-centered design (HCD)*. https://www.interaction-design.org/literature/topics/human-centered-design

International Technology and Engineering Educators Association (ITEEA). (2020). *Standards for technological and engineering literacy: The role of technology and engineering in STEM education*. https://www.iteea.org/stel

K–12 Computer Science Framework Steering Committee. (2016). *K–12 computer science framework*. http://www.k12cs.org

Kumar, A. M., Raj, R. K., Aly, S. G., Anderson, M. D., Becker, B. A., Blumenthal, R. L., Eaton, E., Epstein, S. L., Goldweber, M., Jalote, P., et al. (2024). *Computer science curricula 2023*. Association for Computing Machinery. https://doi.org/10.1145/3664191

Learning for Justice. (n.d.). *Social justice standards: The Learning for Justice anti-bias framework*. https://www.learningforjustice.org/frameworks/social-justice-standards

National Association of State Directors of Career Technical Education Consortium/National Career Technical Education Foundation. (2012). *Common career technical core*. https://osse.dc.gov/sites/default/files/dc/sites/osse/public_tion/attachments/CCTC%20Standards.pdf

National Career Development Association. (2024). *National career development guidelines framework*. https://ncda.org/aws/NCDA/pt/sp/ncdg_framework

National Coalition for Core Arts Standards. (n.d.). *National core arts standards: A conceptual framework for arts learning*. National Core Arts Standards. https://www.nationalartsstandards.org/sites/default/files_Conceptual%20Framework%2007-21-16.pdf

National Council for the Social Studies (NCSS). (2010). *National curriculum standards for social studies: A framework for teaching, learning, and assessment*.

National Governors Association Center for Best Practices (NGA Center) & Council of Chief State School Officers (CCSSO). (2010a). *Common Core State Standards for English language arts & literacy in history/social studies, science, and technical subjects*. https://www.thecorestandards.org/ELA-Literacy/

National Governors Association Center for Best Practices (NGA Center) & Council of Chief State School Officers (CCSSO). (2010b). *Common Core State Standards for mathematics*. https://corestandards.org/wp-content/uploads/2023/09/Math_Standards1.pdf

National Institute of Standards and Technology (NIST). (2021). *Human centered design (HCD)*. https://www.nist.gov/itl/iad/visualization-and-usability-group/human-factors-human-centered-design

National Q-12 Education Partnership. (2025). *QISE key concepts for early learners: K–12 framework*. https://q12education.org/learning-materials/learning-materials-framework

NGSS Lead States. (2013). *Next generation science standards: For states, by states*. The National Academies Press.

Office of Head Start, U.S. Department of Health and Human Services. (2015). *Head Start early learning outcomes framework: Ages birth to five*. https://headstart.gov/school-readiness/article/head-start-early-learning-outcomes-framework

Partnership for 21st Century Skills. (2009). *P21 framework definitions*. https://files.eric.ed.gov/fulltext/ED519462.pd

Phelps, D., Lachney, M., Ryoo, J., Santo, R., Koressel, J., & Twarek, B. (2025). *Computing impacts and ethics in the CSTA standards: A synthesis of expert perspectives*. https://csteachers.org/k12standards/research

Santo, R., Ryoo, J., & Lachney, M. (2025). *Amplifying social impacts of computing standards: Reference materials for standards writers*. https://csteachers.org/k12standards/research

Seehorn, D., Carey, S., Fuschetto, B., Lee, I., Moix, D., O'Grady-Cunniff, ., Owens, B. B., Stephenson, C., & Verno, A. (2011). *CSTA K–12 Computer Science Standards: Revised 2011*. Association for Computing Machinery. https://doi.org/10.1145/2593249

Seehorn, D., Primann, T., Lash, T., Twarek, B., Moix, D., Batista, L., Bell, J., Kuszmaul, C., O'Grady-Cunniff, ., Park, M., Pollock, L., Ray, M., Ryder, D., Sedgwick, V., Smith, G., & Uche, C. (2017). *CSTA K-12 Computer Science Standards, revised 2017*. Computer Science Teachers Association. https://csteachers.org/csta-k-12-computer-science-standards-revised-2017/

Scott, A., White, S. V., Levitt, D., & Bobb, K. (2023). *Justice-centered computing: A framework for action*. Kapor Foundation. https://www.kaporcenter.org/justicecs/

SHAPE America – Society of Health and Physical Educators. (2024a). *National health education standards*. https://www.shapeamerica.org/standards/health/new-he-standards.aspx

SHAPE America – Society of Health and Physical Educators. (2024b). *National physical education standards*. https://www.shapeamerica.org/standards/pe/new-pe-standards.aspx

Tedre, M., Denning, P. J., & Toivonen, T. (2021). CT 2.0. In *Proceedings of the 21st Koli Calling International Conference on Computing Education Research (Koli Calling '21)*, ACM. https://doi.org/10.1145/3488042.3488053

Tucker, A. (2003). *A model curriculum for K–12 computer science: Final report of the ACM K-12 task force curriculum committee*. Association for Computing Machinery. https://dl.acm.org/doi/10.1145/2593247

Tucker, A., Deek, F., Jones, J., McCowan, D., Stephenson, C., & Verno, A. (2006). *A model curriculum for K–12 computer science, 2nd edition*. Association for Computing Machinery. https://people.cs.vt.edu/~kafura/CS6604/Papers/K-12ModelCurr2ndEd.pdf

Twarek, B., Koshy, S., Hinton, L., Cruz Novohatski, L., & Glass, S. (2025). *The 2025 computer science teacher landscape: Insights into teacher preparedness for a world powered by computing*. https://landscape.csteachers.org

Vinney, C. (2023). *What is human-centered design? Everything you need to know*. UX Design Institute. https://www.uxdesigninstitute.com/blog/what-is-human-centered-design/

Acknowledgments

Advisors

State Computer Science Supervisors

- Alabama: Amanda Dykes, Computer Science Specialist, Alabama State Board of Education
- Alaska: Anthony White, Computer Science Content Specialist, Alaska Department of Education
- Arkansas: Adam Musto, Director of Computer Science Education, Arkansas Department of Education
- Arizona: Alecia Henderson, Computer Science and Educational Technology Specialist, Arizona Department of Education
- California: Katherine Goyette, Computer Science Coordinator, California Department of Education
- Georgia: Alba Gutierrez, Computer Science Program Specialist, Georgia Department of Education
- Hawaii: Brett Tanaka, Computer Science Educational Specialist, Hawaii Department of Education
- Idaho: Katie Bosch-Wilson, Computer Science and Emerging Technology Coordinator, Idaho Department of Education
- Illinois: Neha Thakkar, Computer Science Consultant, Illinois State Board of Education
- Indiana: Matthew Baron, Senior Computer Science Specialist, Indiana Department of Education
- Iowa: Michelle Meier, Computer Science Consultant, Iowa Department of Education
- Kansas: Dr. Stephen King, Computer Science Education Program Consultant, Kansas State Department of Education
- Kentucky: Dr. Sean Jackson, Computer Science Program Manager, Kentucky Department of Education
- Maine: Allison Braley, Computer Science Specialist, Maine Department of Education
- Maryland: Quiana Bannerman, Coordinator of School Support, Maryland Department of Education
- Massachusetts: Anne DeMallie, Director of STEM, Massachusetts Department of Elementary and Secondary Education
- Massachusetts: Paula Moore, Digital Literacy and CS Content Lead, Massachusetts Department of Elementary and Secondary Education
- Michigan: Cheryl Wilson, Computer Science Consultant, Michigan Department of Education
- Mississippi: Shelly Hollis, Director, Center for Cyber Education
- Missouri: Michael Corcoran, Assistant Director of Computer Science, Missouri Department of Elementary and Secondary Education

- Nebraska: Shaun Young, Computer Science and Technology Education Specialist, Nebraska Department of Education
- New Mexico: Dr. Melissa DeLaurentis, College and Career Readiness Manager, New Mexico Public Education Department
- North Carolina: Karen McPherson, Digital Technology and Computer Science Education Section Chief, North Carolina Department of Public Instruction
- Ohio: John Wiseman, Educational Program Specialist, Ohio Department of Education and Workforce
- Oklahoma: Jeremy Maker, Director of Computer Science Education, Oklahoma Department of Education
- Oregon: Andrew Cronk, Computer Science Education Specialist, Oregon Department of Education
- Pennsylvania: Dr. Sara Frey, Computer Science Consultant, Pennsylvania Training and Technical Assistance Network
- South Carolina: Sherri Smith, Computer Science Regional Coach, South Carolina Department of Education
- Utah: Kristina Yamada, Career and Technical Education (CTE) Specialist, Utah State Board of Education
- Virginia: Keisha Tennessee, Computer Science Coordinator, Virginia Department of Education
- Washington: Terron Ishihara, Computer Science Program Supervisor, Office of Superintendent of Public Instruction
- West Virginia: Cliff Sullivan, Computer Scien e and AI Coordinator, West Virginia Department of Education
- Wisconsin: Amy Bires, Computer Science and Digital Learning Innovation Education Consultant, Wisconsin Department of Public Instruction

District Leads

- Jared Amalong, Director of Computer Science Education and Distance Learning, Sacramento County Office of Education
- Christy Crawford, Senior Director of Computer Science and AI Education, New York City Deptartment of Education

Curriculum/Professional Development Providers

- Nancye Blair Black, CEO, The Block Uncarved
- Dr. Jamila Cocchiola, Curriculum Product Manager, CodeAI
- Rebecca Dovi, Head of Strategy, BootUp Professional Development
- Laura Gray, Senior Learning Manager, Raspberry Pi Foundation
- Bill Marsland, Director of Education and Training, Code Ninjas
- Dr. Cynthia Mitchell, Director of AP Computer Science Principles, The College Board

- Dr. Brandon Nicholson, CEO, The Hidden Genius Project
- Katie O'Shea, Director of Curriculum, CodeHS
- Hannah Walden, VP of Education, Hello World
- Andrea Wilson Vazquez, Senior Learning Manager, Raspberry Pi Foundation
- Ashley Woodard, Regional Outreach Coordinator, Girls Who Code

Non-profit & Corporate Partners

- Josh Caldwell, Learning Design Program Manager, Google
- Charlene Cooper, Director, CYBER.ORG
- Charlotte Dungan, Chief Learning Officer, Mark Cuban Foundation
- Dr. Jeff Forbes, Di ector, Strategic Initiatives & Partnerships, Computing Research Association
- Dr. Christina Gardner-McCune, Co-Founder, AI4K12
- Mahmoud Harding, Instructional Design Director, Data Science 4 Everyone
- Elissa Hozore, Director of Accessible Learning, Code in the Schools
- Dr. Jaci McCune, Director, ECEP Alliance
- Dr. Julie M. Smith, Senior Education Researcher, Institute for Advancing Computing Education
- Dr. Amy Stephens, Director, The National Academies of Sciences, Engineering, and Medicine
- Shana V. White, Director of CS Equity Initiatives, Kapor Foundation

Researchers

- Dr. Alexis Cobo, Postdoctoral Researcher, University of Florida
- Dr. Diana Franklin, Professor, University of Chicago
- Dr. Mark Guzdial, Professor, University of Michigan
- Dr. Tia Madkins, Assistant Professor, University of Texas at Austin
- Dr. Ryan Torbey, Evaluation Researcher, University of Texas at Austin

School of Education Faculty

- Dr. Michelle Friend, Associate Professor, University of Nebraska at Omaha
- Dr. Todd Lash, Director of Computer Science Teacher Education, University of Illinois
- Dr. Christine Liebe, Professor of Practice, Colorado School of Mines
- Jennifer Rosato, Director, Northern Lights Collaborative for Computing Education, University of Minnesota

Global Partners

- India: Spruha Satavlekar, Research Scholar, Indian Institute of Technology Bombay
- South Korea: Dr. Wonjin Yu, Clinical Assistant Professor, The University of Alabama
- Turkiye: Dr. Yasemin Gulbahar, Visiting Professor, Teachers College, Columbia University
- United Kingdom: Dr. Jane Waite, Senior Research Scientist, Raspberry Pi Foundation

Reviewers

Shawn Abele
Paula Abrantes
Kim Acosta
Kimberly Adams
Mike Afdahl
Dr. Derek Aguiar
Dr. Amna Ahmad
Dr. Israel Aikulola
Leah Aiwohi
Cassandra Williams Allen
Elexis Allen
Aimee Alsop
Gloria Amador
Marilou Anderson
Svea Anderson
Dr. Ajayi Ekuase-Anwansedo
Matt Arnold
Amna Askari
C. Atkins
Dr. Gerald W. Aungst
Skyler Austen
Amanda Austin
Dinesh Ayyappan
Souhaib Azanki
Derek Babb
Mayra Bachrach
Elizabeth Bacon
Vishal Sahebrao Badgujar
DeMario Bailey
Frederick Ballew
Matthew Baron
Dr. Joanne Barrett
Chris Bartlo
Brenda Bass
Dr. Satabdi Basu
Ryan Bean
Kris Beck
Dr. Amanda Bell
Ashley Benitez Smith
Kathy Benson
Lionel Bergeron
Paige Besthoff
Heather Bitter
Lisa J. Blank
Dr. Melanie Blanton
Emily Bleyle
Dan Blier
Dr. Jacqualyn Blizzard-Caron
Josh Blumberg
Nicole Bond
Amy Bove
Cameron Braun
Cynthia Brawner, NBCT
Dr. Lauren Bricker
Valerie Brock
Fran Bromley-Norwood
Claire Brown
Brittany Brown
Ryan Brown
Matthew Brown
Kymberli Bryant
Dr. April Burton
Gyuri Byun
Leslie Cadien
Bess Camara
Maria Luisa Cardenas
Candido Castro Jr III
Angela Chavez
Charlotte Cheng
Alberto Chiecchio
Sarah Ciras
Michele Cislo
Meghan Clark
Beverly Clarke
Cari Cline
Brooke Clugh
Tiffany Cobb
Sheila Marie Coles
Renee Coley
Ife' Colvin
Travis Conley
Randy Connolly
Michelle Coots
Dr. Michelle Crary
David Czechowski
Samantha Dahlby
Dr. Tikyna Dandridge
Caren Daniel
Della Dastur
Rashad Davis
Tonya Davis
Pamela Davis-Ghavami
Sofía De Jesús
Anita Faye Debarlaben
Mark DeLoura
Servane Demol
Crystal DeMoura
Amanda Dennard
Michael Deutsch
Sandy Dibble
Beth Dierker
Shreepriya Dogra
Shawn Draczynski Tobin
Charlotte Dungan
Benjamin Dusek
Lauren P. Dykstra
Jacqueline D. Edwards
Ron Eglash
Amy Eguchi
Ken Enloe
Justin Eom
Rudy Escobar
Dr. Ruth Farmery
S.D. Felder
Randa Felske
Rachel Fenichel
Patricia Ferris
Amy Fetherston
Casey Fiesler
Maria Fernanda Reis Costa Loureiro Filipe
Michael S. Fine
Ellen Fishter
Jessica Fletcher
Jacqui Fouché
Daisy Frederick
Bruce Fuda
Alexandra Fuentes
Manjula M. G.
Judianne M. Ganschow
Ana Garcia
Antero Garcia
Brooke George
Sasha George
Deanna George
Gi Soong Chee
Chris Gibbs
Rob Gibson
Pulkit Goel
Melissa Goodall
Dr. Joanna Goode
Bethany Goossen
Jen Gossert
Emily Grantham
Julia Green
Mr. Adam Grindstaff
Mary Cate Gustafson-Quiett
Bryn Hamamura
Dr. Karen Harper
Eva Harvell
Rory Hathale
Marie K. Heath
Dr. John Heffernan
Jennifer Heidlebaugh
Dr. Tiffany Henderso
Katie Henry
Kimberly Hermans
Jessica Hexsel
Dr. Lenette Hillian
Dr. Michelle Hoda Wilkerson
ShaKira S. Hodges
Jessica Holloway
Sallie Holloway
Janelle Horton
Diane Horvath
Sylvia Huber
Benjamin Hurley

Chidimma Ibeh
Kimberly Ingraham-Beck
Jason Innes
Jasmine Jackson
Lindie Good-Jackson
Johanna Jacob
Miles Jaffee
Hyeona Jang
Jessica Jarboe
Hungyu Je
IlYeong Jeong
Cristina Jimenez
Susan Jinks
Travis Jiskra
Shahrukh Jiwani
Amy Johnson
Joshua Johnson
Tod Johnston
Lee Ann Jones
Dan Jones
Sarah Judd
Kylie Jue
Bung-Woo Jun
De Paul Kannamthanam
Cortney Kargusang
Alexander Kash
Kim Kaufman
Dr. Erdogan Kaya
Paige Kelpine
Irum Khan
Dr. Nikhil Khandar
Elvira Khwatenge
Soohwan Kim
Hansung Kim
Cheun Kim
Hyunji Kim
Dr. Michael W. Kimwele
Yasemin Kinak
Maria Klawe
Joe Kmoch
Amy J. Ko
Barnabas Koech
Pam Koleszar
Ramkumar Komakula
Laura M. Kosky
Jenny Kostka
Kassiani Kotsidou
Dr. Akanksha Kulkarni
Jessica T. Kulp
Krystal LaFontaine
Michelle Lagos
Monika Lambert
Julie Lange
Ning Zhou Langworthy
Alva Laster
Karen Latimer
Shakara Lawrence
Tyresa Leader
Samantha Leav
Dr. Victor Lee
Jane L. Lehr
Laura Leis
Eugene Lemon
Diane Levitt
Christine Liebe
Jianxin Liu
Siming Liu
David Lockett
Kate Lockwood
Charles Logan
Jen Looper
Ralph Losanno
Alex Luciano
Randy Macdonald
Bill MacKenty
Sarah Mantz
Raquel T. Marasigan
Victor Martinez
Sara Mathew
Alex Mayszak
Joe Mazzone
Dr. Amanda McBride
Chris McCormick
Keith A. McCray Jr.
Mike McCue
Leandra McGriff
Dr. Kelly McNeil
Brian Mead
Gabriel Medina-Kim
Caroline Meeks
Kirsten Meltesen
Hazel Sila Mentes Tanaydin
Melissa Meyer
Jessa Miles
Will Miller
Greg Miller
Tara Miller
Timothy Milligan
Lisa Moe
Faisal B. Mohammed
Siti Sakinah Mohd Yusof
Maryann Molishus
Dr. William Mongan
Kristen Montesano
Luis Morales-Navarro
Sheba C. Morrison
Shelly A. Munoz
Lindsay Munoz
Erin Murphy
Guillermo A. Núñez-Salgado
Patrick O'Steen
Emmanuel Ojirhevwe
Kingsley Onyekachi Okpeh
Jose Olivares
Anna Otto
Nese Ozturk Tursin
Krupali D. Panchal
David Parker
Claire Parker
Heather Parr
Oksana Pasichnyk
Shilpa Paul
Rachel Pauley
Joseph Peregrin
Thereza Patricia Pereira Padilha
Mike Phelan
Dr. Sarah Phelps
Natalie Pinta
Rodolfo Pinto
Alejandra Plascencia
Emily Pool
Kelly Powers
Dr. Leanna Prater
Carolyn Preciado
Kiki Prottsman
Caitlin Provance
Rohya Prudhomme
Kelly Pryde
Dr. Ranjidha Rajan
Yulianti Ramahalim
Mr. Nagabala Ramakrishnasetty
Rampur Srinath
Zahra Razi
Heidi Reichart
Michael Renne
Rich Reynolds
Jonathan Rhodea
Leila Ribeiro
Monique M. Rice
Dr. Katie Rich
Stefanie Richardson
Neil Rickus
Dr. Norman "Storm" Robinson III
Jose Rodriguez
David Roggenbuck
Kevin Rose
Dana Runkle
Dr. S. Ananthi
Ryan Sackett
Ram Prasad Reddy Sadi
Fred Sagwe
Charlene Saint-Jean
Edmond Saint-Jean
Jean Salac
Hugo Alberto Salazar Sotelo
Maximilian Salceda
Anthony Sarnelle
Emmanuel Schanzer
Kelly Schuster-Paredes

Susan Schwarz
Tracey Sconyers
David M. Scrofani
Brian Sea
Jordan Seigler
Maria Sellers
Renee Serrano
Christian Servin
Dr. Dipa Shah
Syed Shahrooz Shamim
Jaidev Sharma
Meenakshi Sharma
Crystal Sheldon
Abby Sheridan
Heather Sherwood
Marti Shirley
Samantha Shultz-McGowan
Darshell Silva
Janelle Simpson-Luna
Abhishek Singh
Ava Sirrah
Lea Sloan
Polly Chen Smith
Kimberly Smith
Andrew Smith
Tom Smith
Emma Smith Zbarsky
Sara Smolevitz
Steve Snow
Heather Solis
Jenna Spain
Philisia Spearmon
Sonia Spindt
Dr. Brian M. Stamford
Liza Stark
Pete Steinfeld
Nancy Stevens
Christina Stickney
Dr. Amber Struthers
Steve Svetlik
Alberto Sweeney
Noh Tae Hyun
Nisha Talagala
Zuobin Tang
Lawrence Tanimoto
Dr. Hiroki Tanioka
Kurt Tholking
Alexandra Thomas
Dr. Emily Thomforde
Shelley Thompson
Tom Thompson
Eric Tietze
Abhishek Tiwari
Shawn Draczynski Tobin
Cheryl Tokarski
Kim W. Tracy
Dr. Cathy Tran
Amy Traylor
Rose Truglio
Andreas Tsouchlaris
Steven Turner
John Tusch
Bruce Tuttle
Natalie Ulloa
Christopher Onyilo Uloko
Angelo Urrico
Sepehr Vakil
Karla Veronica Velarde
Sabitha Vinod
Dr. Sara Vogel
William "Liam" J. Von Alt III
Angela Votion
Patrice Wade
Sonia Wadhwa
Bimlesh Wadhwa
Tory Wadlington
Roger Wagner
Moriah Walker
Kevin Walker
Liz Walsh
Annette Walter
Thomas Wang
Dani Ward
Jessica Harless Wash
JoAnna Watson
Joshua Waymire
Jennifer Weaver
Tracey Weller
Dr. Sheree Wheat
Cinamon White
Stephanie Wieczorek
Jeff ey J. Wile
Amber C. Williamson
Delmar Wilson
Vivian Wilson
Dr. Eben Witherspoon
Daryl Wong
Eric Wong
Sylvia Wood
Amy B. Woodman
Shelby Woods
Ajay Kumar Yadav
Dr. Wei Yan
Julie York
Dr. Wonjin Yu
Bin Yu
Dr. Stephanie Zeiger
Melissa Zeitz
Shenghua Zha
Helen Zhang

Research Support

- Hyein Jee, Graduate Student, Michigan State University
- Dr. Michael Lachney, Associate Professor, Michigan State University
- Dr. Monica McGill, Founder and CEO, Institute for Advancing Computing Education
- Dr. David Phelps, Senior Researcher, Telos Learning
- Dr. Jean Ryoo, Director of Research, Computer Science Equity Project, University of California, Los Angeles
- Dr. Rafi Santo, Principal esearcher, Telos Learning
- Brooke Starks, Graduate Student, Michigan State University
- Dr. Dave Touretzky, Research Professor, Carnegie Mellon University
- Xinyue Zhou, Graduate Student, Michigan State University

Additional Support

CSTA

- Dr. Amanda Bell, Project Manager
- Justine Chavez-Crespin, Professional Learning + Content Manager
- Kaylah Doss, Senior Event Coordinator
- Honna George, Chief Finance & Operating Officer
- Shaina Glass, Director of Learning & Community
- Stacy Jeziorowski, Senior Marketing and Communications Lead
- Kenz Mangan, Digital Marketing Manager
- Portia Morrell, Research & Innovation Program Manager
- Jordan Myers, Senior Event Coordinator
- Rachael Steacy, Owner, Rachael Steacy Events
- Dana Weingartner, Conference & Events Senior Manager
- Lauren Welke, Director of Strategic Development

AnLar

- Pearson Barlow, Senior Project Director
- Emily Bytheway, Quality Analyst
- Prabal Dey, Lead Developer
- Anthony Garofano, Creative Director
- Becki Stenger, Senior Designer
- Audrey Ward, Project Manager

WestEd

- Dr. Jennifer Houchins, Research Associate
- Dr. Rachel Lee, Research Associate
- Jennifer Mendenhall, Senior Communications Specialist
- Anita Moorjani, Research Associate
- Rosalind Owen, Research Associate
- Dr. Jennifer Tsan, Research Associate

Appendix A: Research to Inform the CSTA PK–12 Standards Revision

Executive Summary

The revision to the CSTA PK–12 Standards was a collaborative and research-driven process designed to ensure that the resulting standards reflect the l test insights and best practices in computer science education. This report documents the extensive research that informed the revision, resulting in updated standards that reflect the grounded in research" value that underpins the process.

WestEd led a detailed literature review and analysis of related standards and frameworks, as well as a comparison with 12 other sets of instructional guidelines. The research was guided by key questions about alignment with learning progressions, the influen e of emerging technologies such as generative AI, pedagogical practices promoting equity, and coverage across existing computer science curricula. WestEd developed design briefs and literature reviews for each standard concept, ensuring that writers had access to the most relevant and timely research as they drafted new standards. Additionally, the *Amplifying Social Impacts of Computing Standards* (ASICS) project provided recommendations related to impacts and ethics of computing, and the *Identifying AI Priorities for All K-12 Students* project, co-led by CSTA and AI4K12, developed recommendations related to artificial intelligen e.

This report outlines key recommendations pulled from this research as well as specific examples of how the research is reflected in the upd ted standards. Table 1 presents a preview of the findings, highlighting one ecommendation and one example for each topic in the report. The full report includes several examples per concept.

Table 1. Examples of Recommendations and Their Reflection in the Standards

Concept	Research Provided	Sample Recommendation	Example in the Standards
Structure	*Comparison of CSTA Standards to Other Standards and Frameworks* *Standards Alignment Review* *Structure Design Brief*	Grade bands should be organized in the following way: • PK standards should be included as part of a grade band (e.g., PK–2), if they are included at all. • Elementary school standards (grades K–5) can be written for individual grades or for two grade bands: K–2 and 3–5. • Middle school standards (grades 6–8) should be banded together. • High school standards (grades 9–12) should be banded together.	PK/K standards are banded, and grades 1–5 are separated into discrete grade level standards. Middle school standards (grades 6–8) are banded. High school standards (grades 9–12) are banded.

Concept	Research Provided	Sample Recommendation	Example in the Standards
Algorithms & Design	*Algorithms and Programming Literature Review* *Structure Design Brief*	Support students in following and writing step-by-step instructions for solving problems.	This recommendation is exemplified in the ollowing standards, as well as in many standards that ask students to design and create algorithms: • EK-ALG-PS-01: Carry out algorithms in daily activities. • E1-ALG-PS-01: Decompose a problem or task into individual parts to develop an algorithm.
Programming	*Algorithms and Programming Literature Review*	Program comprehension (i.e., reading, interpreting, and communicating about code) is an important concept that seems absent in the 2017 CSTA K–12 Standards.	The Programming subconcept Reading & Documenting addresses program comprehension.
Data & Analysis	*Data and Analysis Literature Review* *Data and Analysis Design Brief*	At the elementary level, students should ask and answer questions of small datasets collected in familiar contexts.	This recommendation is exemplified in the ollowing standards: • E2-DAT-DI-09: Develop a question that can be answered with data. • E4-DAT-DC-09: Organize collected data into a table using a computational tool, with rows representing records and columns representing attributes.
Systems & Security	*Computing Systems and Security Literature Review* *Computing Systems and Security Design Brief*	A correct and detailed understanding of the internet as a network of networks with specific servers and connections is rare and typically only seen in children over the age of 10 or 11, so this should not be addressed until middle school standards.	In the updated standards, MS-SYS-NT-35 (“Explain how the resilience of the internet depends on interconnected devices and their roles and functions within the network.”) is a middle school standard, rather than an elementary school standard.
Computing & Society	*Computing and Society Literature Review*	The History of Computing standards should progress from a small/local scope at lower grades to a larger/ global scope at higher grades.	The updated standards include a comprehensive progression demonstrating how computing relates to society as a whole. This recommendation is specifically exemplified in the ollowing standards progression: • Elementary (EK-SOC-HI-14): Identify computing technologies used in daily life that have changed over time. *(...continued)*

Concept	Research Provided	Sample Recommendation	Example in the Standards
Computing & Society *(continued)*	*Computing and Society Literature Review*	The History of Computing standards should progress from a small/local scope at lower grades to a larger/ global scope at higher grades.	• Middle school (MS-SOC-HI-38): Compare the roles of individuals, communities, organizations, and governments in shaping computing technologies across major eras in computing history. • High school (HS-SOC-HI-38): Analyze the historical trajectory of a specific omputing technology and how its development is linked to societal and environmental factors.
Artificial Intelligence	*Artificial Intelligence Design Brief* *Identifying AI Priorities for All K-12 Students*	Begin introducing AI curriculum in early elementary school.	AI is included in several elementary standards. For example: • E5-ALG-IM-03: Articulate how human-centered design principles are incorporated into the development of a computing technology. • E3-SOC-HU-18: Examine why people design and build computing technologies.
Impacts & Ethics	*Structure Design Brief* *Amplifying Social Impacts of Computing Standards Reference Materials* *Computing Impacts and Ethics in the CSTA Standards: A Synthesis of Expert Perspectives*	Impacts and ethics content is taught inconsistently when it is defined as a separ te concept within standards. Embed this content throughout the standards in order to increase the probability that these are taught more consistently.	Impacts and ethics content is embedded across multiple concepts instead of separated as its own concept.

The revised CSTA PK–12 Standards reflect a di ect line from research to practice. Recommendations from research, literature reviews, curriculum analyses, and expert partnerships were consistently translated into actionable standard language and structure throughout the update process. This meticulous integration ensures that standards not only respond to current instructional needs and technological advancements, but are also poised to support equitable, rigorous, and relevant computer science education for all students.

Introduction

Grounded in research was one of the core values guiding the revision to the CSTA PK–12 Standards. This report describes how research informed the revision process and foregrounds the ways the updated standards reflect esearch-based practices alongside our other values.

Research was one of many inputs that writers considered in drafting the updated standards. Writers also drew on public feedback, advisor feedback, community definitions, and their own computer science education experiences. So, although research informs many aspects of the updated standards, it is not always possible to draw a direct line from each research recommendation to a final standa d. We provide some examples of some direct connections from research to final standa d language in this report.

Given project constraints, the team prioritized research that primarily influen ed the Foundational Standards. *Reimagining CS Pathways* (CSTA et al., 2024) and related projects influen ed standards organization at a high level (including Foundational and Specialty organization) and provided a base layer of content for each of the defined specialty a eas. Due to the varied nature of the specialty standards, the team consulted with individuals and organizations that have conducted research related to each specific specialty a ea. This report focuses primarily on research related to the Foundational Standards.

Overview of the Research Process

WestEd conducted a literature review and detailed analysis of related standards and frameworks to inform the CSTA K–12 Standards update. The literature review was driven by the following questions:

- How well do 2017 CSTA and state standards align with research on learning progressions for K–12 computer science topics?
- How might the increasing use of generative AI influen e learning progressions for K–12 computer science topics, particularly around programming?
- What pedagogical practices have been shown to promote equity for students learning K–12 CS content?
- How well do popular K–12 CS curricula align with 2017 CSTA and state CS standards and practices?
- What learning progressions are suggested or inferred for K–12 computer science education related to:
  - algorithms?
  - programming?
  - data and analysis?
  - computing systems and security?
  - impacts and ethics?
- To what extent do popular K–12 computer science curricula cover 2017 CSTA standards?
- What pedagogical practices promote equity for students learning K–12 computer science content?

WestEd's initial literature review search resulted in 727 articles. The team narrowed the list down to 54 relevant articles matching an agreed-upon criteria for inclusion. The literature review of these 54 articles led to specific findings and ecommendations that writers were able to apply by concept area.

Additionally, WestEd conducted a standards and frameworks comparison to compare the 2017 CSTA standards to 12 other sets of standards, frameworks, and instructional guidelines.

- CS-specific standa ds and frameworks:
  - EU Informatics Reference Framework for School
  - AI4K12 Guidelines
  - K-12 Cybersecurity Learning Standards
- Other STEM standards and frameworks:
  - Common Core State Standards for Mathematics (CCSS-M)
  - Guidelines for Assessment and Instruction in Statistics Education II (GAISE II)
  - Next Generation Science Standards (NGSS)
  - Standards for Technological and Engineering Literacy (STEL)
- Arts and Humanities standards and frameworks:
  - Common Core State Standards for English Language Arts (CCSS-ELA)
  - National Curriculum Standards for Social Studies
  - National Core Arts Standards
  - The Common Career Technical Core
  - Social Justice Standards

The goals of the comparisons were to:

- Use the structure of other standards and frameworks to inform the structure of the revised computer science standards;
- Identify enduring content from the 2017 CSTA standards as well as areas where the field has evolved and the standards should be updated accordingly; and
- Make recommendations for standards writers about the structure and content of the revised computer science standards.

WestEd developed design briefs and literature reviews related to each standard concept based on this research. The WestEd team shared the resulting recommendations with the standards writers, who used the analyses to inform their work. Additionally, the ASICS project provided recommendations related to impacts and ethics of computing, and the *Identifying AI Priorities for All K–12 Students* project, co-led by CSTA and AI4K12, developed recommendations related to Artificial Intelligen e. Table 2 includes links to the research provided to writers during the standards development process.

Table 2. Research Shared with Writers

Concept or Topic	Research Provided
Structure	*Comparison of CSTA Standards to Other Standards and Frameworks* *Standards Alignment Review* *Structure Design Brief*
Algorithms & Design	*Algorithms and Programming Literature Review* *Structure Design Brief*
Programming	*Algorithms and Programming Literature Review*
Data & Analysis	*Data and Analysis Literature Review* *Data and Analysis Design Brief*
Systems & Security	*Computing Systems and Security Literature Review* *Computing Systems and Security Design Brief*
Computing & Society	*Computing and Society Literature Review*
Artificial Intelligence	*Artificial Intelligence Design Brief* *Identifying AI Priorities for All K–12 Students*
Impacts & Ethics	*Structure Design Brief* *Amplifying Social Impacts of Computing Standards Reference Materials* *Computing Impacts and Ethics in the CSTA Standards: A Synthesis of Expert Perspectives*
Other Resources	*Reimagining CS Pathways: Every Student Prepared for a World Powered by Computing* *State K–12 Computer Science Standards Comparison Report* *International Computer Science Standards Comparison Report* *The 2025 Computer Science Teacher Landscape: Insights Into Teacher Preparedness for a World Powered by Computing*

Examples of Research Recommendations and Their Influence on the Standards

The writers incorporated the research into their writing process as they made decisions about how to update the standards. This research was one of many inputs that writers considered in drafting the updated standards. They also drew on public feedback, advisor feedback, the *Reimagining CS Pathways report*, and their own computer science education experiences. Although it is not always possible to draw a direct line from each research recommendation to each standard, we can highlight specific examples f how the research is directly reflected in the upd ted CSTA PK–12 Standards. Included below are some select examples of recommendations from the research that were shared with standards writers as well as samples of the influen e of these recommendations on the final standa ds. These examples do not represent a comprehensive list of all the ways in which research informed the standards development. They instead serve as an exemplific tion of the direct connections between the research and the final standa ds. Examples are provided for all five standa ds concepts (Algorithms & Design, Programming, Data & Analysis, Systems & Security, and Computing & Society), the overall structure of the standards, and how AI and impacts & ethics content was incorporated throughout the standards.

Structure

Writers used the *Standards Alignment Review*, the *Comparison of CSTA Standards to Other Standards and Frameworks*, and the *Structure Design Brief* to inform decisions around the structure of the updated CSTA PK–12 Standards. Table 3 outlines some example recommendations from the research and the resulting structural decisions reflected in the final standa ds.

Table 3. Examples of Structure Recommendations and Resulting Standards Structure

Recommendation	Example(s) in the Standards
Adopt clear descriptions of the content and practices. Explicitly integrate practices and content within the standards.	The standards include clear descriptions of both concepts and practices. Practices are organized into four categories (of two to four practices each): Ethics & Social Responsibility, Inclusive Collaboration, Computational Thinking, and Human-Centered Design. Each standard integrates a subconcept with one or two specific practi es.
Standards should describe what students should know and be able to do after instruction. They should be accompanied by supplementary material that defines the oncepts, practices, and their respective learning progressions, as well as identifies inte disciplinary connections.	The standards themselves are limited to describing what students should know and be able to do after instruction. Each standard is accompanied by implementation examples and interdisciplinary connections. The introduction to the standards contains definitions f the concepts and practices.

Recommendation	Example(s) in the Standards
Grade bands should be organized in the following way: • PK standards should be included as part of a grade band (e.g., PK–2), if they are included at all. • Elementary school standards (grades K–5) can be written for individual grades or for two grade bands: K–2 and 3–5. • Middle school standards (grades 6–8) should be banded together. • High school standards (grades 9–12) should be banded together.	PK/K standards are banded, and grades 1–5 are separated into discrete grade level standards. Middle school standards (grades 6–8) are banded. High school standards (grades 9–12) are banded.
The length of standards should follow these guidelines: • Standards should generally be one sentence of around 20 words, with one verb. • Write 10–15 standards per grade level and aim for a total of 150–200 standards.	Each standard is one sentence of about 15 words and begins with a single, measurable verb from *Bloom's for Computing*.[1] There are about 15 foundational standards per grade level and about 200 foundational standards total.
Impacts and ethics are taught inconsistently when they are defined separ tely from other concepts within the standards. Embed this content throughout the standards in order to increase the probability that these are taught more consistently.	Impacts and ethics content is embedded across concepts instead of separated as its own concept.
In a review of eight widely used K–12 computer science curricula, the Algorithms & Programming topic area received the most coverage across all reviewed curricula except for one.	Algorithms and Programming are two distinct concepts rather than one combined concept.
In a review of eight widely used K–12 computer science curricula, standards for Computing Systems and for Networks & the Internet are covered less than standards for Algorithms and for Programming.	Computing Systems and Networks & the Internet are combined into one concept (Systems & Security) rather than two distinct concepts.

[1] Association for Computing Machinery (ACM) Committee for Computing Education in Community Colleges (CCECC). (2023). *Bloom's for computing: Enhancing Bloom's revised taxonomy with verbs for computing disciplines.* https://ccecc.acm.org/files/public_tions/Blooms-for-Computing-20240814.pdf

Algorithms & Design

Writers used the information presented in the *Algorithms and Programming Literature Review* and the *Structure Design Brief* to inform decisions about Algorithms & Design standards. This research informed the decision to separate Algorithms and Programming into two distinct concepts. Table 4 outlines some example recommendations from the research and how these recommendations are reflected in the final Algorithms & Design stand ds.

Table 4. Examples of Recommendations and Algorithms & Design Standards

Recommendation	Example(s) in the Standards
Due to increasing use of AI, there should be a greater emphasis on algorithms in curricula and teachers' instruction than the 2017 standards reflect. Splitting Algorithms and Programming into two distinct concepts allows for greater emphasis on algorithms and AI use.	Algorithms & Design and Programming are two distinct concepts. While there is a similar number of Programming standards as compared to the number of 2017 standards, there are far more Algorithms & Design standards. AI is integrated across all five oncepts and most significantly included in the Algorithms & Design and Computing & Society standards.
Support students in following and writing step-by-step instructions for solving problems.	This recommendation is exemplified in the ollowing standards, as well as in many standards that ask students to design and create algorithms: • EK-ALG-PS-01: Carry out algorithms in daily activities. • E1-ALG-PS-01: Decompose a problem or task into individual parts to develop an algorithm.
Support students in understanding how to make algorithms more efficient an scalable.	This recommendation is exemplified in the ollowing standards: • HS-ALG-PS-02: Optimize the design of an algorithm using procedural abstraction and control structures. • HS-ALG-PS-03: Evaluate algorithms for efficien , correctness, and clarity, using metrics or test cases.
Incorporate key ideas about human-centered design, "an approach to interactive systems that aims to make systems usable and useful by focusing on the users, their needs and requirements, and by applying human factors/ergonomics, and usability knowledge and techniques. This approach enhances effectiveness and efficien , improves human well-being, user satisfaction, accessibility, and sustainability and counteracts possible adverse effects of use on human health, safety, and performance" (NIST, 2021).	This recommendation is exemplified in the ollowing standards: • E3-ALG-IM-03: Compare how different algorithms may affect outcomes, situations, and people with a wide range of needs. • E4-ALG-IM-03: Evaluate how different algorithms for solving the same problem produce outcomes that may benefit or disadvantage di erent groups of people. • E5-ALG-IM-03: Articulate how human-centered design principles are incorporated into the development of a computing technology.

Programming

Research related to the Programming concept revealed that most of the 2017 Programming standards maintained alignment with current research. The resulting recommendations emphasized the importance of preserving this alignment in the updated standards. Writers used information presented in the *Algorithms and Programming Literature Review* to inform their decisions related to the Programming concept. Table 5 outlines some example recommendations from the research and how these recommendations are reflected in the final P ogramming standards.

Table 5. Examples of Recommendations and Programming Standards

Recommendation	Example(s) in the Standards
Due to increasing use of AI, there is a greater emphasis on algorithms in current curricula and teachers' instruction than the 2017 standards reflect. Splitting Algorithms and Programming into two distinct concepts would allow for greater emphasis on algorithms and AI use.	Algorithms & Design and Programming are two distinct concepts. AI is integrated across all five oncepts and most significantly included in the Algorithms & Design and Computing & Society standards.
The literature reviewed is in accord with many of the 2017 CSTA K–12 Standards. So, the existing standards provide a solid starting ground for the 2026 revision. Keep most of the Programming learning progressions intact, as they are the most aligned to evidence.	The Programming learning progressions from the literature and 2017 CSTA K–12 Standards are maintained.
Program comprehension (i.e., reading, interpreting, and communicating about code) is an important concept that seems absent in the 2017 CSTA K–12 Standards.	The Programming subconcept Reading & Documenting addresses program comprehension.
Instead of making these standalone standards, integrate these topics as practices within other standards: • Attribution and intellectual property • Project roles or project timelines	"Attribution and intellectual property" appears in the second practice in Ethics & Social Responsibility: *Respect other creators of computational technologies. Only use others' work with permission and give appropriate attribution.* "Project roles or project timelines" appears in the second practice of Inclusive Collaboration: *Establish shared goals, break the work into discrete tasks, and set development milestones.*

Data & Analysis

Writers used the *Data and Analysis Literature Review* and the *Data and Analysis Design Brief* to inform decisions related to the Data & Analysis standards. Table 6 outlines some example recommendations from the research and how these recommendations are reflected in the final D ta & Analysis standards.

Table 6. Examples of Recommendations and Data & Analysis Standards

Recommendation	Example(s) in the Standards
Consider using or adapting the four-step statistical problem-solving process from the GAISE II: (1) formulate statistical investigative questions, (2) collect/consider the data, (3) analyze the data, and (4) interpret the results. Data collection should progress in the following way in the standards: • *Elementary school:* Students can collect the data by hand, through counting, categorizing, and measuring objects, or use data that has already been collected. • *Middle school:* Students should use computational tools to collect and store data (e.g., through online surveys or sensors). Students should begin developing understanding of spreadsheets and how computational tools can assist with transforming, summarizing, and representing/visualizing data (especially using tables, bar charts, line graphs, and scatterplots). • *High school:* Students should use spreadsheets and other computational tools to transform, analyze, and visualize data and create simulations. Students should develop test cases to ensure that computations and simulations perform as intended. Advanced students may begin using statistical software that requires programming to clean data, conduct descriptive and inferential statistical analysis, and create/modify simulations.	The Data Investigation and Data Collection & Preparation subconcepts reflect this recommended progression of data collection. This is specifically e emplified in the ollowing Data Collection and Preparation standards progression: • *Elementary school* (E1-DAT-DC-08): Use multiple methods to collect both numeric and non-numeric data to help answer questions. • *Middle school* (MS-DAT-DC-23): Use a computational tool to sort, filte , group, and summarize structured data. • *High school* (HS-DAT-DC-21): Use a computational tool to generate simulated data that fits ertain parameters for use in a simulation. This is specifically e emplified in the ollowing Data Investigation standards progression: • *Elementary school* (EK-DAT-DI-09): Investigate a question that can be answered by collecting data in students' everyday environments. • *Middle school* (MS-DAT-DI-27): Summarize a data investigation process, including potential biases, limitations, and supporting evidence. • *High school* (HS-DAT-DI-25): Create a data visualization of a multivariate dataset to answer a question or make a classific tion or prediction.

Recommendation	Example(s) in the Standards
At the elementary level, students should ask and answer questions of small datasets collected in familiar contexts.	This recommendation is exemplified in the following standards: • E2-DAT-DI-09: Develop a question that can be answered with data. • E4-DAT-DC-09: Organize collected data into a table using a computational tool, with rows representing records and columns representing attributes.
At the elementary level, students should begin using simple computational tools to make basic data tables, picture graphs, bar graphs, and line graphs.	This recommendation is exemplified in the following standard: • E4-DAT-DC-09: Organize collected data into a table using a computational tool, with rows representing records and columns representing attributes.
At the middle school level, students should ask and answer questions of large datasets (hundreds or thousands of cases) that cannot be easily summarized manually and that are collected from a variety of contexts, including science, social science, and humanities.	This recommendation is exemplified in the following standard: • MS-DAT-DC-22: Explain how data and its associated metadata can be used to answer questions.
At the middle school level, students should use computational tools to collect and store data (e.g., online surveys, sensors). Students should begin developing an understanding of spreadsheets and how computational tools can assist with transforming, summarizing, and representing/visualizing data (especially by using tables, bar charts, line graphs, and scatterplots).	This recommendation is exemplified in the following standard: • MS-DAT-DC-23: Use a computational tool to sort, filte , group, and summarize structured data.
At the high school level, students should ask and answer complex questions involving large datasets collected from a variety of contexts, including science, social science, and humanities.	This recommendation is exemplified in the following standard: • HS-DAT-DI-26: Evaluate a data simulation or visualization to answer a data question, inform decision-making, and identify potential limitations.
At the high school level, students should use spreadsheets and other computational tools to transform, analyze, and visualize data and create simulations. Students should develop test cases to ensure computations and simulations perform as intended. Advanced students may begin using statistical software that requires programming to clean data, conduct descriptive and inferential statistical analysis, and create/modify simulations.	This recommendation is exemplified in the following standard: • HS-DAT-DI-25: Create a data visualization of a multivariate dataset to answer a question or make a classific tion or prediction.

Systems & Security

Writers used the information found in the *Computing Systems and Security Literature Review* and the *Computing Systems and Security Design Brief* to inform their work on the Systems & Security Standards. Table 7 outlines some example recommendations from the research and how these recommendations are reflected in the final ystems & Security standards.

Table 7. Examples of Recommendations and Systems & Security Standards

Recommendation	Example(s) in the Standards
There is a lot of content related to Networks and Security that was not addressed in the 2017 standards and should be included in the revised standards.	Writers expanded these standards greatly in the updated Systems & Security concept.
In general, security should be more comprehensively addressed in the standards. For example: • Students should know what a good password is and understand the importance of strong passwords. They should also understand what personal information is and how to protect it. • Students should be knowledgeable about ways to physically and digitally keep devices and information secure. Digital security knowledge should include methods of cryptography to securely transmit data. • Students should have knowledge of the roles of software and software developers in keeping devices and information secure. They should also be knowledgeable about different security measures (including encryption), understand the strengths and weaknesses of the measures, and understand potential security threats.	The Security subconcept is included within the Systems & Security concept in response to this recommendation. This recommendation is specifically e emplified in the following standards: • E1-SYS-SE-12: Describe how to keep devices and online accounts safe from unauthorized access. • E3-SYS-SE-13: Evaluate how sharing information online might reveal personally identifiabl information and other details. • HS-SYS-SE-31: Identify different types of cybersecurity and physical security measures and the trade-offs or users, data, and devices.
A correct and detailed understanding of the internet as a network of networks with specific servers and onnections is rare and typically only seen in children over the age of 10 or 11, so this should not be addressed until middle school standards.	MS-SYS-NT-35: Explain how the resilience of the internet depends on interconnected devices and their roles and functions within the network. This is a middle school standard in the updated standards, rather than an elementary school standard.
At the elementary level, students should know the components or parts of computing devices.	This recommendation is exemplified in the ollowing standard: • E5-SYS-HW-12: Explain how hardware and software components of a computing system work together to perform input and output operations, processing, and storage.

Recommendation	Example(s) in the Standards
At the elementary level, students should know that the internet is different from the devices used to access it and from the activities it facilitates. They should also be able to distinguish online activities from offline activitie	This recommendation is exemplified in the ollowing standard: • E3-SYS-NT-14: Explain how people access the internet to gain information and communicate with each other.
At the middle school level, students should understand that the internet is a network of networks with specific servers and connections.	This recommendation is exemplified in the ollowing standard: • MS-SYS-NT-35: Explain how the resilience of the internet depends on interconnected devices and their roles and functions within the network.
At the middle school level, students should (a) be able to form a comprehensive view of the computing systems they work with and (b) know how to navigate various representations of computing systems (e.g., the physical computing artifact, paper-based representations, code).	This recommendation is exemplified in the ollowing standard: • MS-SYS-NT-34: Model how information in a network is broken down into packets, transmitted between devices, and reassembled.
At the high school level, students should have knowledge about threats and vulnerabilities, including attacks that affect the security of sensitive data.	This recommendation is exemplified in the ollowing standard: • HS-SYS-SE-31: Identify different types of cybersecurity and physical security measures and the trade-offs or users, data, and devices.

Computing & Society

Writers used the recommendations and research found in the *Computing and Society Literature Review* to inform their work on the Computing & Society standards. The literature review specifically ocused on the history of computing because it is not commonly part of computer science standards, and one of the goals of the literature review was to learn how historical information was represented and incorporated into standards in other disciplines. Table 8 outlines some example recommendations from the research and how these recommendations are reflected in the final Computing & Society stand ds.

Table 8. Examples of Recommendations and Computing & Society Standards

Recommendation	Example(s) in the Standards
The History of Computing standards should progress from a small/local scope at lower grades to a larger/global scope at higher grades.	The updated standards include a comprehensive progression demonstrating how computing relates to society as a whole. This recommendation is specifically exemplified in the ollowing standards progression: • *Elementary school* (EK-SOC-HI-14): Identify computing technologies used in daily life that have changed over time. • *Middle school* (MS-SOC-HI-38): Compare the roles of individuals, communities, organizations, and governments in shaping computing technologies across major eras in computing history. • *High school* (HS-SOC-HI-38): Analyze the historical trajectory of a specific omputing technology and how its development is linked to societal and environmental factors.
CS standards should prioritize historical content over historical thinking, and history standards should focus on historical topics and not specific facts	The History of Computing subconcept is included within the Computing & Society concept, which highlights computer science–specific historical ontent without being overly prescriptive about which content to cover.
At the elementary level, focal topics for computing history standards should include comparison of daily life with computing today and in the past.	This recommendation is exemplified in the ollowing standards: • EK-SOC-HI-14: Identify computing technologies used in daily life that have changed over time. • E1-SOC-HI-14: Compare how an everyday activity changed after a specific omputing technology was introduced. • E3-SOC-HI-16: Examine how computing innovations have changed the ways people live, work, or communicate over time.
At the elementary level, focal topics for computing history standards should include computing and computer scientists in students' daily life and community.	This recommendation is exemplified in the ollowing standard: • E1-SOC-CE-16: Describe how computing is used by people at home, at school, and in the community.

Recommendation	Example(s) in the Standards
At the elementary level, focal topics for computing history standards should include notable computer scientists from a range of backgrounds, eras, industries, and areas of computing.	This recommendation is exemplified in the ollowing standard: • E4-SOC-HI-16: Investigate the contributions of diverse individuals and communities in the history of computing.
At the middle school level, focal topics for computing history standards should include notable computer scientists and organizations related to each generation of computing.	This recommendation is exemplified in the ollowing standard: • MS-SOC-HI-38: Compare the roles of individuals, communities, organizations, and governments in shaping computing technologies across major eras in computing history.
At the middle school level, focal topics for computing history standards should include notable impacts of computing on social, cultural, political, and economic spheres.	This recommendation is exemplified in the ollowing standard: • MS-SOC-HI-39: Analyze intended and unintended impacts of a historical computing technology on society and the environment.
At the high school level, focal topics for computing history standards should include beneficial and har ful consequences of past advances in computing on social, cultural, political, and economic spheres.	This recommendation is exemplified in the ollowing standard: • HS-SOC-HI-38: Analyze the historical trajectory of a specific omputing technology and how its development is linked to societal and environmental factors.
At the high school level, focal topics for computing history standards should include ethical and socially responsible computing for the present, based on lessons learned from computing's influen e on the past.	This recommendation is exemplified in the ollowing standard: • HS-SOC-ET-41: Evaluate the societal and environmental impacts of an emerging technology, including those that lead to inequities in access and outcomes.

Artificial Intelligence

Writers used the information in the *Artificial Intelligence Design Brief* and in *Identifying AI Priorities for All K–12 Students* to inform how they incorporated impacts and ethics content throughout the standards. Table 9 outlines some example recommendations from the research and how these recommendations impacted the way AI content is reflected in the final standa ds.

Table 9. Examples of Recommendations and Resulting Incorporation of Artificial Intelligence in the Standards

Recommendation	Example(s) in the Standards
Due to increasing use of AI, there should be a greater emphasis on algorithms in curricula and in teachers' instruction than the 2017 standards reflect. Splitting Algorithms and Programming into two distinct concepts would allow for greater emphasis on algorithms and AI use.	Algorithms & Design and Programming are two distinct concepts. Although there is a similar number of Programming standards as compared to the number of 2017 standards, there are far more Algorithms & Design standards. AI is integrated across all five oncepts and most significantly included in the Algorithms & Design and Computing & Society standards.

Minimally, include all prioritized learning outcomes within each of the following five c tegories identified in the *Identifying AI Priorities for All K–12 Students* report:
• Humans and AI
• Representation and Reasoning
• Machine Learning
• Ethical AI System Design and Programming
• Societal Impacts of AI
	All prioritized learning outcomes plus some additional learning outcomes from the AI Priorities project were incorporated into the standards.
	• Humans and AI learning outcomes are incorporated into
	  ▪ Computing & Society: Humans & Computing standards
	• Representation and Reasoning learning outcomes are incorporated into
	  ▪ Algorithms & Design: Machine Learning standards
	• Machine Learning learning outcomes are incorporated into
	  ▪ Algorithms & Design: Machine Learning standards
	  ▪ Computing & Society: Humans & Computing standards
	• Ethical AI System Design and Programming learning outcomes are incorporated into
	  ▪ Algorithms & Design: Machine Learning and Impacts of Algorithms & Design standards
	  ▪ Data & Analysis: Impacts of Data Science standards
	  ▪ Computing & Society: History of Computing standards
	• Societal Impacts of AI learning outcomes are incorporated into
	  ▪ Algorithms & Design: Impacts of Algorithms & Design standards
	  ▪ Systems & Security: Impacts of Computing Systems standards
	  ▪ Computing & Society: Emerging Technologies standards
	  ▪ Computing & Society: Career Exploration standards

Recommendation	Example(s) in the Standards
Begin introducing AI curriculum in early elementary school.	AI is included in several elementary standards. For example: • E5-ALG-IM-03: Articulate how human-centered design principles are incorporated into the development of a computing technology. • E3-SOC-HU-18: Examine why people design and build computing technologies.
In middle school, students should explain how people's goals and values can influen e the design of AI, how the design process can incorporate those goals and values, and how AI can show biases toward different cultures.	This recommendation is exemplified in the ollowing standards: • MS-DAT-IM-29: Analyze how decisions made at different stages of working with data can lead to biased data, misleading conclusions, and compromised AI models. • MS-SOC-ET-40: Evaluate when it is appropriate to use AI and other emerging technologies to solve a problem based on their capabilities, limitations, and environmental impacts.
In middle school, students should use generative AI to aid their programming of novel applications.	This recommendation is exemplified in the ollowing standard: • MS-PRO-RD-18: Analyze AI-generated code for accuracy and usability in a programming project.
In high school, students should evaluate the design of AI systems to ensure that those systems include a plan for accountability and a respect for privacy. Students should use an ethical design process to design AI, explain ethical dilemmas that might arise due to AI, and predict how AI can change society.	This recommendation is exemplified in the ollowing standard: • HS-DAT-IM-27: Evaluate the societal, environmental, and ethical implications of large-scale data collection and processing, including within AI applications.
Prepare students to be critical consumers, responsible creators, and informed citizens.	This recommendation is exemplified in the ollowing standards: • E5-ALG-ML-02: Train a machine learning model to make a classific tion or prediction. • HS-ALG-PS-05: Evaluate AI-generated output to assess bias, accuracy, and potential harms. • HS-ALG-ML-08: Develop a machine learning model for a chosen task using appropriate data and tools.

Impacts & Ethics

Writers used the information in the *Structure Design Brief*, *Amplifying Social Impacts of Computing Standards Reference Materials*, and *Computing Impacts and Ethics in the CSTA Standards: A Synthesis of Expert Perspectives* to inform how they incorporated impacts and ethics content throughout the standards. Table 10 outlines some example recommendations from the research and how these recommendations impacted the way impacts and ethics content is reflected in the final stand ds.

Table 10. Examples of Recommendations and Resulting Incorporation of Impacts & Ethics in the Standards

Recommendation	Example(s) in the Standards
Impacts and ethics are taught inconsistently when they are defined separ tely from other concepts within the standards. Embed this content throughout the standards in order to increase the probability that these are taught more consistently.	Impacts and ethics content is embedded across concepts instead of separated as its own concept.
Impacts and ethics should also be embedded within practices to encourage students to use computing for positive social impact and to respect others' rights and dignity when creating computational technologies.	Impacts and ethics content is contained within these Ethics & Social Responsibility practices: • 1. Use computing for positive social impact. • 2. Respect others' rights and dignity when creating computing technologies. Impacts and ethics content is also contained within these Human-Centered Design practices: • 10. Understand and involve diverse users in design decisions. • 12. Design computing technologies that empower and inform users.
Elevate responsible design practices.	This recommendation is exemplified in the ollowing standards: • E5-ALG-IM-03: Articulate how human-centered design principles are incorporated into the development of a computing technology. • MS-ALG-IM-10: Examine evidence of beneficial and har ful impacts, ethical issues, and biases of algorithms encountered in daily life. • HS-PRO-TR-20: Refine a omputing technology based on user feedback, testing results, and responsible design values to improve its effectiveness and impact.

Recommendation	Example(s) in the Standards
Identify values and specifically focus on harms and benefits f computational decisions.	This recommendation is exemplified in the ollowing standards: • E4-DAT-IM-11: Investigate how data collected about people may affect individuals and groups. • HS-ALG-IM-11: Articulate the values embedded in the design of an algorithmic system. • HS-SOC-ET-42: Design a conceptual solution to a real-world problem using an emerging technology, analyzing its potential benefits and harms
Avoid harm caused by computing technologies, including physical or mental injury, unjustified destruction or disclosure of information, and unjustified damage to p operty, reputation, or the environment.	This recommendation is exemplified in the ollowing standards: • E2-SYS-IM-13: Describe the benefits and harms th t arise from an individual's use of computing systems and digital tools. • MS-ALG-IM-10: Examine evidence of beneficial and har ful impacts, ethical issues, and biases of algorithms encountered in daily life.
Respect the work required to produce new ideas, inventions, creative works, and computing artifacts.	This recommendation is exemplified in the ollowing Ethics & Social Responsibility practice: • 2. Respect others' rights and dignity when creating computing technologies. This recommendation is exemplified in the ollowing standard: • HS-PRO-PD-14: Apply appropriate attribution of intellectual property when developing a computing technology.

Conclusion

As this report exemplifies, the upd ted CSTA PK–12 Standards are grounded in extensive research, ensuring both their relevance and their rigor for contemporary computer science education. The revision process thoughtfully integrated a wide range of research, literature reviews, analyses of standards and frameworks, and expert recommendations, with findings and design briefs directly informing the content and structure of the standards. The examples provided show clear connections between research and the resulting language and topics of the updated standards. This direct alignment reinforces the credibility and adaptability of the standards, supporting equitable, future-oriented, and research-based practices for all learners and ensuring that the standards reflect the evolving needs and insights f the computer science education community.

Appendix B:
PK–12 Foundational Standards by Level

PK – Kindergarten (Ages 4–6)

Concept	Identifier	Standard
Algorithms & Design	EK-ALG-PS-01	Carry out algorithms in daily activities.
	EK-ALG-ML-02	Recognize attributes of objects to notice patterns and make decisions.
	EK-ALG-IM-03	Describe how people create algorithms to solve problems.
Programming	EK-PRO-PD-04	Create a sequence of commands to solve a problem or express an idea.
	EK-PRO-VD-05	Identify everyday gestures and symbols that represent information people use to make choices.
	EK-PRO-RD-06	Describe how a sequence of commands completes a task.
	EK-PRO-TR-07	Identify a step in a sequence of commands that does not work as expected.
Data & Analysis	EK-DAT-DC-08	Use collected data to help answer questions.
	EK-DAT-DI-09	Investigate a question that can be answered by collecting data in students' everyday environments.
	EK-DAT-IM-10	Investigate how data can help a person make informed decisions in everyday life.
Systems & Security	EK-SYS-HW-11	Examine the use of tools to accomplish a task or solve a problem for different users.
	EK-SYS-SE-12	Differentiate between public and private information.
	EK-SYS-IM-13	Identify responsible behavior when using computing systems and digital tools.
Computing & Society	EK-SOC-HI-14	Identify computing technologies used in daily life that have changed over time.
	EK-SOC-HU-15	Explain that people design and develop computing technologies.
	EK-SOC-CE-16	Identify how people use digital devices at home, at school, and at work.

Grade 1 (Ages 6–7)

Concept	Identifier	Standard
Algorithms & Design	E1-ALG-PS-01	Decompose a problem or task into individual parts to develop an algorithm.
	E1-ALG-ML-02	Recognize that AI systems are technologies that use patterns in data to make decisions or generate new things.
	E1-ALG-IM-03	Explain how a change to an algorithm leads to a different outcome.
Programming	E1-PRO-PD-04	Create code from an algorithm that includes sequence to solve a problem or express an idea.
	E1-PRO-VD-05	Identify terms that refer to data values that change over time in everyday life.
	E1-PRO-RD-06	Explain the function of code that includes an event and sequence.
	E1-PRO-TR-07	Debug a program that includes a sequence of commands.
Data & Analysis	E1-DAT-DC-08	Use multiple methods to collect both numeric and non-numeric data to help answer questions.
	E1-DAT-DI-09	Compare a question that can be answered through a data investigation and a question that can be answered through other means.
	E1-DAT-IM-10	Examine a variety of data questions that address the needs of a person or community.
Systems & Security	E1-SYS-HW-11	Describe the purpose of basic hardware components of a computing system, using accurate terminology.
	E1-SYS-SE-12	Describe how to keep devices and online accounts safe from unauthorized access.
	E1-SYS-IM-13	Describe an individual's role in responsibly using computing systems and digital tools.
Computing & Society	E1-SOC-HI-14	Compare how an everyday activity changed after a specific computing technology was introduced.
	E1-SOC-HU-15	Differentiate between activities that humans do well and activities that computing technologies do well.
	E1-SOC-CE-16	Describe how computing is used by people at home, at school, and in the community.

Grade 2 (Ages 7–8)

Concept	Identifier	Standard
Algorithms & Design	E2-ALG-PS-01	Create an algorithm that includes sequence, events, and iteration to solve a problem or express an idea.
	E2-ALG-ML-02	Examine how data is used to train a machine learning model.
	E2-ALG-IM-03	Describe how an algorithm might impact peers in varied situations.
Programming	E2-PRO-PD-04	Create code from an algorithm that includes sequence, events, and iteration to solve a problem or express an idea.
	E2-PRO-VD-05	Label different representations of information with a name and whether its value is constant or changes.
	E2-PRO-RD-06	Explain the steps taken to create a program.
	E2-PRO-TR-07	Debug a program that includes sequence, events, and iteration.
Data & Analysis	E2-DAT-DC-08	Compare numeric and non-numeric types of data in terms of how they are collected and what information they provide.
	E2-DAT-DI-09	Develop a question that can be answered with data.
	E2-DAT-IM-10	Distinguish between data collection approaches, including those that may lead to inaccurate or biased data.
Systems & Security	E2-SYS-HW-11	Explain how the basic hardware components of a computing system work together to perform input and output (I/O) operations.
	E2-SYS-SE-12	Explain how online actions have real-world consequences.
	E2-SYS-IM-13	Describe the benefits and harms th t arise from an individual's use of computing systems and digital tools.
Computing & Society	E2-SOC-HI-14	Analyze the ways that people from different cultures, backgrounds, and time periods have designed computing technologies to help them solve problems and express ideas.
	E2-SOC-HU-15	Investigate situations where humans have created computing technologies to solve problems.
	E2-SOC-CE-16	Investigate how personal interests connect to computing in different industries and careers.

Grade 3 (Ages 8–9)

Concept	Identifier	Standard
Algorithms & Design	E3-ALG-PS-01	Create an algorithm that includes a combination of sequence, events, iteration, and selection to solve a problem or express an idea.
	E3-ALG-ML-02	Investigate how a machine learning model can change when new data is added to a training set.
	E3-ALG-IM-03	Compare how different algorithms may affect outcomes, situations, and people with a wide range of needs.
Programming	E3-PRO-PD-04	Develop code from a student-created algorithm that includes a combination of sequence, events, iteration, and selection to solve a problem or express an idea.
	E3-PRO-PD-05	Use constructive feedback to improve a program.
	E3-PRO-VD-06	Identify the variables being stored and manipulated in a program.
	E3-PRO-RD-07	Articulate how a specific segment f code contributes to the overall purpose of a program.
	E3-PRO-TR-08	Debug a program that includes a combination of sequence, events, iteration, and selection.
Data & Analysis	E3-DAT-DC-09	Evaluate numeric and non-numeric data for accuracy and completeness.
	E3-DAT-DI-10	Investigate a data question involving relationships between multiple attributes.
	E3-DAT-IM-11	Design a data collection process that addresses the needs of people from different backgrounds or groups.
Systems & Security	E3-SYS-HW-12	Describe the role of software in a computing system to accomplish tasks or solve problems.
	E3-SYS-SE-13	Evaluate how sharing information online might reveal personally identifiable i formation and other details.
	E3-SYS-NT-14	Explain how people access the internet to gain information and communicate with each other.
	E3-SYS-IM-15	Describe how widely used computing systems may impact an individual's life and community.
Computing & Society	E3-SOC-HI-16	Examine how computing innovations have changed the ways people live, work, or communicate over time.
	E3-SOC-ET-17	Describe how new technologies create both benefits and risks in personal and family life.
	E3-SOC-HU-18	Examine why people design and build computing technologies.
	E3-SOC-CE-19	Explain how people in different industries use computing technologies and skills to accomplish their work.

Grade 4 (Ages 9–10)

Concept	Identifier	Standard
Algorithms & Design	E4-ALG-PS-01	Create a written representation of an algorithm that includes a combination of sequence, events, iteration, and selection to solve a problem or express an idea.
	E4-ALG-ML-02	Analyze relationships between the properties of training data and a machine learning model's output.
	E4-ALG-IM-03	Evaluate how different algorithms for solving the same problem produce outcomes that may benefit or disadvantage di erent groups of people.
Programming	E4-PRO-PD-04	Compare different programming solutions to the same problem based on correctness and clarity.
	E4-PRO-PD-05	Collaborate with a team by offering a meaningful contribution to creating a program.
	E4-PRO-VD-06	Trace how data flows and changes variable values in a p ogram.
	E4-PRO-RD-07	Document a program to clarify its functionality.
	E4-PRO-TR-08	Debug a program incrementally and repeatedly throughout the development process.
Data & Analysis	E4-DAT-DC-09	Organize collected data into a table using a computational tool, with rows representing records and columns representing attributes.
	E4-DAT-DI-10	Create an explanation that includes at least one data visualization to report the process and results of a data investigation.
	E4-DAT-IM-11	Investigate how data collected about people may affect individuals and groups.
Systems & Security	E4-SYS-HW-12	Apply a basic troubleshooting process to identify and fix ommon hardware and software issues.
	E4-SYS-SE-13	Distinguish between authentication and authorization in protecting devices and private information.
	E4-SYS-NT-14	Compare wired and wireless methods that computing devices use to connect to the internet.
	E4-SYS-IM-15	Investigate the impacts of widely used computing systems on natural resources and the environment.
Computing & Society	E4-SOC-HI-16	Investigate the contributions of diverse individuals and communities in the history of computing.
	E4-SOC-ET-17	Analyze how the limitations of existing technologies can lead to emerging technologies.
	E4-SOC-HU-18	Distinguish between human learning and machine learning processes.
	E4-SOC-CE-19	Investigate how the workforce adopts new computing technologies and continues to update their computing skills.

Grade 5 (Ages 10–11)

Concept	Identifier	Standard
Algorithms & Design	E5-ALG-PS-01	Create a visual representation of an algorithm that includes variables and a combination of sequence, events, iteration, and selection to solve a problem or express an idea.
	E5-ALG-ML-02	Train a machine learning model to make a classific tion or prediction.
	E5-ALG-IM-03	Articulate how human-centered design principles are incorporated into the development of a computing technology.
Programming	E5-PRO-PD-04	Create a novel program by modifying or combining elements of existing programs.
	E5-PRO-PD-05	Construct individual components of a program that are collaboratively assembled into a programming project.
	E5-PRO-VD-06	Use variables to store, compare, and modify data within a program.
	E5-PRO-RD-07	Create embedded or external documentation for a programming project.
	E5-PRO-TR-08	Debug a program using systematic strategies.
Data & Analysis	E5-DAT-DC-09	Use computational tools to collect and organize different types of data.
	E5-DAT-DI-10	Analyze a dataset to identify the nature and possible sources of variability in the data.
	E5-DAT-IM-11	Analyze the benefits and risks f a computing technology that uses collected data.
Systems & Security	E5-SYS-HW-12	Explain how hardware and software components of a computing system work together to perform input and output operations, processing, and storage.
	E5-SYS-SE-13	Describe the concepts of the CIA triad and how each component is important in protecting information.
	E5-SYS-NT-14	Distinguish between the components of wired and wireless networks.
	E5-SYS-IM-15	Examine how computing systems impact culture and the ways people live and work.
Computing & Society	E5-SOC-HI-16	Analyze how the inclusion or exclusion of diverse individuals and communities has shaped the design, development, and societal impact of computing technologies.
	E5-SOC-ET-17	Examine how people decide whether or not to use emerging technologies.
	E5-SOC-HU-18	Evaluate when it is appropriate to use or not use computing technologies to solve a problem.
	E5-SOC-CE-19	Examine how professionals collaborate while using computing technologies to solve problems.

Middle School (Grades 6–8, Ages 11–14)

Concept	Identifier	Standard
Algorithms & Design	MS-ALG-PS-01	Design an algorithm that includes variables of multiple data types to solve a problem or express an idea.
	MS-ALG-PS-02	Model a given algorithm with a flowchart or pseudo ode that includes a combination of control structures and procedures.
	MS-ALG-PS-03	Verify the accuracy of an algorithm for given inputs.
	MS-ALG-PS-04	Justify whether a problem is best solved using procedural instructions, rule-based logic, data-driven methods, or a combination of these approaches.
	MS-ALG-PS-05	Use an AI tool to generate outputs that assist in solving a computational problem.
	MS-ALG-ML-06	Hypothesize how a machine learning model generates classific tions or predictions.
	MS-ALG-ML-07	Investigate ways to improve the accuracy of a machine learning model and reduce bias by refining the quality f examples and nonexamples in the training data.
	MS-ALG-ML-08	Evaluate the features and limitations of a machine learning model.
	MS-ALG-IM-09	Evaluate which human-centered design principles are present or missing in an existing computing technology.
	MS-ALG-IM-10	Examine evidence of beneficial and har ful impacts, ethical issues, and biases of algorithms encountered in daily life.
	MS-ALG-IM-11	Modify an algorithm to address a specific societal impact, ethical issue, or bias
Programming	MS-PRO-PD-12	Use a procedure to structure code for clarity and reusability.
	MS-PRO-PD-13	Use reference documentation in program development.
	MS-PRO-PD-14	Justify the importance of attribution and intellectual property when developing computing technologies.
	MS-PRO-PD-15	Develop a program utilizing inclusive collaboration practices.
	MS-PRO-VD-16	Use variables of multiple data types to store, access, and manipulate data within a program.
	MS-PRO-RD-17	Analyze the roles of iteration, selection, variables, and procedures in a segment of code.
	MS-PRO-RD-18	Analyze AI-generated code for accuracy and usability in a programming project.
	MS-PRO-TR-19	Use systematic strategies to test, refine, and document changes to a computing technology to meet the intended purpose.
	MS-PRO-TR-20	Refine a omputing technology based on user feedback to improve its usability and accessibility.
Data & Analysis	MS-DAT-DC-21	Evaluate how different levels of precision and granularity in data collection affect accuracy, storage, and analysis.
	MS-DAT-DC-22	Explain how data and its associated metadata can be used to answer questions.
	MS-DAT-DC-23	Use a computational tool to sort, filte , group, and summarize structured data.
	MS-DAT-DC-24	Analyze options to address a data quality issue.

Concept	Identifier	Standard
Data & Analysis *(continued)*	MS-DAT-DI-25	Use a computational tool to identify relationships among variables in a dataset and make a classific tion or prediction.
	MS-DAT-DI-26	Create data visualizations to show how different design choices can impact the interpretation of the same data.
	MS-DAT-DI-27	Summarize a data investigation process, including potential biases, limitations, and supporting evidence.
	MS-DAT-IM-28	Explain the benefits and risks f allowing personal data and metadata to be collected and used in datasets, including issues of data ownership, privacy, and sovereignty.
	MS-DAT-IM-29	Analyze how decisions made at different stages of working with data can lead to biased data, misleading conclusions, and compromised AI models.
Systems & Security	MS-SYS-HW-30	Examine differences between computing systems based on user needs, system requirements, and potential societal, environmental, and ethical impacts.
	MS-SYS-HW-31	Describe computing devices used in various industries, their basic functions, and how they are used to accomplish tasks or solve problems.
	MS-SYS-SE-32	Explain the effects of not using the CIA triad when working with data.
	MS-SYS-SE-33	Evaluate common types of cyber attacks and preventions.
	MS-SYS-NT-34	Model how information in a network is broken down into packets, transmitted between devices, and reassembled.
	MS-SYS-NT-35	Explain how the resilience of the internet depends on interconnected devices and their roles and functions within the network.
	MS-SYS-IM-36	Collaborate to improve the design of a computing system to meet the needs of diverse users.
	MS-SYS-IM-37	Examine how access to computing systems can vary based on personal and social factors, such as physical ability, geographic location, socioeconomic status, and age.
Computing & Society	MS-SOC-HI-38	Compare the roles of individuals, communities, organizations, and governments in shaping computing technologies across major eras in computing history.
	MS-SOC-HI-39	Analyze intended and unintended impacts of a historical computing technology on society and the environment.
	MS-SOC-ET-40	Evaluate when it is appropriate to use AI and other emerging technologies to solve a problem based on their capabilities, limitations, and environmental impacts.
	MS-SOC-ET-41	Evaluate how the decisions made while designing an emerging technology influen e user experiences differently across different communities.
	MS-SOC-ET-42	Debate ways an emerging technology impacts the social, cultural, and environmental issues in local communities.
	MS-SOC-HU-43	Analyze how the decisions humans make when using computing technologies have ethical and social consequences.
	MS-SOC-CE-44	Analyze how workers in different careers use computational thinking to solve real-world problems.
	MS-SOC-CE-45	Evaluate how automation in technology can create or replace jobs and change how people work.

High School Foundational (Grades 9–12, Ages 14–18)

Concept	Identifier	Standard
Algorithms & Design	HS-ALG-PS-01	Design an algorithm using appropriate data structures to solve a problem or express an idea.
	HS-ALG-PS-02	Optimize the design of an algorithm using procedural abstraction and control structures.
	HS-ALG-PS-03	Evaluate algorithms for efficien , correctness, and clarity, using metrics or test cases.
	HS-ALG-PS-04	Describe the differences between deterministic and probabilistic algorithms.
	HS-ALG-PS-05	Evaluate AI-generated output to assess bias, accuracy, and potential harms.
	HS-ALG-ML-06	Justify the selection of a type of AI algorithm to accomplish a task.
	HS-ALG-ML-07	Evaluate training data by examining its source, quality, representativeness, potential biases, and privacy implications.
	HS-ALG-ML-08	Develop a machine learning model for a chosen task using appropriate data and tools.
	HS-ALG-IM-09	Design a computing technology using human-centered design principles.
	HS-ALG-IM-10	Evaluate the ethical implications, societal impacts, and potential biases of rule-based and data-driven algorithms.
	HS-ALG-IM-11	Articulate the values embedded in the design of an algorithmic system.
Programming	HS-PRO-PD-12	Create a modular program that uses procedures, external libraries, or objects to improve reusability and readability.
	HS-PRO-PD-13	Use documentation, libraries, application programming interfaces (APIs), and other tools in program development.
	HS-PRO-PD-14	Apply appropriate attribution of intellectual property when developing a computing technology.
	HS-PRO-PD-15	Collaborate on a programming project using a defined workflow t t includes design documentation and clear task roles.
	HS-PRO-VD-16	Create a program that uses appropriate data structures to store, access, and manipulate data.
	HS-PRO-RD-17	Analyze how a segment of code works, including the role of parameters, return values, and data structures.
	HS-PRO-RD-18	Evaluate AI-generated code for accuracy, reliability, and alignment with program requirements.
	HS-PRO-TR-19	Evaluate a computing technology's alignment with design specific tions and responsible design values, including its correctness, effectiveness, and user experience.
	HS-PRO-TR-20	Refine a omputing technology based on user feedback, testing results, and responsible design values to improve its effectiveness and impact.
Data & Analysis	HS-DAT-DC-21	Use a computational tool to generate simulated data that fits ertain parameters for use in a simulation.
	HS-DAT-DC-22	Create a data dictionary that describes the name, type, and allowable values for each attribute and the logical relationships between variables in a dataset.
	HS-DAT-DC-23	Use a computational tool to clean and organize text-based data.

Concept	Identifier	Standard
Data & Analysis *(continued)*	HS-DAT-DC-24	Evaluate different approaches to verifying consistency and compliance with expected data types, values, and ranges.
	HS-DAT-DI-25	Create a data visualization of a multivariate dataset to answer a question or make a classific tion or prediction.
	HS-DAT-DI-26	Evaluate a data simulation or visualization to answer a data question, inform decision-making, and identify potential limitations.
	HS-DAT-IM-27	Evaluate the societal, environmental, and ethical implications of large-scale data collection and processing, including within AI applications.
	HS-DAT-IM-28	Debate the efficacy f a policy or regulation to ensure responsible data use.
Systems & Security	HS-SYS-HW-29	Differentiate an operating system as a special type of software that manages both the hardware and other software components of a computing system, including handling memory and peripherals.
	HS-SYS-HW-30	Demonstrate the capabilities and limitations of a physical or simulated computing device to address a task or problem.
	HS-SYS-SE-31	Identify different types of cybersecurity and physical security measures and the trade-offs or users, data, and devices.
	HS-SYS-SE-32	Classify the causes and impacts of security breaches and social engineering attacks for individuals, industries, communities, and governments.
	HS-SYS-SE-33	Formulate a solution to a security flaw in a given system
	HS-SYS-NT-34	Diagram a network of computing systems, including hardware and software.
	HS-SYS-NT-35	Analyze how the internet functions as a network of networks and how it differs from other types of networks.
	HS-SYS-IM-36	Evaluate the rationale behind a law or policy governing the design and use of computing systems.
	HS-SYS-IM-37	Investigate how computing systems and infrastructure impact society and the environment, identifying who is affected and why.
Computing & Society	HS-SOC-HI-38	Analyze the historical trajectory of a specific computing technology and how its development is linked to societal and environmental factors.
	HS-SOC-HI-39	Propose modific tions to an existing policy or piece of legislation that encourages ethical innovation and minimizes societal risks associated with technology.
	HS-SOC-ET-40	Evaluate the fundamental technological differences between an emerging technology and established technologies and how those differences influen e computing.
	HS-SOC-ET-41	Evaluate the societal and environmental impacts of an emerging technology, including those that lead to inequities in access and outcomes.
	HS-SOC-ET-42	Design a conceptual solution to a real-world problem using an emerging technology, analyzing its potential benefits and harms
	HS-SOC-HU-43	Evaluate how human choices in using, designing, deploying, and regulating computing technologies have risks, benefits, and long-term impacts
	HS-SOC-HU-44	Debate perspectives on differences between human and artificial intelligence and their implications for consciousness, ethics, and human responsibility.
	HS-SOC-CE-45	Analyze how diverse teams of people use computational thinking and computing technologies to solve problems and express ideas.
	HS-SOC-CE-46	Connect computing knowledge and skills acquired to students' personal goals and career aspirations.

Appendix C:
PK–12 Foundational Standards Progressions Charts

		PK/K	Grade 1	Grade 2	Grade 3	Grade 4	Grade 5	Middle School	High School
Algorithms & Design	Algorithmic Problem Solving	EK-ALG-PS-01: Carry out algorithms in daily activities.	E1-ALG-PS-01: Decompose a problem or task into individual parts to develop an algorithm.	E2-ALG-PS-01: Create an algorithm that includes sequence, events, and iteration to solve a problem or express an idea.	E3-ALG-PS-01: Create an algorithm that includes a combination of sequence, events, iteration, and selection to solve a problem or express an idea.	E4-ALG-PS-01: Create a written representation of an algorithm that includes a combination of sequence, events, iteration, and selection to solve a problem or express an idea.	E5-ALG-PS-01: Create a visual representation of an algorithm that includes variables and a combination of sequence, events, iteration, and selection to solve a problem or express an idea.	MS-ALG-PS-01: Design an algorithm that includes variables of multiple data types to solve a problem or express an idea. MS-ALG-PS-02: Model a given algorithm with a flowchart or pseudocode that includes a combination of control structures and procedures. MS-ALG-PS-03: Verify the accuracy of an algorithm for given inputs. MS-ALG-PS-04: Justify whether a problem is best solved using procedural instructions, rule-based logic, data-driven methods, or a combination of these approaches. MS-ALG-PS-05: Use an AI tool to generate outputs that assist in solving a computational problem.	HS-ALG-PS-01: Design an algorithm using appropriate data structures to solve a problem or express an idea. HS-ALG-PS-02: Optimize the design of an algorithm using procedural abstraction and control structures. HS-ALG-PS-03: Evaluate algorithms for efficien , correctness, and clarity, using metrics or test cases. HS-ALG-PS-04: Describe the differences between deterministic and probabilistic algorithms. HS-ALG-PS-05: Evaluate AI-generated output to assess bias, accuracy, and potential harms.
	Machine Learning	EK-ALG-ML-02: Recognize attributes of objects to notice patterns and make decisions.	E1-ALG-ML-02: Recognize that AI systems are technologies that use patterns in data to make decisions or generate new things.	E2-ALG-ML-02: Examine how data is used to train a machine learning model.	E3-ALG-ML-02: Investigate how a machine learning model can change when new data is added to a training set.	E4-ALG-ML-02: Analyze relationships between the properties of training data and a machine learning model's output.	E5-ALG-ML-02: Train a machine learning model to make a classific tion or prediction.	MS-ALG-ML-06: Hypothesize how a machine learning model generates classific tions or predictions. MS-ALG-ML-07: Investigate ways to improve the accuracy of a machine learning model and reduce bias by refining the quality of examples and nonexamples in the training data. MS-ALG-ML-08: Evaluate the features and limitations of a machine learning model.	HS-ALG-ML-06: Justify the selection of a type of AI algorithm to accomplish a task. HS-ALG-ML-07: Evaluate training data by examining its source, quality, representativeness, potential biases, and privacy implications. HS-ALG-ML-08: Develop a machine learning model for a chosen task using appropriate data and tools.
	Impacts of Algorithms & Design	EK-ALG-IM-03: Describe how people create algorithms to solve problems.	E1-ALG-IM-03: Explain how a change to an algorithm leads to a different outcome.	E2-ALG-IM-03: Describe how an algorithm might impact peers in varied situations.	E3-ALG-IM-03: Compare how different algorithms may affect outcomes, situations, and people with a wide range of needs.	E4-ALG-IM-03: Evaluate how different algorithms for solving the same problem produce outcomes that may benefit or disadvantage different groups of people.	E5-ALG-IM-03: Articulate how human-centered design principles are incorporated into the development of a computing technology.	MS-ALG-IM-09: Evaluate which human-centered design principles are present or missing in an existing computing technology. MS-ALG-IM-10: Examine evidence of beneficial and har ful impacts, ethical issues, and biases of algorithms encountered in daily life. MS-ALG-IM-11: Modify an algorithm to address a specific societal impact, ethical issue, or bias.	HS-ALG-IM-09: Design a computing technology using human-centered design principles. HS-ALG-IM-10: Evaluate the ethical implications, societal impacts, and potential biases of rule-based and data-driven algorithms. HS-ALG-IM-11: Articulate the values embedded in the design of an algorithmic system.

		PK/K	Grade 1	Grade 2	Grade 3	Grade 4	Grade 5	Middle School	High School
Programming	Program Development	EK-PRO-PD-04: Create a sequence of commands to solve a problem or express an idea.	E1-PRO-PD-04: Create code from an algorithm that includes sequence to solve a problem or express an idea.	E2-PRO-PD-04: Create code from an algorithm that includes sequence, events, and iteration to solve a problem or express an idea.	E3-PRO-PD-04: Develop code from a student-created algorithm that includes a combination of sequence, events, iteration, and selection to solve a problem or express an idea. E3-PRO-PD-05: Use constructive feedback to improve a program.	E4-PRO-PD-04: Compare different programming solutions to the same problem based on correctness and clarity. E4-PRO-PD-05: Collaborate with a team by offering a meaningful contribution to creating a program.	E5-PRO-PD-04: Create a novel program by modifying or combining elements of existing programs. E5-PRO-PD-05: Construct individual components of a program that are collaboratively assembled into a programming project.	MS-PRO-PD-12: Use a procedure to structure code for clarity and reusability. MS-PRO-PD-13: Use reference documentation in program development. MS-PRO-PD-14: Justify the importance of attribution and intellectual property when developing computing technologies. MS-PRO-PD-15: Develop a program utilizing inclusive collaboration practices.	HS-PRO-PD-12: Create a modular program that uses procedures, external libraries, or objects to improve reusability and readability. HS-PRO-PD-13: Use documentation, libraries, application programming interfaces (APIs), and other tools in program development. HS-PRO-PD-14: Apply appropriate attribution of intellectual property when developing a computing technology. HS-PRO-PD-15: Collaborate on a programming project using a defined workflow t t includes design documentation and clear task roles.
	Variables & Data Storage	EK-PRO-VD-05: Identify everyday gestures and symbols that represent information people use to make choices.	E1-PRO-VD-05: Identify terms that refer to data values that change over time in everyday life.	E2-PRO-VD-05: Label different representations of information with a name and whether its value is constant or changes.	E3-PRO-VD-06: Identify the variables being stored and manipulated in a program.	E4-PRO-VD-06: Trace how data flows and changes variable values in a program.	E5-PRO-VD-06: Use variables to store, compare, and modify data within a program.	MS-PRO-VD-16: Use variables of multiple data types to store, access, and manipulate data within a program.	HS-PRO-VD-16: Create a program that uses appropriate data structures to store, access, and manipulate data.
	Reading & Documenting	EK-PRO-RD-06: Describe how a sequence of commands completes a task.	E1-PRO-RD-06: Explain the function of code that includes an event and sequence.	E2-PRO-RD-06: Explain the steps taken to create a program.	E3-PRO-RD-07: Articulate how a specific segment of code contributes to the overall purpose of a program.	E4-PRO-RD-07: Document a program to clarify its functionality.	E5-PRO-RD-07: Create embedded or external documentation for a programming project.	MS-PRO-RD-17: Analyze the roles of iteration, selection, variables, and procedures in a segment of code. MS-PRO-RD-18: Analyze AI-generated code for accuracy and usability in a programming project.	HS-PRO-RD-17: Analyze how a segment of code works, including the role of parameters, return values, and data structures. HS-PRO-RD-18: Evaluate AI-generated code for accuracy, reliability, and alignment with program requirements.
	Testing & Refining	EK-PRO-TR-07: Identify a step in a sequence of commands that does not work as expected.	E1-PRO-TR-07: Debug a program that includes a sequence of commands.	E2-PRO-TR-07: Debug a program that includes sequence, events, and iteration.	E3-PRO-TR-08: Debug a program that includes a combination of sequence, events, iteration, and selection.	E4-PRO-TR-08: Debug a program incrementally and repeatedly throughout the development process.	E5-PRO-TR-08: Debug a program using systematic strategies.	MS-PRO-TR-19: Use systematic strategies to test, refine, and document changes to a computing technology to meet the intended purpose. MS-PRO-TR-20: Refine a computing technology based on user feedback to improve its usability and accessibility.	HS-PRO-TR-19: Evaluate a computing technology's alignment with design specific tions and responsible design values, including its correctness, effectiveness, and user experience. HS-PRO-TR-20: Refine a computing technology based on user feedback, testing results, and responsible design values to improve its effectiveness and impact.

		PK/K	Grade 1	Grade 2	Grade 3	Grade 4	Grade 5	Middle School	High School
Data & Analysis	Data Collection & Preparation	EK-DAT-DC-08: Use collected data to help answer questions.	E1-DAT-DC-08: Use multiple methods to collect both numeric and non-numeric data to help answer questions.	E2-DAT-DC-08: Compare numeric and non-numeric types of data in terms of how they are collected and what information they provide.	E3-DAT-DC-09: Evaluate numeric and non-numeric data for accuracy and completeness.	E4-DAT-DC-09: Organize collected data into a table using a computational tool, with rows representing records and columns representing attributes.	E5-DAT-DC-09: Use computational tools to collect and organize different types of data.	MS-DAT-DC-21: Evaluate how different levels of precision and granularity in data collection affect accuracy, storage, and analysis. MS-DAT-DC-22: Explain how data and its associated metadata can be used to answer questions. MS-DAT-DC-23: Use a computational tool to sort, filte , group, and summarize structured data. MS-DAT-DC-24: Analyze options to address a data quality issue.	HS-DAT-DC-21: Use a computational tool to generate simulated data that fits ertain parameters for use in a simulation. HS-DAT-DC-22: Create a data dictionary that describes the name, type, and allowable values for each attribute and the logical relationships between variables in a dataset. HS-DAT-DC-23: Use a computational tool to clean and organize text-based data. HS-DAT-DC-24: Evaluate different approaches to verifying consistency and compliance with expected data types, values, and ranges.
	Data Investigation	EK-DAT-DI-09: Investigate a question that can be answered by collecting data in students' everyday environments.	E1-DAT-DI-09: Compare a question that can be answered through a data investigation and a question that can be answered through other means.	E2-DAT-DI-09: Develop a question that can be answered with data.	E3-DAT-DI-10: Investigate a data question involving relationships between multiple attributes.	E4-DAT-DI-10: Create an explanation that includes at least one data visualization to report the process and results of a data investigation.	E5-DAT-DI-10: Analyze a dataset to identify the nature and possible sources of variability in the data.	MS-DAT-DI-25: Use a computational tool to identify relationships among variables in a dataset and make a classific tion or prediction. MS-DAT-DI-26: Create data visualizations to show how different design choices can impact the interpretation of the same data. MS-DAT-DI-27: Summarize a data investigation process, including potential biases, limitations, and supporting evidence.	HS-DAT-DI-25: Create a data visualization of a multivariate dataset to answer a question or make a classific tion or prediction. HS-DAT-DI-26: Evaluate a data simulation or visualization to answer a data question, inform decision-making, and identify potential limitations.
	Impacts of Data Science	EK-DAT-IM-10: Investigate how data can help a person make informed decisions in everyday life.	E1-DAT-IM-10: Examine a variety of data questions that address the needs of a person or community.	E2-DAT-IM-10: Distinguish between data collection approaches, including those that may lead to inaccurate or biased data.	E3-DAT-IM-11: Design a data collection process that addresses the needs of people from different backgrounds or groups.	E4-DAT-IM-11: Investigate how data collected about people may affect individuals and groups.	E5-DAT-IM-11: Analyze the benefits and risks of a computing technology that uses collected data.	MS-DAT-IM-28: Explain the benefits and risks of allowing personal data and metadata to be collected and used in datasets, including issues of data ownership, privacy, and sovereignty. MS-DAT-IM-29: Analyze how decisions made at different stages of working with data can lead to biased data, misleading conclusions, and compromised AI models.	HS-DAT-IM-27: Evaluate the societal, environmental, and ethical implications of large-scale data collection and processing, including within AI applications. HS-DAT-IM-28: Debate the efficacy f a policy or regulation to ensure responsible data use.

		PK/K	Grade 1	Grade 2	Grade 3	Grade 4	Grade 5	Middle School	High School
Systems & Security	Hardware & Software	EK-SYS-HW-11: Examine the use of tools to accomplish a task or solve a problem for different users.	E1-SYS-HW-11: Describe the purpose of basic hardware components of a computing system, using accurate terminology.	E2-SYS-HW-11: Explain how the basic hardware components of a computing system work together to perform input and output (I/O) operations.	E3-SYS-HW-12: Describe the role of software in a computing system to accomplish tasks or solve problems.	E4-SYS-HW-12: Apply a basic troubleshooting process to identify and fix ommon hardware and software issues.	E5-SYS-HW-12: Explain how hardware and software components of a computing system work together to perform input and output operations, processing, and storage.	MS-SYS-HW-30: Examine differences between computing systems based on user needs, system requirements, and potential societal, environmental, and ethical impacts. MS-SYS-HW-31: Describe computing devices used in various industries, their basic functions, and how they are used to accomplish tasks or solve problems.	HS-SYS-HW-29: Differentiate an operating system as a special type of software that manages both the hardware and other software components of a computing system, including handling memory and peripherals. HS-SYS-HW-30: Demonstrate the capabilities and limitations of a physical or simulated computing device to address a task or problem.
	Security	EK-SYS-SE-12: Differentiate between public and private information.	E1-SYS-SE-12: Describe how to keep devices and online accounts safe from unauthorized access.	E2-SYS-SE-12: Explain how online actions have real-world consequences.	E3-SYS-SE-13: Evaluate how sharing information online might reveal personally identifiable information and other details.	E4-SYS-SE-13: Distinguish between authentication and authorization in protecting devices and private information.	E5-SYS-SE-13: Describe the concepts of the CIA triad and how each component is important in protecting information.	MS-SYS-SE-32: Explain the effects of not using the CIA triad when working with data. MS-SYS-SE-33: Evaluate common types of cyber attacks and preventions.	HS-SYS-SE-31: Identify different types of cybersecurity and physical security measures and the trade-offs or users, data, and devices. HS-SYS-SE-32: Classify the causes and impacts of security breaches and social engineering attacks for individuals, industries, communities, and governments. HS-SYS-SE-33: Formulate a solution to a security flaw in a given system.
	Networks				E3-SYS-NT-14: Explain how people access the internet to gain information and communicate with each other.	E4-SYS-NT-14: Compare wired and wireless methods that computing devices use to connect to the internet.	E5-SYS-NT-14: Distinguish between the components of wired and wireless networks.	MS-SYS-NT-34: Model how information in a network is broken down into packets, transmitted between devices, and reassembled. MS-SYS-NT-35: Explain how the resilience of the internet depends on interconnected devices and their roles and functions within the network.	HS-SYS-NT-34: Diagram a network of computing systems, including hardware and software. HS-SYS-NT-35: Analyze how the internet functions as a network of networks and how it differs from other types of networks.
	Impacts of Computing Systems	EK-SYS-IM-13: Identify responsible behavior when using computing systems and digital tools.	E1-SYS-IM-13: Describe an individual's role in responsibly using computing systems and digital tools.	E2-SYS-IM-13: Describe the benefits and harms that arise from an individual's use of computing systems and digital tools.	E3-SYS-IM-15: Describe how widely used computing systems may impact an individual's life and community.	E4-SYS-IM-15: Investigate the impacts of widely used computing systems on natural resources and the environment.	E5-SYS-IM-15: Examine how computing systems impact culture and the ways people live and work.	MS-SYS-IM-36: Collaborate to improve the design of a computing system to meet the needs of diverse users. MS-SYS-IM-37: Examine how access to computing systems can vary based on personal and social factors, such as physical ability, geographic location, socioeconomic status, and age.	HS-SYS-IM-36: Evaluate the rationale behind a law or policy governing the design and use of computing systems. HS-SYS-IM-37: Investigate how computing systems and infrastructure impact society and the environment, identifying who is affected and why.

		PK/K	Grade 1	Grade 2	Grade 3	Grade 4	Grade 5	Middle School	High School
Computing & Society	History of Computing	EK-SOC-HI-14: Identify computing technologies used in daily life that have changed over time.	E1-SOC-HI-14: Compare how an everyday activity changed after a specific computing technology was introduced.	E2-SOC-HI-14: Analyze the ways that people from different cultures, backgrounds, and time periods have designed computing technologies to help them solve problems and express ideas.	E3-SOC-HI-16: Examine how computing innovations have changed the ways people live, work, or communicate over time.	E4-SOC-HI-16: Investigate the contributions of diverse individuals and communities in the history of computing.	E5-SOC-HI-16: Analyze how the inclusion or exclusion of diverse individuals and communities has shaped the design, development, and societal impact of computing technologies.	MS-SOC-HI-38: Compare the roles of individuals, communities, organizations, and governments in shaping computing technologies across major eras in computing history. MS-SOC-HI-39: Analyze intended and unintended impacts of a historical computing technology on society and the environment.	HS-SOC-HI-38: Analyze the historical trajectory of a specific computing technology and how its development is linked to societal and environmental factors. HS-SOC-HI-39: Propose modific tions to an existing policy or piece of legislation that encourages ethical innovation and minimizes societal risks associated with technology.
	Emerging Technologies				E3-SOC-ET-17: Describe how new technologies create both benefits and risks in personal and family life.	E4-SOC-ET-17: Analyze how the limitations of existing technologies can lead to emerging technologies.	E5-SOC-ET-17: Examine how people decide whether or not to use emerging technologies.	MS-SOC-ET-40: Evaluate when it is appropriate to use AI and other emerging technologies to solve a problem based on their capabilities, limitations, and environmental impacts. MS-SOC-ET-41: Evaluate how the decisions made while designing an emerging technology influen e user experiences differently across different communities. MS-SOC-ET-42: Debate ways an emerging technology impacts the social, cultural, and environmental issues in local communities.	HS-SOC-ET-40: Evaluate the fundamental technological differences between an emerging technology and established technologies and how those differences influen e computing. HS-SOC-ET-41: Evaluate the societal and environmental impacts of an emerging technology, including those that lead to inequities in access and outcomes. HS-SOC-ET-42: Design a conceptual solution to a real-world problem using an emerging technology, analyzing its potential benefits and harms
	Humans & Computing	EK-SOC-HU-15: Explain that people design and develop computing technologies.	E1-SOC-HU-15: Differentiate between activities that humans do well and activities that computing technologies do well.	E2-SOC-HU-15: Investigate situations where humans have created computing technologies to solve problems.	E3-SOC-HU-18: Examine why people design and build computing technologies.	E4-SOC-HU-18: Distinguish between human learning and machine learning processes.	E5-SOC-HU-18: Evaluate when it is appropriate to use or not use computing technologies to solve a problem.	MS-SOC-HU-43: Analyze how the decisions humans make when using computing technologies have ethical and social consequences.	HS-SOC-HU-43: Evaluate how human choices in using, designing, deploying, and regulating computing technologies have risks, benefits, and long-term impacts HS-SOC-HU-44: Debate perspectives on differences between human and artificial intelligence and their implications for consciousness, ethics, and human responsibility.
	Career Exploration	EK-SOC-CE-16: Identify how people use digital devices at home, at school, and at work.	E1-SOC-CE-16: Describe how computing is used by people at home, at school, and in the community.	E2-SOC-CE-16: Investigate how personal interests connect to computing in different industries and careers.	E3-SOC-CE-19: Explain how people in different industries use computing technologies and skills to accomplish their work.	E4-SOC-CE-19: Investigate how the workforce adopts new computing technologies and continues to update their computing skills.	E5-SOC-CE-19: Examine how professionals collaborate while using computing technologies to solve problems.	MS-SOC-CE-44: Analyze how workers in different careers use computational thinking to solve real-world problems. MS-SOC-CE-45: Evaluate how automation in technology can create or replace jobs and change how people work.	HS-SOC-CE-45: Analyze how diverse teams of people use computational thinking and computing technologies to solve problems and express ideas. HS-SOC-CE-46: Connect computing knowledge and skills acquired to students' personal goals and career aspirations.

Appendix D:
High School Specialty Standards Progressions Charts

		Specialty I	Specialty II
Artificial Intelligenc	Design & Development	S1-AIN-DD-01: Analyze AI systems to differentiate the types of problems they address. S1-AIN-DD-02: Modify AI system training data to improve fairness and accuracy in outputs. S1-AIN-DD-03: Create an application using a prebuilt supervised learning model to make a classific tion or prediction. S1-AIN-DD-04: Compare data representations and how representation choice constrains applicable algorithms. S1-AIN-DD-05: Evaluate whether an AI or non-AI computational solution is appropriate for a real-world problem.	S2-AIN-DD-01: Analyze problems that can be addressed using unsupervised learning. S2-AIN-DD-02: Optimize an AI algorithm's performance for a given problem. S2-AIN-DD-03: Create an application using a complex supervised learning model.
	Data Science for AI	S1-AIN-DS-06: Examine how data flows through a neural network structure. S1-AIN-DS-07: Apply data acquisition, cleaning, and transformation techniques to prepare data for AI analysis.	S2-AIN-DS-04: Evaluate metadata and data pipelines to support transparency in AI model selection and deployment. S2-AIN-DS-05: Transform data to meet the input requirements of an AI algorithm.
	Human Responsibility	S1-AIN-HR-08: Plan safeguards for AI systems that protect human well-being and privacy while ensuring meaningful human involvement in decision-making. S1-AIN-HR-09: Analyze the potential biases and limitations of AI systems. S1-AIN-HR-10: Analyze the environmental impacts of widespread AI adoption.	S2-AIN-HR-06: Develop a policy that promotes transparency in AI applications. S2-AIN-HR-07: Analyze how model optimization choices impact a model's performance, interpretability, and fairness.
	Professional Practice	S1-AIN-PP-11: Integrate a prebuilt AI agent into an application. S1-AIN-PP-12: Analyze how AI tools shape user experiences for people with diverse backgrounds and characteristics. S1-AIN-PP-13: Assess how unauthorized data collection has influen ed the practice of training AI models. S1-AIN-PP-14: Evaluate how ethical implications of AI have changed over time.	S2-AIN-PP-08: Develop an AI agent to accomplish a defined task S2-AIN-PP-09: Develop a professional communication artifact by interpreting technical AI results for diverse audiences. S2-AIN-PP-10: Debate aspects of AI regulatory frameworks and legislation across countries.

		Specialty I	Specialty II
Cybersecurity	Network Theory & Design	S1-CYB-ND-01: Analyze the role of network services and protocols in secure communication and their potential vulnerabilities. S1-CYB-ND-02: Examine the concepts of the Open Systems Interconnection (OSI) model and their role in network communication. S1-CYB-ND-03: Distinguish between networks based on their protocols, topologies, and addressing schemes. S1-CYB-ND-04: Examine security risks in digital systems and corresponding mitigation strategies.	S2-CYB-ND-01: Integrate security features in networking hardware and software. S2-CYB-ND-02: Design a secure network, including servers, switches, routers, endpoints, and fi ewalls. S2-CYB-ND-03: Evaluate the security implications of different network topologies.
	Network Operations	S1-CYB-NO-05: Apply diagnostic tools and techniques to resolve network connectivity issues. S1-CYB-NO-06: Analyze system processes and network activity using command-line tools to identify potential security concerns.	S2-CYB-NO-04: Analyze network traffi patterns to distinguish between normal and potentially malicious behavior. S2-CYB-NO-05: Analyze cybersecurity tasks to improve efficiency and eliability.
	Threats & Security Measures	S1-CYB-TS-07: Analyze how common cyber threats related to network activity exploit system vulnerabilities. S1-CYB-TS-08: Evaluate a security threat using the CIA triad, states of data, and types of control. S1-CYB-TS-09: Analyze how encryption methods in network communication protect privacy and security. S1-CYB-TS-10: Evaluate security measures as part of a layered defense strategy to protect sensitive information.	S2-CYB-TS-06: Design access controls to protect sensitive information. S2-CYB-TS-07: Evaluate security risks to determine appropriate risk management strategies. S2-CYB-TS-08: Examine an incident response plan for a real-world scenario.
	Cybersecurity Policies	S1-CYB-CP-11: Explain the importance of cybersecurity policies in protecting organizational assets and mitigating risks. S1-CYB-CP-12: Explain key components of effective security policies.	S2-CYB-CP-09: Debate how various regulations impact organizational security policies, procedures, and compliance.
	Professional Practice	S1-CYB-PP-13: Analyze the potential consequences of a cybersecurity decision on individuals, organizations, and society. S1-CYB-PP-14: Analyze social engineering techniques and how they exploit human cognitive biases and organizational weaknesses. S1-CYB-PP-15: Translate cybersecurity concepts, risks, and solutions clearly to both technical and nontechnical audiences.	S2-CYB-PP-10: Analyze the benefits, risks, and ethical implications of AI in cybersecurity. S2-CYB-PP-11: Analyze cybersecurity vulnerabilities and incidents to inform responsible disclosure practices. S2-CYB-PP-12: Create documentation of cybersecurity processes and decisions that supports team coordination and accountability.

		Specialty I	Specialty II
Data Science	Creation & Curation	S1-DSC-CC-01: Apply exploratory data analysis techniques to non-hierarchical quantitative data. S1-DSC-CC-02: Interpret metadata when using data collected by others. S1-DSC-CC-03: Develop a program to manipulate and transform data to prepare for analysis.	S2-DSC-CC-01: Apply exploratory data analysis techniques to a hierarchical structured data source. S2-DSC-CC-02: Document the origin, structure, and preparation of a dataset to support clarity and reproducibility. S2-DSC-CC-03: Develop a program to collect and integrate data from multiple sources. S2-DSC-CC-04: Choose appropriate data to collect for a data science project based on available tools, skills, and project goals.
	Analysis & Modeling Techniques	S1-DSC-AM-04: Apply appropriate analytic and visualization techniques for categorical and quantitative data. S1-DSC-AM-05: Examine missing data and its impact on data analysis. S1-DSC-AM-06: Analyze structured categorical and/or quantitative datasets, using computational tools and libraries.	S2-DSC-AM-05: Build a model to explain relationships, make predictions, and evaluate the influen e of different variables. S2-DSC-AM-06: Evaluate strategies for handling missing data. S2-DSC-AM-07: Analyze an unstructured, mixed-type, or high-dimensional dataset using computational tools and libraries.
	Model Interpretation & Reasoning	S1-DSC-MI-07: Interpret the results of a data analysis to explain patterns, anomalies, and trends, and connect them back to the original problem or research question. S1-DSC-MI-08: Explain the appropriateness of predictive models for the specific p oblem being addressed. S1-DSC-MI-09: Explain how dataset size affects model stability and performance.	S2-DSC-MI-08: Evaluate the performance of models using established metrics. S2-DSC-MI-09: Analyze the trade-offs between interpretability, accuracy, and generalizability as they relate to model complexity. S2-DSC-MI-10: Analyze how adding or removing variables affects model behavior and performance.
	Visualization	S1-DSC-VZ-10: Create a data visualization that communicates key findings to diverse audiences. S1-DSC-VZ-11: Analyze how graphical conventions in data visualizations support accurate interpretation and how breaking conventions can mislead.	S2-DSC-VZ-11: Create a visualization that accurately represents data and avoids misleading design choices. S2-DSC-VZ-12: Critique a data visualization for misleading elements and their role in spreading misinformation.
	Professional Practice	S1-DSC-PP-12: Apply ethical principles to data collection, analysis, and communication to promote privacy, transparency, and accountability. S1-DSC-PP-13: Assess how data collection and use may impact marginalized and underrepresented groups. S1-DSC-PP-14: Document data analysis results in a format appropriate for an audience with diverse backgrounds and perspectives.	S2-DSC-PP-13: Evaluate ethical, legal, and social considerations when working with large-scale datasets, predictive models, and emerging technologies. S2-DSC-PP-14: Evaluate protective measures in data collection, usage, and governance for privacy, security, and fairness. S2-DSC-PP-15: Translate technical results for diverse audiences in professional communication artifacts.

		Specialty I	Specialty II
Game Development	Design & Development	S1-GMD-DD-01: Analyze the fundamental components of a game, including players, rules, game mechanics, and outcomes. S1-GMD-DD-02: Create a storyboard to plan and communicate a game's narrative and interactive experience. S1-GMD-DD-03: Develop an interactive experience to support gameplay. S1-GMD-DD-04: Adapt rule-based logic to improve control of game agents or systems.	S2-GMD-DD-01: Justify the use of responsible design principles in developing an engaging, meaningful game experience. S2-GMD-DD-02: Develop a system that programmatically controls 2D or 3D animation states based on game logic and player interactions. S2-GMD-DD-03: Construct in-game artificial intelligence to control NPC behavior.
	Player Experience	S1-GMD-PX-05: Create a user-friendly interface for a game, considering accessibility, usability, and aesthetics. S1-GMD-PX-06: Optimize basic game usability through user testing and identifying actionable design revisions.	S2-GMD-PX-04: Evaluate game interactions using diverse input devices to enhance immersion and player experience. S2-GMD-PX-05: Evaluate two versions of a game using defined metrics to determine which performs better.
	Architecture	S1-GMD-AR-07: Examine the key architectural features of computing hardware and their implications for game development.	S2-GMD-AR-06: Analyze game system architectures that support scalability, maintainability, and performance.
	Professional Practice	S1-GMD-PP-08: Analyze the ethical implications of copyright and intellectual property in game development. S1-GMD-PP-09: Develop a game project collaboratively within a diverse team.	S2-GMD-PP-07: Create a game utilizing ethical principles in the design and development process. S2-GMD-PP-08: Collaborate effectively in a project team by defining and e ecuting assigned roles, communicating clearly, and managing shared resources to deliver a game project.

		Specialty I	Specialty II
Physical Computing	Hardware & Circuit Design	S1-PHY-HC-01: Construct an electrical circuit to power and control a physical computing device.	S2-PHY-HC-01: Develop an electromechanical system using a CAD tool for design and testing.
	Inputs & Outputs	S1-PHY-IO-02: Integrate sensors, actuators, and peripherals into a physical computing system.	S2-PHY-IO-02: Design a closed-loop feedback control system to meet a desired outcome using varying sensor types.
	Design & Development	S1-PHY-DD-03: Develop software to control a physical device using an iterative design process.	S2-PHY-DD-03: Develop an application that extends functionality and user engagement with physical devices.
	Connectivity & Internet of Things	S1-PHY-CI-04: Use IoT devices to collect sensor data and transmit it locally using device-to-device or device-to-gateway communication. S1-PHY-CI-05: Implement IoT communication by connecting physical devices with protocols, applying security practices.	S2-PHY-CI-04: Develop an IoT system to manage the collection and transmission of data and enable remote monitoring and control of a physical system. S2-PHY-CI-05: Develop an embedded network, evaluating trade-offs among network protocols, security measures, and scalability.
	Professional Practice	S1-PHY-PP-06: Use a version control system to coordinate development in a physical computing project. S1-PHY-PP-07: Evaluate social, technical, and sociotechnical impacts of a physical computing system. S1-PHY-PP-08: Evaluate the security implications of a physical computing system, including data privacy, unauthorized access, and potential vulnerabilities.	S2-PHY-PP-06: Apply a project management methodology to collaboratively design, develop, and test a physical computing project. S2-PHY-PP-07: Evaluate how physical computing technologies shape social processes, communities, power, and equity in global society. S2-PHY-PP-08: Develop a physical computing solution for diverse users using the engineering design process.

		Specialty I	Specialty II
Software Development	Design & Development	S1-SWD-DD-01: Implement linear data structures to organize and access collections of data when solving computational problems. S1-SWD-DD-02: Design software that accounts for complexity through abstraction. S1-SWD-DD-03: Use integrated development environment (IDE) features to streamline software development, including code editing, debugging, version control, and project management.	S2-SWD-DD-01: Apply associative and hierarchical data structures to solve computational problems. S2-SWD-DD-02: Develop a prototype using diverse user personas and user journey maps to guide design decisions. S2-SWD-DD-03: Develop a project in iterative cycles, documenting changes and the rationale for each cycle. S2-SWD-DD-04: Modify an existing algorithm to improve efficien , considering factors such as data structures and algorithmic paradigms.
	User Experience	S1-SWD-UX-04: Refine a user interfa e design using accessibility and responsive design tools. S1-SWD-UX-05: Create software that serves intended users and contexts using human-centered design principles.	S2-SWD-UX-05: Evaluate the user experience to improve usability. S2-SWD-UX-06: Create a user-friendly, accessible, and responsively designed interface.
	Testing & Refining	S1-SWD-TR-06: Use AI-assisted IDE tools or features to understand unfamiliar code and identify errors during debugging. S1-SWD-TR-07: Design a test plan that exercises program functionality, including edge cases, error conditions, and varied user inputs.	S2-SWD-TR-07: Use an IDE to test and refine a complex software project. S2-SWD-TR-08: Debug a complex software project to identify, isolate, and fix er ors.
	Professional Practice	S1-SWD-PP-08: Collaborate to develop software that reflects diverse perspectives	S2-SWD-PP-09: Apply an industry-standard software development process to plan and deliver a project while prioritizing equity and justice.

		Specialty I	Specialty II
X+CS	X+CS	S1-XCS-XC-01: Explain connections between CS concepts and practices and those from a non-CS discipline (X). S1-XCS-XC-02: Formulate a computational solution to a problem in a non-CS discipline (X) using computational thinking. S1-XCS-XC-03: Analyze a computer science technology or approach used in a non-CS discipline (X). S1-XCS-XC-04: Model relationships within a non-CS discipline using data, visualizations, and computational methods. S1-XCS-XC-05: Assess how well an algorithm solves problems within a non-CS discipline (X) by evaluating its accuracy, efficien , and relevance to the intended goal.	

https://csteachers.org/pk12standards/

ISBN 979-8-4007-2780-1