

WISCONSIN STANDARDS FOR **Computer Science**



WISCONSIN STANDARDS FOR **Computer Science**

December 2025



Wisconsin Department of Public Instruction
Jill K. Underly, State Superintendent
Madison, Wisconsin

This publication is available from:
Wisconsin Department of Public Instruction
201 W Washington Ave
Madison, WI 53703
(608) 266-8960
<http://dpi.wi.gov/computer-science>

December 2025
Wisconsin Department of Public Instruction

The Wisconsin Department of Public Instruction does not discriminate on the basis of sex, race, color, religion, creed, age, national origin, ancestry, pregnancy, marital status or parental status, sexual orientation, or ability and provides equal access to the Boy Scouts of America and other designated youth groups.

Table of Contents

Foreword.....	2
Acknowledgements.....	3
Section I: Wisconsin’s Approach to Academic Standards	4
Purpose of the Document.....	4
What Are Academic Standards?	5
Relating the Academic Standards to All Students	6
Ensuring a Process for Student Success.....	8
Section II: Wisconsin Standards for Computer Science	9
What is Computer Science Education in Wisconsin?.....	9
Wisconsin’s Approach to Academic Standards for Computer Science	10
Content Standards Structure	11
Section III: Computer Science Standards	13
Algorithms and Design (ALG)	14
Programming (PRO)	19
Data and Analysis	23
Computing Systems and Security (CSS)	31
Computing and Society (FUT)	38

Foreword

In Fall 2025, I formally adopted the Wisconsin Standards for Computer Science. This revised set of academic standards provides a foundational framework that identifies what students should know and be able to do in computer science.

Computer science is an essential part of a comprehensive PK-12 education for all students. The knowledge, techniques, and essential skills for navigating a digital, interconnected world are supported through computer science education in Wisconsin schools to advance the overall goal of helping all students become college and career ready. Wisconsin's 2025 standards for computer science also result in the following:

- Wisconsin students are cultivated learners who approach challenges with curiosity, and logic; integrating computational thinking alongside creative reasoning to solve problems throughout interdisciplinary contexts.
- Wisconsin students learn to navigate the digital landscape thoughtfully, developing a strong sense of agency, empathy, and ethics in their use of technology.
- Wisconsin students promote collaboration and diverse perspectives, encourage students to express ideas clearly, adapt to new tools, and engage in inclusive teamwork.

The Wisconsin Department of Public Instruction will continue to build on this work to support implementation of the standards with resources for the field. I am excited to share the Wisconsin Standards for Computer Science, which aim to build skills, knowledge, and engagement opportunities for all Wisconsin students.

Jill K. Underly, PhD
State Superintendent

Acknowledgements

The Wisconsin Department of Public Instruction (DPI) wishes to acknowledge the ongoing work, commitment, and various contributions of individuals to develop our state's first academic standards for computer science. Thank you to the State Superintendent's Standards Review Council for their work and guidance through the standards process. A special thanks to the Computer Science Writing Committee for taking on this important project that will shape the classrooms of today and tomorrow. Thanks to the many staff members across the division and other teams at DPI who have contributed their time and talent to this project. Finally, a special thanks to Wisconsin educators, businesspeople, parents, and citizens who provided comment and feedback to drafts of these standards.

Wisconsin Standards for Computer Science Writing Team

Co-Chairs: **Dennis Brylow**, Professor of Mathematics, Statistics, and Computer Science, Marquette University
Tina Lemmens, Math and Computer Science Teacher Education Consultant, CESA 7

DPI Liaison: **Amy Bires**, Computer Science and Digital Learning Innovation Consultant

Dan Hanley
Cashton High School

Kate Kaster
Stevens Point High School

Amy Fetherston
Wauwatosa West High School

Samantha Kable
New London High School

Jake Krueger
Greenfield High School

Mary Walz
Sauk Prairie High School

Fatima Tlemsani
Brookfield East High School

Michael Cullen
Marquette University

Joe Kmoch
Retired Math & Computer Science
Teacher

Wisconsin Department of Public Instruction, Academic Standards

Gregory Coleman, Education Administrative Director, PK-12 Standards and Instruction

Laura Roeker, Education Administrative Director, Bureau of Teaching and Learning

Josh Robinson, Assistant State Superintendent, Division of Learning Support

Wisconsin Department of Public Instruction Leaders

John W. Johnson, PhD, Policy Initiatives Advisor, Office of the State Superintendent

Sara Knueve, Assistant Deputy State Superintendent

SECTION I

Wisconsin's Approach to Academic Standards

Purpose of the Document

The purpose of this document is to improve computer science education for students and communities across Wisconsin. The Wisconsin Department of Public Instruction (DPI) has developed these standards to assist educators, administrators, and community members in understanding, developing, and implementing high-quality computer Science education programs in school districts statewide.

This publication provides a vision for student success in computer science education and aligns with [The Guiding Principles for Teaching and Learning](#) (2011). In brief, these principles are:

1. Every student has the right to learn.
2. Instruction must be rigorous and relevant.
3. Purposeful assessment drives instruction and affects learning.
4. Learning is a collaborative responsibility.
5. Students bring strengths and experiences to learning.
6. Responsive environments engage learners.

Computer science program leaders and educators will find this guide valuable for making informed decisions related to:

- Program structure and integration
- Curriculum redesign
- Staffing and staff development
- Scheduling and student grouping

- Facility organization
- Learning spaces and materials development
- Resource allocation and accountability
- Collaborative work with other units of the school, district and community

What Are Academic Standards?

Wisconsin Academic Standards specify what students should know and be able to do. They serve as goals for teaching and learning. Setting high standards enables students, parents, educators, and citizens to know what students should have learned at a given point in time. In Wisconsin, all state standards serve as a model. Locally elected school boards adopt academic standards in each subject area to best serve their local communities. We must ensure that all children have equal access to high-quality educational programs. Clear statements about what students must know and be able to do are essential in making sure our schools offer opportunities to get the knowledge and skills necessary for success beyond the classroom.

Adopting these standards is voluntary. Districts may use the academic standards as guides for developing local grade-by-grade level curricula. Implementing standards may require some school districts to upgrade school and district curricula. This may result in changes in instructional methods and materials, local assessments, and professional development opportunities for the teaching and administrative staff.

What is the Difference Between Academic Standards and Curriculum?

Standards are statements about what students should know and be able to do, what they might be asked to do to give evidence of learning, and how well they should be expected to know or do it. Curriculum is the program devised by local school districts used to prepare students to meet standards. It consists of activities and lessons at each grade level, instructional materials, and various instructional techniques. In short, standards define what is to be learned at certain points in time and, from a broad perspective, what performances will be accepted as evidence that the learning has occurred. Curricula specify the details of the day-to-day schooling at the local level.

Developing the Academic Standards

DPI has a transparent and comprehensive process for reviewing and revising academic standards. The process begins with a notice of intent to review an academic area with a public comment period. The State Superintendent's Academic Standards Review Council examines those comments and may recommend revision or development of standards in that academic area.

The state superintendent authorizes whether or not to pursue a revision or development process. Following this, a state writing committee is formed to work on those standards for all grade levels. That draft is then made available for open review to get feedback from the public, key stakeholders, educators, and the legislature, with further review by the State Superintendent's Academic Standards Review Council. The State Superintendent then determines the adoption of the standards.

Aligning for Student Success

To build and sustain schools that support every student in achieving success, educators must work together with caregivers, community members, and business partners to connect the most promising practices in the most meaningful contexts. The release of the Wisconsin Standards for Computer Science education provides a set of important academic standards for school districts to implement. This is connected to a larger vision of engaged learners creating a better Wisconsin together. Academic standards work together with other critical principles and efforts to educate every child to be an engaged learner capable of creating a better Wisconsin together. Here, the vision and Guiding Principles form the foundation for building a supportive process for teaching and learning rigorous and relevant content. The following sections articulate this integrated approach to increasing student success in Wisconsin schools and communities.

Relating the Academic Standards to All Students

Academic standards should allow all students to engage, access, and be assessed in ways that fit their strengths, needs, and interests. This applies to students with individualized education plans (IEPs), English learners, and gifted and talented pupils, consistent with all other students. Academic standards serve as a foundation for individualized programming decisions for all students.

Academic standards serve as a valuable basis for establishing concrete, meaningful goals for each student's developmental progress and demonstration of proficiency. Students with IEPs are provided with specially designed instruction that meets their individual needs. It is expected that each individual student with an IEP will require unique services and supports matched to their strengths and needs in order to close achievement gaps in grade-level standards. Alternate standards are only available for students with the most significant cognitive disabilities.

Gifted and talented students may achieve well beyond the academic standards and move into advanced grade levels or into advanced coursework.

Connection to DPI Vision and Mission

This work is anchored in DPI's current statements:

- **Our Vision:** *Engaged learners creating a better Wisconsin together.*
- **Our Mission:** *To advance equitable, transformative, and sustainable educational experiences that develop learners, schools, libraries, and communities in Wisconsin.*

Guided by Principles

All educational initiatives are guided and impacted by important and often unstated attitudes or principles for teaching and learning. [The Guiding Principles for Teaching and Learning](#) (2011) were drawn from research and provide the touchstone for practices that truly affect the vision of “engaged learners creating a better Wisconsin together.” When made transparent, these principles inform what happens in the classroom, direct the implementation and evaluation of programs, and, most importantly, remind us of our own beliefs and expectations for students.

Engaging Learners Through Career Readiness

When educators connect their students' learning to future career opportunities, they begin to engage students in a very personal and powerful way. In addition to career readiness as a strategy to engage learners, it is also a conduit through which every student in Wisconsin, including students with an IEP, can graduate from high school with the knowledge, skills, and abilities needed to be successful in their chosen career pathway. Regardless of the postsecondary path that a graduate pursues immediately after their K-12 education, we believe in preparing all students to be lifelong learners and acknowledge that one's education and career path are inextricably linked.

The Wisconsin Career Readiness Standards (WCRS) provide the framework for educators to integrate career-readiness skills across all disciplines and at every grade level from K through 12. Because people begin to develop interests and biases at an early age, it is important to start integrating WCRS in the elementary grades. By middle school, students may have already developed beliefs about their abilities related to careers. Or they may have formed stereotypes about which careers are appropriate for a particular gender, race, or socioeconomic background. Exposing students to careers and helping them develop skills related to careers when they are young is one way to keep students' minds open to all possibilities.

Implementing the Wisconsin Career Readiness Standards may look different for every teacher, every program, every course, and potentially every unit or lesson. These standards were designed to be naturally and intentionally integrated into other content standards. The Wisconsin Career Readiness Standards can be found here.

Ensuring a Process for Student Success

For Wisconsin schools and districts, implementing [Wisconsin's Framework for Multi-Level Systems of Supports \(2025\)](#) means providing equitable services, practices, and resources to every learner based on responsiveness to effective instruction and intervention. In this system, high-quality instruction, strategic use of data, and collaboration interact within a continuum of supports to facilitate learner success. Schools provide varying types of supports with differing levels of intensity to proactively and responsibly adjust to the needs of the whole child. These include the knowledge, skills, and habits learners need for success beyond high school, including developmental, academic, behavioral, social, and emotional skills.

Connecting to Content: Wisconsin Academic Standards

Within this vision for increased student success, rigorous, internationally benchmarked academic standards provide the content for high-quality curriculum and instruction and for a strategic assessment system aligned to those standards. With the adoption of the standards, Wisconsin educators have the tools to design curriculum, instruction, and assessments to maximize student learning. The standards articulate what we teach so that educators can focus on how instruction can best meet the needs of each student. When implemented within an equitable multilevel system of support, the standards can help to ensure that engaged learners create a better Wisconsin, together.



SECTION II:

Wisconsin Standards for Computer Science

What is computer science education?

Wisconsin defines computer science (CS) as an academic discipline that encompasses the study of computers and algorithmic processes, including their principles, their hardware and software designs, their applications, networks, and their impact on society. The standards outlined in this document provide an important foundation to prepare students for post-secondary education and careers.

Computer science education in Wisconsin

Computer science is a cornerstone of innovation, problem solving, and career readiness across every sector of Wisconsin's economy and society. As technology continues to shape how we live, learn, and work, foundational knowledge in computer science—especially computational thinking, data literacy, and ethical decision-making—is essential for all students, regardless of their chosen pathway.

The Wisconsin Standards for Computer Science (2025) reflect a vision where computer science is for everyone. By building age-appropriate, interdisciplinary learning experiences from elementary through high school, we ensure students develop the skills and confidence to engage with the digital world thoughtfully and creatively. Elementary educators integrate CS through activities that build logic, pattern recognition, and collaboration. Middle and high school students deepen their CS understanding with structured coursework, design projects, and opportunities for advanced study.

These standards establish clear end-of-grade-band expectations to support high-quality, inclusive instruction. Students who receive special education services through an Individualized Education Program (IEP), students with gifts and talents, multilingual learners, and other students with diverse needs may benefit from additional support or extensions. Educators are encouraged to apply Universal Design for Learning (UDL) principles and collaborate with families and support staff to personalize the learning experience for every student.

Computer science in Wisconsin is more than a content area—it's a commitment to access, agency, and adaptability. Through thoughtful implementation and educator collaboration, CS prepares students to design solutions, think critically, and contribute meaningfully to their communities and futures.

Wisconsin's Vision for K–12 Computer Science

Wisconsin's vision for computer science education is shaped by local educators, industry leaders, and community partners, and is informed by national research and best practices. The goal is to ensure all students have meaningful opportunities to understand, apply, and shape computer science concepts throughout their K–12 experience—regardless of background or future aspirations.

Wisconsin envisions a future where:

1. All students are introduced to fundamental CS ideas starting in elementary school, with opportunities to explore problem solving, computational thinking, and ethical technology use through integrated and interdisciplinary learning.
2. Middle and high school students engage in accessible, rigorous CS coursework that is recognized for graduation credit and prepares them for advanced study, career pathways, and civic participation in a digital world.
3. Students have opportunities to study specialized facets of computer science, including artificial intelligence, data science, cybersecurity, and software development, while building transferable skills like collaboration, design thinking, and systems awareness.
4. Historically underrepresented groups in CS—including girls, students of color, multilingual learners, and students with disabilities—are actively supported and represented, with equitable access to coursework, role models, and pathways that empower and inspire.

This vision aligns with Wisconsin's commitment to equity, innovation, and lifelong learning—ensuring that computer science is not just available, but engaging, inclusive, and transformative for every student.

Wisconsin's Approach to Academic Standards for Computer Science

The Wisconsin Standards for Computer Science provide K–12 educators with clear guidance to support the development of equitable, high-quality learning experiences that build both foundational knowledge and future-ready skills. These standards reflect a shared commitment among educators, business and industry leaders, and computer science practitioners to prepare all students for success in a rapidly evolving digital world.

Developed through collaboration with stakeholders across Wisconsin, the standards are grounded in national best practices and ongoing professional dialogue. They support formal coursework and integration across disciplines, ensuring students can engage with computer science as both a stand-alone subject and a lens for solving real-world problems.

Wisconsin's academic standards were informed by foundational documents including the K–12 Computer Science Framework

(<https://k12cs.org/>) and multiple iterations of the Computer Science Teachers Association (CSTA) K–12 Standards, which are currently undergoing revision through the Reimagining CS Standards project.

The Wisconsin Standards for Computer Science are organized around five conceptual strands drawn from this foundational content:

- Algorithms and Design
- Programming
- Data and Analysis
- Computing Systems and Security
- Computing and Society

These strands reflect the essential skills, knowledge, and dispositions needed to understand computer science as a discipline and to apply CS ideas across domains—from science and mathematics to civics, art, and beyond.

Wisconsin’s approach emphasizes flexibility and local innovation. CS standards may be taught through dedicated computer science courses, integrated within other subject areas, or introduced through interdisciplinary projects and design-based learning. Each district, school, and educational program is encouraged to determine the most impactful and inclusive way to meet these standards and support all learners.

Whether designing algorithms, exploring data, or addressing issues of equity and ethics in technology, Wisconsin students deserve access to computer science learning that is meaningful, developmentally appropriate, and built for the challenges and possibilities of the future.

Content Standards Structure

Standards Coding

Standards Formatting

- **Standard:** Broad statement that tells what students are expected to know or be able to do
- **Learning priority:** Breaks down the broad statement into manageable learning pieces
- **Performance indicator by grade band:** Measurable degree to which a standard has been developed or met

Grade Bands:

Grade bands of K-5, 6-8, and 9-12 align to typical elementary, middle, and high school levels.

- Grade band K-5 performance indicators represent knowledge and skills that should be integrated throughout the elementary curriculum.
- Computer science education should be part of the core curriculum for all middle school students. Awareness, exploration, and building foundational skills should occur in middle school.
- Computer science education at the high school level continues to develop student foundational understanding of CS in the world through in-depth CS learning, including awareness and exploration activities.

References

Wisconsin Standards for Computer Science 12 References Computer Science Teachers Association (2011). CSTA K-12 Computer Science Standards, Revised 2011. Retrieved from https://c.ymcdn.com/sites/www.csteachers.org/resource/resmgr/Docs/Standards/CSTA_K-12_CSS.pdf.

Reimagining CS. Foundational Content Overview. Retrieved from <https://reimaginingcs.org/foundational-content/>

Computer Science Teachers Association (2025). CSTA K-12 Computer Science Standards, Revised 2017. Retrieved from <http://www.csteachers.org/standards>.

K-12 Computer Science Framework. (2016). Retrieved from <http://www.k12cs.org>.

The National Career Clusters® Framework. (2023). Retrieved from <https://careertech.org/career-clusters/>

SECTION III

Computer Science Standards

Algorithms and Design (ALG)

Programming (PRO)

Data and Analysis (DA)

Computing Systems and Security (CSS)

Computing and Society (FUT)

Algorithms and Design (ALG)

Standard: ALG.1: Fundamentals: *Students will identify and develop algorithms to solve problems across a variety of contexts.*

Performance Indicators (By Grade Band)

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
ALG.1.a: Use algorithms to show how problems can be solved.	ALG.1.a.e.1: Identify algorithms as step-by-step instructions in daily activities, and represent them using visual formats such as diagrams, arrows, or storytelling.	ALG.1.a.m.1: Represent algorithms visually using structured formats such as flowcharts, pseudocode, or block-based tools to solve computational problems.	ALG.1.a.h.1: Optimize algorithmic representations using abstraction and encapsulation to improve clarity, reusability, and efficiency.
	ALG.1.a.e.2: Model daily processes by creating and following algorithms that include sequence, events, and repetition to complete tasks and solve problems.	ALG.1.a.m.2 Model decision-making and repetition in algorithm representations to reflect problem-solving strategies.	ALG.1.a.h.2: Model computational algorithms using structured visual formats such as flowcharts or pseudocode to demonstrate problem-solving processes.
ALG.1.b: Use communication tools to describe how data will be transformed	ALG.1.b.e.1: Use visual tools to represent how data moves and changes through sequence, events, and repetition.	ALG.1.b.m.1: Describe how data is taken in (as input), stored, processed, and then produced as a result (as output) in a computational solution.	ALG1.b.h.1: Describe the differences between deterministic algorithms and probabilistic algorithms.

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
	<p>ALG.1.b.e.2: Describe problem-solving steps by creating representations of algorithms that include basic elements such as selection and variables.</p>	<p>ALG.1.b.m.2: Represent and explain data transformations using visual or textual formats such as flowcharts, pseudocode, or structured diagrams.</p>	<p>ALG1.b.h.2: Communicate how abstraction and encapsulation shape data flow and transformation in complex algorithms.</p>
<p>ALG.1.c: Design algorithms to solve problems using structured steps and logical reasoning.</p>	<p>ALG.1.c.e.1: Write the steps in algorithms that include sequence, events, iteration, and selection to complete a task or solve a problem using everyday language.</p>	<p>ALG.1.c.m.1: Write algorithm steps using sequence, iteration, and selection to solve a task involving at least one data element.</p>	<p>ALG.1.c.h.1: Develop algorithms using variables, data, and storage systems that solve real-world problems with authenticity and efficiency.</p>
	<p>ALG.1.c.e.2: Create simple representations of algorithms through storytelling, flowcharts, or block-based tools to show how tasks are completed.</p>	<p>ALG.1.c.m.2: Create and refine algorithms using structure formats (e.g. pseudocode, or flowcharts), incorporating inputs, logics, and outputs.</p>	<p>ALG.1.c.h.2: Design and justify algorithmic decisions - such as the use of sequence, selection, or iteration - to optimize clarity, performance, and accuracy.</p>

Standard: ALG.2: Human-centered Design: *Students will design algorithms intended to help people by assessing user needs through communication, and evaluating them for fairness, accessibility, and inclusivity.*

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
ALG.2.a: Design, evaluate and refine algorithms based on user needs and interfacing with the user effectively.	ALG.2.a.e.1: Discuss how human problems might be solved with the assistance of algorithms or programs.	ALG.2.a.m.1: Design algorithms, using human-centered design principles such as empathy, user needs and requirements, and accessibility.	ALG.2.a.h.1: Design and evaluate algorithms for diverse users, incorporating feedback and considering fairness, impact, and potential harms.
	ALG.2.a.e.2: Develop an algorithm to solve a problem by considering others' needs and ideas, and sharing feedback, using a process that considers the needs, requirements, and feedback of others.	ALG.2.a.m.2: Refine algorithms iteratively through user feedback to improve usability, accessibility, and user experience.	ALG.2.a.h.2: Evaluate and refine algorithms based on diverse user feedback, prioritizing accessibility, fairness, and the identification of potential harm or unintended consequences.

Standard: ALG.3: Problem Solving: *Students will improve algorithmic solutions by comparing alternatives and using pattern recognition and decomposition to make them more efficient and accurate.*

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
ALG.3.a: Describe and analyze how algorithms solve a problem and compare different solutions to improve accuracy and efficiency.	ALG.3.a.e.1: Describe how different parts of an algorithm or task work together to get something done.	ALG.3.a.m.1: Compare different sets of steps to see which works better or faster.	ALG.3.a.h.1: Develop two or more algorithms to solve the same problem and explain which one works best.
	ALG.3.a.e.2: Compare different sets of steps to see which works better or faster.	ALG.3.a.m.2: Compare two or more algorithms to decide which is more accurate or efficient.	ALG.3.a.h.2: Use metrics or test cases to evaluate correctness, clarity, and efficiency in different solutions.
	ALG.3.a.e.3: Improve basic algorithms by adding steps like repetition or decisions to make them more accurate or efficient.	ALG.3.a.m.3: Suggest improvements to make an algorithm work better or faster based on what the problem needs.	ALG.3.a.h.3: Explain how improving an algorithm's structure or logic can lead to better results or faster performance.
ALG.3.b: Break down problems and analyze algorithms using computational thinking strategies like patterns, decomposition, and abstraction.	ALG.3.b.1.e: Decompose a problem or task into smaller steps to design an algorithm that solves it.	ALG.3.b.1.m: Explain how a system gives outputs even if the parts inside aren't fully visible (like a black box).	ALG.3.b.1.h: Improve algorithms by organizing repeated actions or steps into functions, modules, or reusable parts.
	ALG3.b.2.e Find patterns in problems or tasks that can make algorithms simpler or faster.	ALG3.b.2.m Use patterns or structure in problems to help predict how algorithms work and where they can be improved.	ALG3.b.2.h Use abstraction—like lists, objects, or procedures—to make algorithms clearer and more efficient.

Standard: ALG.4: Impacts of Algorithms: *Students will explain how algorithm decisions affect different users, and identify issues like bias, fairness, and accessibility in the results.*

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
<p>ALG.4.a: Examine how algorithms can be unfair or inaccessible, and analyze their impact on different people and communities.</p>	<p>ALG.4.a.e.1: Compare how different solutions might treat people differently, and talk about when they work well for some but not for others.</p>	<p>ALG.4.a.m.1: Describe how algorithms—including deterministic and probabilistic types—can impact people or groups unfairly, and explain why accessibility and bias matter.</p>	<p>ALG.4.a.h.1: Evaluate how algorithm decisions affect different users and communities by identifying potential bias, ethical concerns, and unintended consequences in both deterministic and probabilistic models.</p>
<p>ALG.4.b: Implement changes to algorithms to resolve issues of fairness, accessibility, or bias.</p>	<p>ALG.4.b.e.1: Examine different perspectives, abilities, and points of view when designing algorithms and programs.</p>	<p>ALG.4.b.m.1: Modify an algorithm to address a specific societal impact, ethical issue, or bias.</p>	<p>ALG.4.b.h.1: Redesign an algorithm to address fairness, accessibility, or bias by analyzing its impact and implementing targeted improvements for equity.</p>

Programming (PRO)

Standard PRO.1: *Programming Fundamentals: Students will create computational artifacts using fundamental programming skills to solve problems and express ideas.*

Performance Indicators (By Grade Band)

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
PRO.1.a: Apply programming fundamentals to create and implement solutions that address a given problem.	PRO.1.a.e.1: Create code from an algorithm that includes sequence, events, and iteration to express ideas or complete a task.	PRO.1.a.m.1: Analyze how a segment of code works, identifying the roles of variables, selection, and iteration.	PRO.1.a.h.1: Analyze the purpose and structure of code segments, including the role of variables, selection, iteration, and procedures.
		PRO.1.a.m.2: Use procedures without parameters to structure code for clarity and readability.	PRO.1.a.h.2: Convert an algorithm written in pseudocode into a program that uses sequence, selection, iteration, and procedures with parameters.

Standard PRO.2: Data Handling: *Students will organize and manipulate data using appropriate data structures.*

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
PRO.2.a: Organize and present data using common structures to highlight patterns, support reasoning, and guide decision-making.	PRO.2.a.e.1: Identify and explain how symbols and terms in everyday life represent information, including values that remain constant or change over time.		
	PRO.2.a.e.2: Label and describe variables and fundamental data types (e.g., integers, strings, Booleans) used to represent and manage information in programs.	PRO.2.a.m.2: Use variables and fundamental data types (e.g., integers, strings, Booleans) to represent and organize data.	PRO.2.a.h.2: Compare and contrast fundamental data types and their uses.
PRO.2.b: Apply variables to store and manage different kinds of information to help programs respond and perform tasks.	PRO.2.b.e.1: Identify and trace how variables are stored, manipulated, and changed throughout a program.	PRO.2.b.m.1: Use appropriate data types and variables to store, update, and manage data in programs.	PRO.2.b.h.1: Design and implement programs that utilize complex data types and variables to efficiently store, manipulate, and maintain data integrity, including the use of scope, type conversion, and memory management principles.
PRO.2.c: Use basic data structures to group related information.	PRO.2.c.e.1: Use variables to store, compare, and modify data.	PRO.2.c.m.1: Use appropriate data structures to store related data and iterate them to process elements.	PRO.2.c.h.1: Create programs that use data structures and iteration to store, access, manipulate, and generalize solutions.

Standard PRO3: Testing and Refining Code: *Students will be able to test and debug programs to ensure accuracy, efficiency, and reliability.*

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
PRO.3.a: Test programs to find errors and use debugging strategies to fix problems and improve how the program works.	PRO.3.a.1.e: Analyze and debug a program, which includes sequencing, events, loops, conditionals, and variables.	PRO.3.a.1.m: Use testing and debugging methods to ensure program correctness and completeness.	PRO.3.a.1.h: Use a systematic approach and debugging tools to independently debug a program (e.g., setting breakpoints, inspecting variables with a debugger).

Standard PRO4: Project Management: *Students will be able to plan, manage, and reflect on the development of a coding project from start to finish.*

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
PRO.4.a: Clarify project goals by identifying the problem and gathering user requirements, then plan steps and timelines to guide a solution.	PRO.4.a.e.1: Describe the purpose of a software project, set a simple goal, and use checklists or visual tools (e.g., timelines or storyboards) to plan what needs to be completed.	PRO.5.a.m.1: Define clear project objectives and develop a step-by-step plan or timeline to guide workflow.	PRO.4.a.h.1: Define project goals based on user needs or requirements and develop a detailed plan with milestones and deadlines—using methods like Agile or SCRUM to guide iterative progress.

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
<p>PRO.4.b: Track and share project progress clearly by documenting work and communicating updates to others.</p>	<p>PRO.4.b.e.1: Share and explain what the code does using drawings, notes, or short class presentations.</p>	<p>PRO.4.b.m.1: Document design choices and development process through notes, journals, or basic version tracking.</p>	<p>PRO.4.b.h.1: Maintain detailed documentation of code and decision-making process, use version control tools (e.g., Git).</p>
		<p>PRO.4.b.m.2: Communicate progress and challenges to peers or instructors.</p>	<p>PRO.4.b.h.2: Communicate progress through reports, presentations, or team meetings.</p>
<p>PRO.4.c: Work collaboratively to gather feedback, reflect on your own decisions, and make meaningful revisions to improve project outcomes.</p>	<p>PRO.4.c.e.1: Review finished work, talk about what went well or what was hard.</p>	<p>PRO.4.c.m.1: Reflect on project outcomes and process, identify areas for improvement.</p>	<p>PRO.4.c.h.1: Evaluate the effectiveness of solutions, reflect on team and individual contributions.</p>
	<p>PRO.4.c.e.2: Review code with peers to improve projects.</p>	<p>PRO.4.c.m.2: Revise code or plan based on testing or peer feedback.</p>	<p>PRO.4.c.h.2: Revise project using structured feedback and performance testing results.</p>

Data and Analysis (DA)

Standard: DA.1: Data Fundamentals

Students will develop foundational knowledge of how data is represented, stored, and processed in computing systems.

Performance Indicators (By Grade Band)

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
DA.1.a: Gather data to support computational problem solving.	DA.1.a.e.1: Collect data using a variety of methods (e.g., observation, survey, measurement) and tools (e.g., sensors, online forms), including overtime and from multiple sources.	DA.1.a.m.1: Collect both quantitative and qualitative data relevant to a problem or question using computational tools (e.g., spreadsheets, sensors, forms).	DA.1.a.h.1: Generate data that fits specific parameters (e.g., random samples, normal distributions) and interpret patterns and variability using computational tools (e.g., spreadsheets, programs, or simulations).
	DA.1.a.e.2: Describe basic information about data collected, such as what the data is about and how it was gathered, to help understand and use the data.	DA.1.a.m.2: Generate and interpret metadata to describe the purpose, structure, source, and collection of methods of datasets, supporting accurate data interpretation and reuse.	DA.1.a.h.2: Develop and apply data dictionaries to describe datasets, including attribute names and types, allowable values or ranges, and logical relationships among variables, supporting data integrity and reuse.
DA.1.b: Manipulate and represent data to support computation and communication.	DA.1.b.e.1: Use numeric values to represent non-numeric data in computing systems (e.g., binary, ASCII, RGB), and understand how such representations support storage, analysis, and communication of information.	DA.1.b.m.1: Represent and interpret data using standard and student-created encoding systems (e.g., binary, Unicode, Morse code), recognizing how encoding supports communication and data processing.	DA.1.b.h.1: Compare and convert common data formats and structures (e.g., tall vs. wide), selecting and justifying appropriate formats for different types of data analysis and computational processes.

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
	<p>DA.1.b.e.2: Identify different ways information can be shown using symbols or codes (e.g., numbers, letters, sounds, pictures).</p>	<p>DA.1.b.m.2: Explain and convert between basic number systems (such as binary and decimal) and describe how digital systems use these to represent various types of data (e.g., text, images, sound).</p>	<p>DA.1.b.h.2: Convert between binary, decimal, and hexadecimal number systems and explain how sequences of data can represent different forms (e.g., numbers, text, sound, images, instructions) in digital systems.</p>
<p>DA.1.c: Evaluate data characteristics and quality for effective analysis.</p>	<p>DA.1.c.e.1: Compare different types of data (numeric and non-numeric) and explain how their collection methods affect the kinds of information they provide.</p>	<p>DA.1.c.m.1: Evaluate how precision and granularity (e.g., rounding, sampling rates, image resolution) affect data accuracy, storage, and interpretation in computational analysis.</p>	<p>DA.1.c.h.1: Differentiate data types (nominal, ordinal, discrete, continuous), give examples of their use, and explain how their characteristics guide data cleaning and error detection.</p>

Standard: DA.2: Data Processing

Students will organize, transform, and analyze data with computational tools.

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
DA.2.a: Use computational tools to organize, transform, and apply data.	DA.2.a.e.1: Organize data into structured formats such as tables with records and attributes.	DA.2.a.m.1: Use computational tools (e.g., spreadsheets, apps) to gather, organize, sort, filter, group, and aggregate quantitative data, including calculating summary statistics (e.g., totals, averages, frequencies, measures of center and spread).	DA.2.a.h.1: Use computational tools to transform, clean, and organize data addressing issues like missing, out-of-range, or anomalous values; and evaluate how these approaches impact data analysis and communication of findings.
	DA.2.a.e.2: Organize and represent data using basic visual formats (e.g., picture graphs, bar graphs, tables) to help answer questions or identify simple patterns.	DA.2.a.m.2: Manipulate and transform data to support analysis by creating new variables or attributes, selecting appropriate data, and generating visualizations (e.g., charts, graphs) to answer specific questions or investigate patterns.	DA.2.a.h.2: Restructure data using computational methods (e.g., converting between tall and wide formats, flattening hierarchical data) to support specific types of analysis.
	DA.2.a.e.3: Identify different types of digital information (e.g., pictures, text, music) and match them with how they are stored or displayed.	DA.2.a.m.3: Compare data storage formats (e.g., image, text, and music files) and explain tradeoffs between file size, resolution, and quality in terms of computational efficiency and use.	DA.2.a.h.3: Analyze and optimize data storage and compression techniques for various digital formats, evaluating tradeoffs among file size, quality, resolution, and computational efficiency in real-world applications.

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
DA.2.b: Evaluate and ensure data quality for accurate analysis.	DA.2.b.e.1: Identify when data is missing or incorrect in a simple dataset (e.g., a number in a list of names) and discuss how it might affect answers to a question.	DA.2.b.m.1: Apply appropriate strategies to address missing, out-of-range, or anomalous data (e.g., using imputation, removing outliers, or marking invalid entries), and explain the impact on data quality.	DA.2.b.h.1: Assess data quality by checking for logical consistency, verifying expected data types and ranges, and determining if the dataset is appropriate and reliable for the intended analysis.

Standard: DA.3: Data Investigation

Students will explore, analyze, and interpret data to discover patterns, make inferences, and support conclusions.

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
DA.3.a: Develop data questions and gather or evaluate relevant data.	DA.3.a.e.1: Ask and investigate questions that can be answered with data and distinguish these from questions that require other methods to answer.	DA.3.a.m.1: Pose data-driven questions that anticipate patterns or variability and guide purposeful investigations.	DA.3.a.h.1: Formulate data questions involving multiple variables and refine them based on available or computationally collected data to support analysis, classification, or prediction.
	DA.3.a.e.2: Use data collected from everyday environments to support investigations.	DA.3.a.m.2: Collect, organize, and analyze data using computational tools (e.g., spreadsheets, databases) to identify relationships, classify information, and make predictions or informed decisions.	DA.3.a.h.2: Design and apply appropriate computational data collection processes (e.g., surveys, sensors, mobile GPS, open datasets) for answering questions, solving problems, or informing simulations.

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
	<p>DA.3.a.e.3: Identify and describe different types of information (e.g., pictures, charts, text) and ask questions about whether the information helps answer a specific question.</p>	<p>DA.3.a.m.3: Evaluate datasets and digital information sources by analyzing their reliability, accuracy, bias, format, and relevance for answering specific data questions.</p>	<p>DA.3.a.h.3: Assess and synthesize multiple datasets and digital sources, identifying limitations and drawing evidence-based conclusions while considering ethical implications, data provenance, and potential misrepresentation of data.</p>
<p>DA.3.b: Analyze data to identify patterns, relationships, and draw inferences.</p>	<p>DA.3.b.e.1: Sort, classify, and group data based on attributes (e.g., color, size, shape), and recognize patterns that people and machines can use to make decisions or predictions.</p>	<p>DA.3.b.m.1: Analyze data using computational tools (e.g., spreadsheets, databases) to identify relationships, classify information, and make predictions or informed decisions.</p>	<p>DA.3.b.h.1: Create and use computational models or simulations (e.g., epidemics, ecosystems, spread of information) to interpret data, test hypotheses, and evaluate outcomes, exploring how AI uses data patterns and how data properties influence results.</p>
<p>DA.3.c: Interpret and communicate data-driven conclusions.</p>	<p>DA.3.c.e.1: Represent data using visual tools (e.g., charts, graphs), and communicate insights or predictions based on identified trends or relationships.</p>	<p>DA.3.c.m.1: Create and refine data visualizations to clearly and accurately communicate findings, considering how design choices affect clarity, accessibility, and interpretation.</p>	<p>DA.3.c.h.1: Evaluate and create data visualizations with attention to clarity, accessibility, accuracy, and visual integrity to effectively communicate patterns, trends, and insights.</p>

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
	DA.3.c.e.2: Compare and evaluate different representations of the same data for clarity, accuracy, and accessibility. Create visualizations and brief explanations that highlight patterns or support claims.	DA.3.c.m.2: Communicate the results of a data investigation by explaining the question, data collection methods, analysis process, and evidence that supports conclusions.	DA.3.c.h.2: Construct formal data reports that justify conclusions using evidence, models, and visualizations for acknowledging limitations, potential biases, and tradeoffs (e.g., data representation, compression, or storage formats).

Standard: DA.4: Impacts of Data Science

Students will explore the impact of data science on society, including its ethical considerations, privacy concerns, and influence on decision-making.

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
DA.4.a: Explore how data impacts people and privacy.	DA.4.a.e.1: Ask data-driven questions to understand and address the needs of people in everyday life and community settings.	DA.4.a.m.1: Explain the benefits and risks of collecting personal data and incorporating it into datasets, including impacts on privacy and individual rights.	DA.4.a.h.1: Evaluate the societal and environmental consequences of large-scale data collection, storage, and processing, considering issues such as energy consumption, data ownership, and digital equity.
	DA.4.a.e.2: Identify benefits, risks, and basic privacy concerns related to data and AI, specifically in	DA.4.a.m.2: Explain how automated decisions and AI influence individuals and society by exploring these effects	DA.4.a.h.2: Evaluate how data-driven technologies, including Artificial Intelligence and Machine Learning, influence

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
	the context of data collection and sharing.	through simple computational models.	society examining impacts such as privacy, surveillance, misinformation, disinformation, and automated decision-making.
DA.4.b: Analyze bias and fairness in data systems.	DA.4.b.e.1: Identify different ways to collect data, recognize approaches that may lead to poor or biased information, and design collection methods that are inclusive and respectful of diverse backgrounds.	DA.4.b.m.1: Explain how decisions made during data collection, processing, analysis, and presentation can introduce bias, lead to misleading conclusions, and affect the fairness of AI models.	DA.4.b.h.1: Analyze how biased or incomplete training data in Artificial Intelligence/ Machine Learning systems can produce harmful or unfair outcomes, reinforcing stereotypes and social inequalities.
	DA.4.b.e.2: Collect, discuss, or interpret data, and communicate findings in a way that considers different perspectives.	DA.4.b.m.2: Evaluate the credibility of data from AI and machine learning sources by assessing the source, accuracy, potential biases, and relevance of the information.	DA.4.b.h.2: Evaluate data from AI and machine learning systems by analyzing algorithmic biases, data provenance, model limitations, and ethical implications to determine the validity and impact of information.
DA.4.c: Apply ethical principles and policies for data and AI use.	DA.4.c.e.1: Identify ways data and technology relate to privacy and fairness and recognize the importance of being responsible with information.	DA.4.c.m.1: Explain how data-driven algorithms and AI systems impact society, including issues of privacy, fairness, and ethics.	DA.4.c.h.1: Compare and contrast laws, policies, and frameworks that aim to ensure ethical and responsible data use, including regulations related to consent, data privacy, fairness, and transparency.

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
	DA.4.c.e.2: Use simple data investigations to explore real-world problems and share observations.	DA.4.c.m.2: Use data investigations and basic models to propose solutions to real-world problems, considering potential benefits and ethical impacts.	DA.4.c.h.2: Develop and justify actionable solutions to real-world problems using data investigations and modeling techniques, clearly communicating both benefits and potential ethical concerns.

Computing Systems and Security (CSS)

Standard: CSS.1: Hardware and Software: *Students will learn how computers use both physical parts (hardware) and programs (software) to work with and share information in digital form.*

Performance Indicators (By Grade Band)

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
CSS.1.a: Communicate clearly about hardware and software to help all learners make sense of computing systems.	CSS.1.a.e.1: Identify and explain the function of basic hardware components in a computing system (such as the display, system unit, keyboard, and other input/output devices).	CSS.1.a.m.1: Explain how computers organize and store information, including how files are named, grouped, and protected with settings that control who can use them.	CSS.1.a.h.1: Explain digital tools are used in areas like healthcare, industry, and robotics to solve problems and support important work.
	CSS.1.a.e.2: Describe how the physical parts (hardware) and the programs (software) work together to take in (input) and show/send out (output) information.	CSS.1.a.m.2: Describe how different types of computing devices (such as smartphones, tablets, or cloud-based systems) connect and share information through networks.	CSS.1.a.h.2: Explain how operating systems manage hardware and software, including memory, storage, and connected devices.
	CSS.1.a.e.3: Observe and explain how sensors are used in everyday places (like automatic doors or motion-activated lights).	CSS.1.a.m.3: Explain how software and hardware interact to perform tasks, including how updates or settings affect device performance and user experience.	CSS.1.a.h.3: Evaluate risks and protections related to computing system security, including how users, organizations, and designers manage vulnerabilities, data, and device access.

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
<p>CSS.1.b: Use hardware and software in safe and helpful ways to find solutions and complete tasks.</p>	<p>CSS.1.b.e.1: Use technology tools (tablets, computers, or apps) in safe and appropriate ways to help complete tasks or solve problems.</p>	<p>CSS.1.b.m.1: Use instructions, help guides, or online resources to learn how to complete computer tasks or solve problems.</p>	<p>CSS.1.b.h.1: Create and use specific factors to evaluate a computer system based on the task it needs to perform, such as running a game, browsing the web, or doing graphic design, considering processing power, storage, battery life, cost, and software compatibility.</p>
	<p>CSS.1.b.e.2: Talk about different types of apps or tools students use—like those that need the internet and those that work on the device—and explain what each one helps them do.</p>	<p>CSS.1.b.m.2: Explain the differences between apps that run on the internet and ones installed on a device, and describe when and why someone might use one over the other.</p>	<p>CSS.1.b.h.2: Compare different software solutions or platforms for a task, and justify a selection based on factors like user needs, accessibility, collaboration features, cost, and security.</p>
<p>CSS.1.c: Identify and fix hardware and software issues.</p>	<p>CSS.1.c.e.1: Follow simple steps to fix common computer problems (like no sound or the device won't turn on), by using solutions like checking if it's plugged in or restarting.</p>	<p>CSS.1.c.m.1: Use basic troubleshooting methods to identify and fix issues such as a frozen screen or apps that won't open.</p>	<p>CSS.1.c.h.1: Diagnose and resolve more complex computer and software problems by following a structured process - for example, freeing up storage, checking background programs, running security scans, or analyzing error logs.</p> <p>CSS.1.c.h.2: Create or modify troubleshooting guides to help others identify and fix hardware or software issues.</p>

Standard: CSS.2: Networks: *Students will learn how digital information travels through the Internet.*

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
CSS.2.a: Understand how physical components like cables, routers, and servers help computing devices connect and share information across the Internet.	CSS.2.a.e.1: Describe how computers and other devices connect to the Internet, using cables (Ethernet) or wireless signals (Wi-Fi).	CSS.2.a.m.1: Compare the strengths and challenges of wired vs. wireless networks in places like homes or schools.	CSS.2.a.h.1: Create a visual model (diagram) showing how hardware, software, and networks work together to complete a task or solve a problem.
	CSS.2.a.e.2: Give examples of how people all over the world use the Internet to stay connected with others through messages, video calling, or finding information.	CSS.2.a.m.2: Describe how the parts of the Internet work (servers, routers, and cables) keep it working reliably and fix problems.	CSS.2.a.h.2: Explore how networks are set up (routers, cables, and switches) and investigate ways to make them work better and faster.
CSS.2.b: Learn how the Internet uses rules (protocols) and software tools to move digital information between devices so people can connect, communicate, and share data.	CSS.2.b.e.1: Describe how website addresses (URLs) and email addresses allow individuals to connect and communicate over the Internet.	CSS.2.b.m.1: Show how the Internet sends data in packets - breaking it up, choosing different paths, and reassembling on the receiving end.	CSS.2.b.h.1: Explain how the Internet is made up of smaller networks. Compare it to other network types based on structure, performance, and scalability.

Standard CSS3: Security: *Students will understand and apply basic principles of Internet security.*

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
CSS.3.a: Recognize common online threats and explain how they can affect people, devices, and personal information.	CSS.3.a.e.1: Describe how sharing things online - like names, photos, or locations - can give personal information to people you didn't mean to share it with.	CSS.3.a.m.1: Identify and describe common types of cyber-attacks - like tricking people (social engineering) or installing harmful software (malware) and explain how they can affect users and devices.	CSS.3.a.h.1: Explain how cyber-attacks - such as phishing, malware, or ransomware - can harm individuals or organizations by stealing or damaging data.
		CSS.3.a.m.2: Explain how computers and data centers can be at risk from both natural events (storms, earthquakes) or intentional harm (hacking to steal data).	CSS.3.a.h.2: Analyze how cyber-attacks can affect companies and governments, causing problems like financial loss, data theft, and broken trust.
			CSS.3.a.h.3: Describe the steps organizations take to respond to security incidents, including detection, containment, recovery, and communication.
CSS.3.b: Apply safe practices to protect personal information, devices, and networks.	CSS.3.b.e.1: Describe ways to keep personal information safe - using strong passwords, logging out, and sharing with trusted sources.	CSS.3.b.m.1: Compare ways to protect computers and information—using both digital tools (like passwords, firewalls) and physical tools (like locks, cameras) and describing how protections impact ease of use.	CSS.3.b.h.1: Create a diagram of a computer system that includes security features (e.g., encryption, login systems), showing how the design reflects user's needs through testing and improvement.

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
	<p>CSS.3.b.e.2: Explain how passwords, account settings, and other tools help protect devices and share systems like school networks or public Wi Fi.</p>	<p>CSS.3.b.m.2: Describe how using multiple layers of protection (Defense in Depth) keeps systems safer and explain steps to take if a system is attacked.</p>	<p>CSS.3.b.h.2: Examine how networks can be at risk (vulnerabilities) and describe ways to reduce those risks (mitigation), such as firewalls, updates, or access controls.</p>
	<p>CSS.3.b.e.3: Show how information can be scrambled (encrypted) and unscrambled (decrypted) to stay safe, using drawings, activities, or digital tools.</p>	<p>CSS.2.b.m.3: Describe how encryption helps keep digital information private and secure and give examples of where it's used—like in messaging apps, websites, or online banking.</p>	<p>CSS.3.b.h.3: Demonstrate how encryption and decryption techniques protect digital information, and explain how these ensure confidentiality, integrity, and access control. Use simulations to model symmetric and asymmetric encryption.</p>
	<p>CSS.3.b.e.4: Explain why it is important to check and update passwords, software, and settings to protect personal and shared information.</p>	<p>CSS.2.b.m.4: Describe safe practices for accessing systems, including recognizing secure sites, managing passwords, and avoiding risky downloads or pop-ups.</p>	<p>CSS.3.b.h.4: Evaluate how security policies (like password rules, two-factor authentication, or acceptable use policies) balance protection and user experience in schools, companies, or public systems.</p>

Standard CSS.4: Impacts of Computing: *Students will understand how computing systems have created both helpful changes and challenges for people, communities, and the world.*

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
CSS.4.a: Understand how computing systems shape everyday experiences - helping people learn, connect, and solve problems, while also creating new challenges.	CSS.4.a.e.1: Identify the benefits of using computers and devices - like learning new things or staying in touch - and describe potential problems, such as too much screen time or sharing personal information.	CSS.4.m.1: Analyze how widely used technology helps individuals solve problems, (education, communication) while also creating challenges (cyberbullying, privacy issues, screen time).	CSS.4.h.1: Discuss how collecting and analyzing personal data affects individuals, including how businesses, governments, and platforms use data for ads, surveillance, or content recommendations.
	CSS.4.a.e.2: Give examples of ways people use the Internet with devices - like watching videos, playing games, learning, or talking with others.		
CSS.4.b: Examine how computing systems shape communities, influence society, and create global change.	CSS.4.b.e.1: Explore how using computers and the Internet can help the environment and also cause harm (electricity or electronic waste).	CSS.4.b.m.1: Investigate how access to computing systems varies by socioeconomic status, location, ability, and age - and analyze how these differences affect individuals and communities.	CSS.4.b.h.1: Investigate how computing systems and physical technologies (e.g. data centers, servers, mobile devices) affect society, the environment, and the economy - considering both benefits and harms.
	CSS.4.b.e.2: Work together to make technology easier for everyone to use, including people with different needs, abilities, and ways of thinking.	CSS.4.b.m.2: Design a user interface (e.g., web pages, app, animation) to be more inclusive and accessible, minimizing the impact of designer bias.	CSS.4.b.h.2: Discuss the benefits and challenges of global technology use, including how it affects culture, communication, and online interaction.

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
			<p>CSS.4.b.h.3: Evaluate how responsible and harmful uses of computing affect people and communities socially and economically. Use real-world examples to consider different perspectives and consequences.</p>
			<p>CSS.4.b.h.4: Examine the reasons behind rules and policies that guide how we use technology - such as the Children's Online Privacy Protection Act (COPPA) to protect kids' privacy, acceptable use policies for safe online behavior, and copyright laws to respect creators' rights.</p>

Computing and Society (FUT)

Standard: FUT.1: Innovations in Computing: *Students will analyze the historical development of computing technologies and their impact on society, from early mechanical devices to modern digital systems.*

Performance Indicators (By Grade Band)

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
FUT.1.a: Describe technological advances and changes in society that have influenced the field of computer science.	FUT.1.a.e.1: Describe how important events have inspired people to create computer science ideas and inventions.	FUT.1.a.m.1: Compare early and modern computing devices and describe how changes in society have influenced their development.	FUT.1.a.h.1: Compare and contrast early computing devices and how they influence emerging technologies.
FUT.1.b: Analyze societal and environmental impacts of historical computing technologies.	FUT.1.b.e.1: Examine how computing innovations have changed the way people live, work, or communicate over time.	FUT.1.b.m.1: Analyze intended and unintended impacts of historical computing technologies on society and the natural environment.	FUT.1.b.h.1: Compare different historical computing innovations and their effects on present-day society and the natural environment.
FUT.1.c: Explain the contributions of significant individuals and other stakeholders in computer science history.	FUT.1.c.e.1: Explore the contributions of people from different cultures, backgrounds, and time periods who helped shape computing technologies.	FUT.1.c.m.1: Compare roles of individuals, communities, organizations, and governments in advancing technology.	FUT.1.c.h.1: Evaluate how people (e.g. diverse and underrepresented groups), organizations, and governments have shaped computing innovations and their impact on society.
FUT.1.d: Evaluate the role of policy and ethics in historical development and present-day concerns related to computing.	FUT.1.d.e.1: Discuss why it is important to use computing tools fairly and respectfully, and that rules (policies) exist for their use.	FUT.1.d.m.1: Identify and discuss how historical computing technologies have raised ethical questions or unintended societal issues.	FUT.1.d.h.1: Evaluate policies and legislation designed to encourage ethical innovation and minimize societal risks associated with technology.

Standard: FUT.2: Emerging Technologies: *Students will identify, explain, and design solutions to problems that allow equitable access to new and emerging technologies.*

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
FUT.2.a: Test and refine digital artifacts that are created with both accessibility and ethics in mind.	FUT.2.a.e.1: Use online collaborative spaces ethically and safely to work with another student to solve a problem, seek out diverse perspectives, or improve a project.	FUT.2.a.m.1: Use the internet ethically and safely to critically evaluate and redesign a computational artifact to remove barriers to universal access. (e.g. using captions on images, high contrast colors, and/or larger font sizes)	FUT.2.a.h.1: Demonstrate how emerging technologies (e.g. artificial intelligence, virtual reality, automation) enable new forms of experience, expression, communication, and collaboration.
	FUT.2.a.e.2: Brainstorm ways in which computing devices could be made more accessible to all users.	FUT.2.a.m.2: Evaluate how design decisions in emerging technologies influence user experiences differently across different communities.	FUT.2.a.h.2: Design a user interface (e.g. web pages, mobile applications, animations) to be more inclusive and accessible, minimizing the impact of the designer's inherent bias.
FUT2.b: Understand the impact technology has on our everyday lives and the effects of computing on the economy and culture.	FUT.2.b.e.1: Identify everyday technologies and discuss how people use them to meet needs and how they affect daily life.	FUT.2.b.m.1: Provide examples of how computational innovations and devices impact health and wellbeing, both positively and negatively, locally and globally and create new solutions that depend on expertise from multiple fields (e.g. effects of globalization and automation).	FUT.2.b.h.1: Compare and debate the positive and negative impacts of computing on behavior and culture (e.g., smartphone usage, online reservation services, artificial intelligence, emerging technologies).

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
	FUT.2.b.e.2: Describe what technology can and cannot do and explore how new and emerging technologies can change the way people live, work, and communicate.	FUT.2.b.m.2: Contrast the features, functionality, and characteristics of emerging technologies with technologies that came before.	FUT.2.b.h.2: Develop criteria to evaluate the beneficial and harmful effects of computing innovations on people and society.
	FUT.2.b.e.3: Identify ways new technologies can help solve problems and justify why people choose to use or not use them.	FUT.2.b.m.3: Explain how emerging technologies create new solutions that depend on expertise from multiple fields	FUT.2.b.h.3: Describe how emerging technology could improve an existing solution by offering new approaches, features, or insights.
FUT.2.c: Understand the effects of the digital divide.	FUT.2.c.e.1: Brainstorm and advocate for ways in which computing devices and the Internet could be made more available to all people.	FUT.2.c.m.1: Explain the impact of the digital divide (e.g. uneven access to computing, computing education, and interfaces) on access to critical information.	FUT.2.c.h.1: Evaluate the impact of equity, access, and influence on the distribution of computing resources in a global society.
FUT.2.d: Understand intellectual property and fair use.	FUT.2.d.e.1: Understand that ideas, code, and digital creations belong to the people who made them, and describe why it is important to give credit.	FUT.2.d.m.1: Explain the role of licenses and permissions in using or sharing technology and digital content and compare how different types of intellectual property (like open source vs. proprietary) affect creators and users.	FUT.2.d.h.1: Evaluate scenarios involving copyright, patents, or open-source software to determine how intellectual property rights impact innovation, collaboration, and ethical computing.
FUT.2.e: Understand the effects of computing on communication and relationships.	FUT.2.e.e.1: Explain the differences between communicating electronically and communicating in person.	FUT.2.e.m.1: Analyze and present beneficial and harmful effects of personal electronic communication and social electronic communication.	FUT.2.e.h.1: Evaluate the negative impacts of electronic communication on personal relationships and evaluate differences between face-to-face and electronic communication.

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
	<p>FUT.2.e.e.2: Compare and contrast the effects of communicating electronically to communicating in person.</p>	<p>FUT.2.e.m.2: Describe ways in which the Internet impacts global communication and collaboration.</p>	<p>FUT.2.e.h.2: Create a list of practices that individuals and organizations can use to encourage proper use of both electronic and face-to-face communication.</p>
<p>FUT.2.f: Understand the fundamentals of artificial intelligence and its applications.</p>	<p>FUT.2.f.e.1: Explain how artificial intelligence systems learn from data and make decisions based on patterns.</p>	<p>FUT.2.f.m.1: Build and test a basic machine learning model and describe how the model makes predictions.</p>	<p>FUT.2.f.h.1: Analyze how artificial intelligence works, and evaluate its applications—including strengths, limitations, and ethical considerations—in real-world contexts.</p>
	<p>FUT2.f.e.2: Students can identify and describe examples of artificial intelligence in everyday life (e.g., voice assistants, recommendations).</p>	<p>FUT2.f.m.2: Students analyze how the choice of training data affects artificial intelligence outcomes and can recognize examples of biased outputs.</p>	<p>FUT2.f.h.2: Evaluate the societal impact of artificial intelligence usage. Students investigate and debate issues such as algorithmic bias, surveillance, intellectual property, and accountability.</p>

Standard: FUT.3: Career Exploration: *Students will explore, identify, and evaluate computer science careers and pathways, understand the roles and responsibilities within these careers, and reflect on the skills and education needed to pursue them.*

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
FUT.3.a: Identify and explore computer science occupations.	FUT.3.a.e.1: Describe how people use digital tools in different jobs and explain how personal interests can connect to computing in work or careers.	FUT.3.a.m.1: Compare how professionals in different careers use computational thinking to solve real-world problems and identify how your own interests and strengths might connect to those applications."	FUT.3.a.h.1: Evaluate how computing skills align with personal interests and career goals by investigating how professionals apply computer science in various roles and the impact of those careers.
	FUT.3.a.e.2: Describe how computing is used in a variety of industries and careers (e.g. healthcare, transportation, and entertainment) and how people learn new skills as technology changes.	FUT.3.a.m.2: Examine how changes in technology can create new jobs or change how people work.	FUT.3.a.h.2: Explore postsecondary education, industry certifications, and training pathways comparing how professionals apply computer science in their work.
FUT.3.b: Use strategies and software from industry to prepare for a career in computer science.	FUT.3.b.e.1: Investigate how professionals collaborate with computing technologies to solve problems creatively, accurately, and efficiently.	FUT.3.b.m.1: Engage in real-world computational thinking projects that reflect professional, diverse CS problem-solving approaches, and integrate skills with expertise from diverse fields.	FUT.3.b.h.1: Develop and showcase computational solutions that reflect real-world industry practices and collaborative problem-solving methods.
	FUT.3.b.e.2: Identify situations where technology choices can affect people differently and talk about what it means to be	FUT.3.b.m.2: Investigate how professionals in computing careers address ethical dilemmas.	FUT.3.b.h.2: Debate ethical challenges in emerging technologies and defend positions using

Learning Priority	PK-5 (e)	6-8 (m)	9-12 (h)
	responsible or fair when using technology.		professional codes of conduct and real-world examples.