CSTA

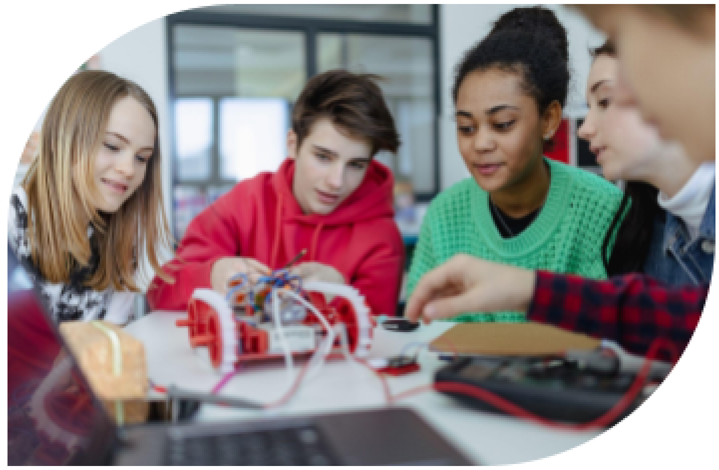
2026 CSTA PK–12

COMPUTER SCIENCE STANDARDS

Middle School Implementation Guidance

Introduction

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.



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.

Purpose of This Guidance

This guidance is designed for instructional leadership teams making decisions about how to implement middle school computer science in ways that are coherent, realistic, and aligned to the standards. The standards at the middle school level are all essential knowledge that builds on what is taught in elementary school and in the middle to high school computer science. Since middle schools have such a variety of course offerings and scheduling options, there is no single answer to how to divide up the standards in order to cover everything.

This document suggests multiple paradigms to accomplish this, including a single full-year course, a series of courses by topic area, and integration into other subject areas. Throughout planning, teams should draw on the implementation examples and interdisciplinary connections included with each standard, which illustrate what instruction can look like across a variety of classroom contexts without prescribing specific curriculum.

For easier viewing and more details, check out the [interactive web display](#).

Design Principles for Middle School

Principle	Description
Equitable Access	Middle school CS should not be reserved for a small group of students or treated as optional enrichment for those already interested in technology. Decisions should be made with participation in mind: who has access, who is being left out, and what policies and schedules are needed to ensure every student has meaningful opportunities to learn.
Coherent Learning	The standards form a progression across grades 6, 7, and 8. Middle school implementation should be planned as a coherent experience across the band, with attention to what students have ideally learned before middle school and what they should be prepared for in preparation for high school. Even when schools organize instruction differently, learning should build intentionally over time, connecting each grade to the next rather than repeating disconnected introductory experiences.
Flexibility	Strong implementation can look different across schools and districts. This document does not present a single required model; it offers implementation models schools can combine to fit their own context while maintaining the integrity of the standards.
Alignment to Intent	Implementation should reflect the full intent of the standards and the breadth of computer science as a discipline. In middle school, students should encounter a broad and developmentally appropriate computer science experience that goes beyond coding or general technology use, reflecting both foundational concepts and the ways those concepts connect to people, communities, and real-world problems.
Evidence-Based Pedagogy	Research-based, inclusive pedagogy is fundamental to a classroom where every student feels valued and supported. Teachers should build understanding before independent creation, make abstract ideas concrete through hands-on tasks, and treat debugging and revision as normal parts of learning. Instruction should reduce barriers using the Universal Design for Learning Guidelines for Computer Science Education, with practical strategies available through the Raspberry Pi Foundation's Pedagogy Quick Reads.
Attention to Local Capacity	Implementation decisions should be shaped by who will teach the content, when students will experience it, and what support educators will need to teach it well. Middle school CS implementation is not only a curriculum question; it is also a question of staffing models, instructional time, PD, and the systems needed to support consistent, high-quality learning opportunities for students.

Key Decisions Schools Need to Make

Successful implementation requires intentional planning about how CS will function within the middle school program as a whole. Some schools already offer standalone CS courses, others integrate computing across subjects, and many have hybrid or exploratory models that have evolved. The questions below help schools assess current practice and identify the adjustments needed for alignment, coherence, and equitable access under the updated middle school standards. Schools may find it useful to revisit these questions as implementation evolves.

1. **How do students currently experience computer science across grades 6–8?**

   Before making changes, schools should examine:
   - Where students already encounter computer science learning
   - Whether access is consistent across grade levels and student groups
   - Which experiences are required versus optional

2. **How are computer science standards organized instructionally? Or, how will they be?**

   Schools should consider:
   - Which current courses or units already address portions of the standards
   - Where alignment is strong and can be built upon
   - Where revisions or redistribution of standards may be needed

   Transitioning to new standards often involves refining goals and sequencing, not eliminating effective instruction. To organize instruction, schools can draw on the three pathways outlined in this document, and many will use elements of more than one.

3. **Who is responsible for computer science instruction? Who will be?**

   Different implementation models distribute responsibility differently. Schools should identify:
   - Which educators are responsible for direct CS instruction
   - How responsibilities are shared in integrated settings

   Clear role definition supports instructional quality and consistency across grades and departments, especially in integrated models.

4. **How will coherence and progression be maintained?**

   Schools should plan for:
   - Intentional sequencing of standards and practices
   - Shared expectations for depth and complexity
   - Coordination across grade levels and subject areas

The following reflection framework is designed to help leadership teams take stock of what is currently happening in grades 6–8 and identify the most important shifts needed for standards-aligned implementation. Teams can complete it collaboratively, revisiting it as scheduling, staffing, and curriculum decisions evolve.

Current State → Future State Reflection

Reflection Question	Current State	Where We Want to Go
Where do students encounter CS?		
Who has access?		
Where do concepts repeat without growth?		
Where are standards not yet addressed?		

Once you have considered the framework above, use the matrix below to compare the three implementation models across key decision areas. The models describe ways to organize instruction, not mandates to replace existing courses, and many schools will combine elements of more than one. The matrix can support clearer conversations about trade-offs and help schools select an approach suited to their context.

Key Decisions × Implementation Models Matrix

Decision Area	Model 1: Grade-Level Progressions	Model 2: Topic-Based Clusters	Model 3: Interdisciplinary Integration
Student Exposure	Every year, all students	Varies by course	Depends on subject integration
Structure	Distributed across grades	Organized by theme	Embedded in content areas
Staffing	CS educators across grades	Specialized by topic	Content teachers + support
Coherence	Vertical planning	Course sequencing	Cross-department coordination
Best Fit	Schools prioritizing continuity	Schools with electives (often rotating)	Schools with schedule constraints

Each of these models is explored in depth in the sections that follow, including sample course sequences and planning considerations.

Model 1: Grade-Level Progressions

This model organizes middle school CS as a **coherent three-year experience across grades 6–8.** Since the middle school standards are a grade band, schools must decide how to distribute them across the three years. Not every concept needs to advance the same way or at the same pace; the goal is one workable progression that preserves the standards' intent while ensuring students encounter the full breadth of foundational middle school CS.

Standards are distributed across the three grades based on conceptual dependencies, instructional readiness, and the need to build toward greater independence and complexity:

Progression	Grade 6	Grade 7	Grade 8
Focus	Foundations	Analysis	Evaluation
What It Looks Like	Emphasizes foundational structures and broad orientation across the five concepts.	Shifts toward analysis, verification, interpretation, and investigation.	Culminates in evaluating, critiquing, and refining computational approaches, data, systems, and societal impacts.

Is This Model Right for Your School?

This model fits schools that want all students to experience computer science every year of middle school and that have the instructional time, staffing, and coordination to support a cross-grade model.

- **Main strength – coherence**: Students build knowledge and practices through repeated opportunities over time.
- **Main challenge – coordination:** Standards should carry across grades without repetition or gaps.

Schools using this model should treat the progression as a shared 6-8 responsibility rather than as three separate grade-level plans.

Planning Resources

Two resources support planning in this model, together informing decisions about scope and sequence, pacing, curriculum design, and teacher planning:

- **Progression Across Grades table** – The following below shows the proposed grade-level distribution of standards.
- **Progression Within Grades infographic** – Identifies an instructional flow within each grade, from where learning should begin to where it should build by the end of the year.

Progressions Across Grades

Concept	Grade 6	Grade 7	Grade 8
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 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-05:** Use an AI tool to generate outputs that assist in solving a computational problem.	**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-ML-06: Hypothesize how a machine learning model generates classifications 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.
	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 harmful 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-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-PD-13:** Use reference documentation in program development.	
	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 computing technology based on user feedback to improve its usability and accessibility.

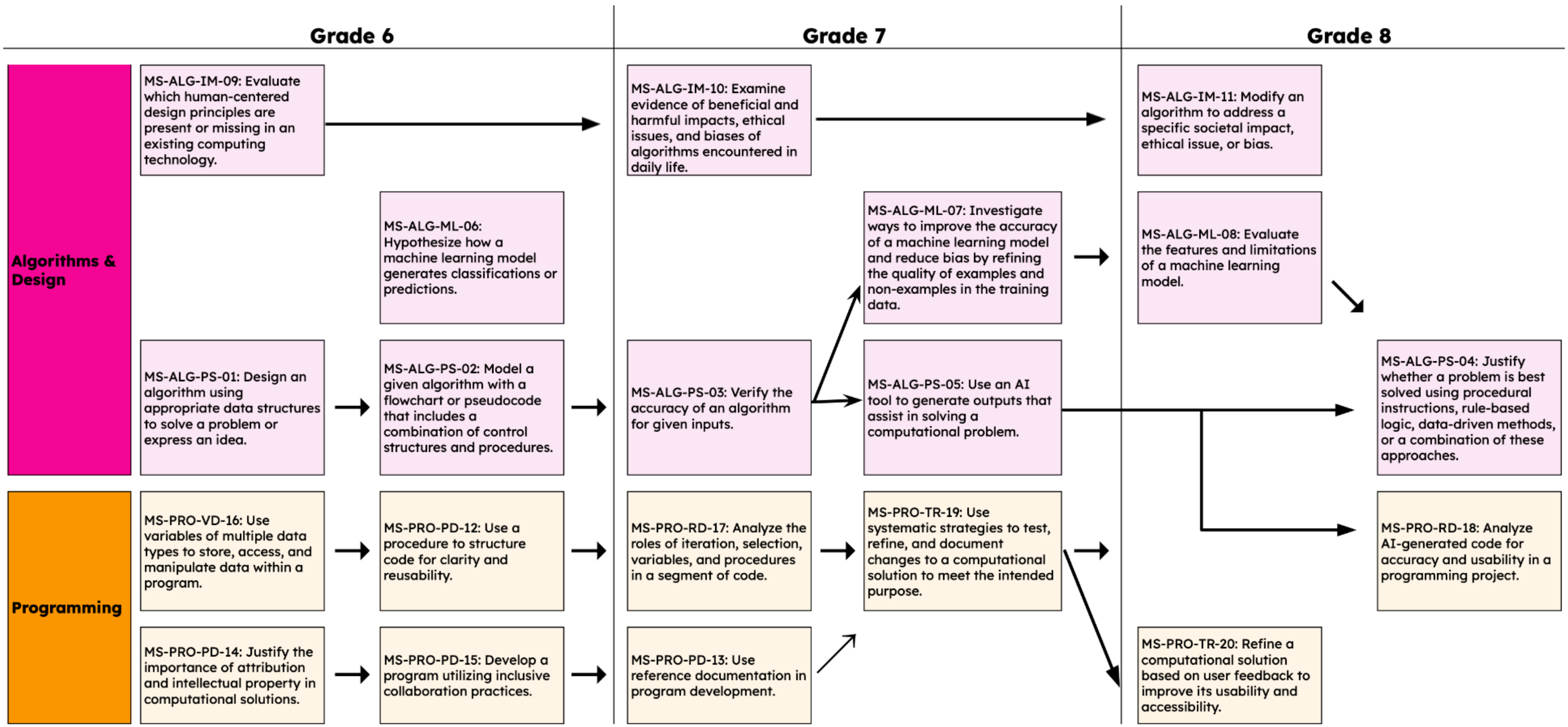
Progressions Across Grades

Concept	Grade 6	Grade 7	Grade 8
Data & 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, filter, group, and summarize structured data.	**MS-DAT-DC-24:** Analyze options to address a data quality issue.	**MS-DAT-DC-21:** Evaluate how different levels of precision and granularity in data collection affect accuracy, storage, and analysis.
		MS-DAT-DI-26: Create data visualizations to show how different design choices can impact the interpretation of the same data.	**MS-DAT-DI-25:** Use a computational tool to identify relationships among variables in a dataset and make a classification or prediction. **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 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.
Systems & Security	**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-HW-30:** Examine differences between computing systems based on user needs, system requirements, and potential societal, environmental, and ethical impacts.
		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.	

Progressions Across Grades

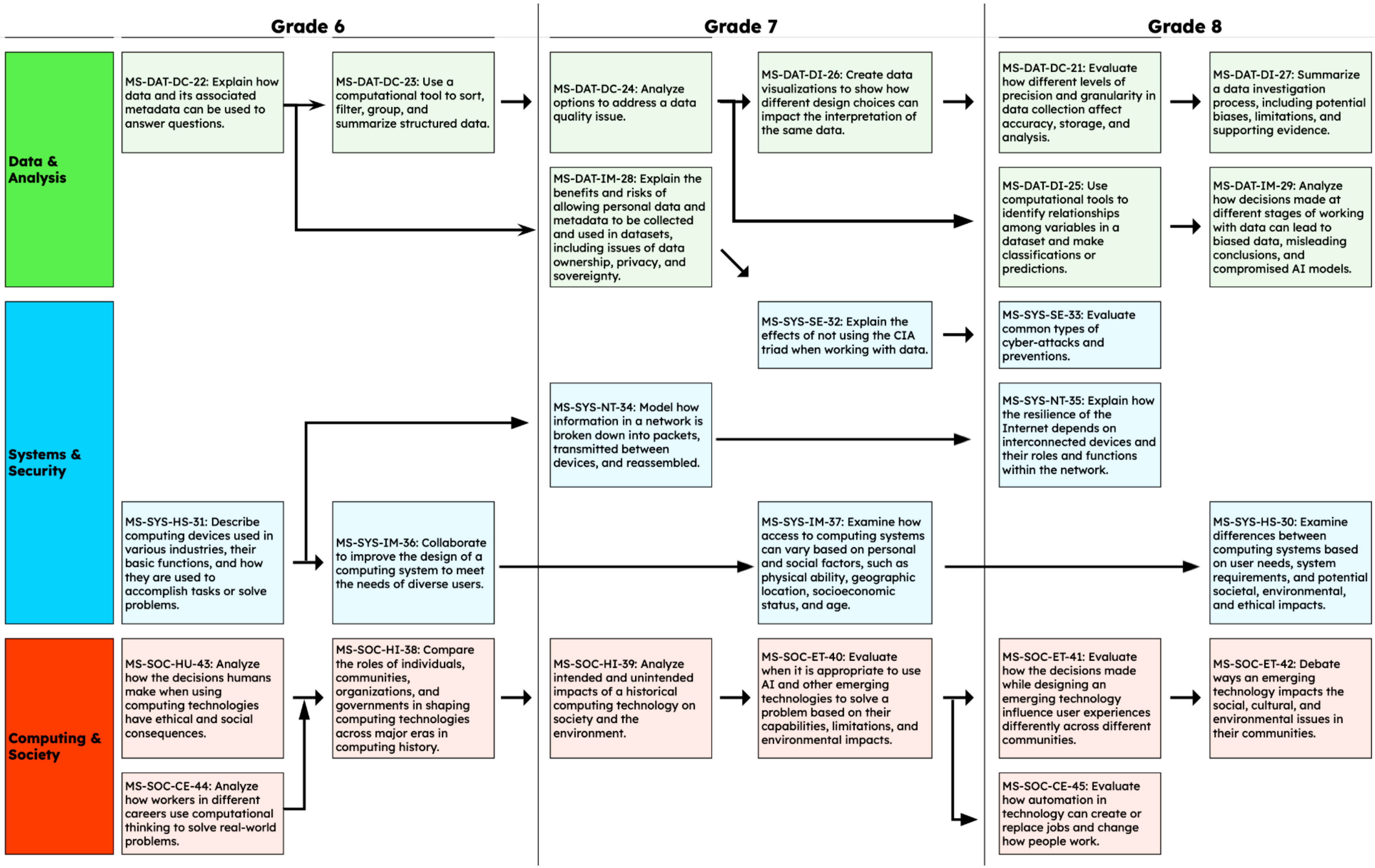
Concept	Grade 6	Grade 7	Grade 8
Computing & Society	**MS-SOC-HI-38**: Compare the roles of individuals, communities, organizations, and governments in shaping computing technologies across major eras in computing history.	**MS-SOC-HI-39:** Analyze intended and unintended impacts of a historical computing 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 influence 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.		

Progressions Within Grades



[View this Progressions Within Grades infographic as a spreadsheet.](#)

Progressions Within Grades



Model 2: Topic-Based Clusters

Many middle schools weave arts and technology classes into a rotating schedule by quarter, trimester, or semester. The topic-based cluster model organizes computer science as a set of hands-on, high-interest topic-based courses or unit clusters spread over a single year or multiple years. The four units outlined here provide robust coverage of the standards:

- Game Development
- Physical Computing / Robotics
- Artificial Intelligence
- Cybersecurity

These units are examples used to demonstrate the breadth of computational concepts and projects that can be offered in middle school. Appendix B includes a learning progression for each unit, mapping standards to learning targets and their application in the unit context.

Is This Model Right for Your School?

This model fits schools with rotation-based schedules already in place, or that want to build CS participation through high-interest, project-driven units staffed by teachers with topic-specific strengths, provided unit selection is treated as a coverage decision, and schools verify what each student actually experiences across grades 6–8.

- **Main strength – engagement and schedule fit:** Hands-on, themed units connect to student interests and slot into existing quarter, trimester, or semester rotations without restructuring the master schedule.
- **Main challenge – coverage:** Topic-based units do not automatically address the full breadth of the standards, so schools must deliberately map units to standards and confirm that every student's combination of experiences covers the band.

Planning Resources

Four resources support planning in this model: one learning progression per unit, each mapping the unit's standards to a learning target and its application in that unit's context. Each table follows below:

- **Game Design Unit Learning Progression:** sequences the unit's standards from algorithm design through testing and collaborative programming.
- **Robotics / Physical Computing Unit Learning Progression:** sequences the unit's standards from understanding devices through inclusive design.
- **AI Unit Learning Progression:** sequences the unit's standards from foundational AI concepts through training, refining, and critically evaluating AI tools.
- **Cybersecurity Unit Learning Progression:** sequences the unit's standards from foundational concepts through threats, protections, and societal impact.

Game Design

Game design is one of the most reliable entry points into programming for middle school students. Building a game motivates students to learn algorithms, variables, conditionals, and procedures because the concepts are immediately visible in something they want to make work.

In a 4–6 week unit, students design, build, and playtest an original game. They begin by brainstorming game mechanics and writing algorithms. Using a block-based or text-based coding language, students implement variables, conditionals, and procedures to control scoring, lives, and levels. They test their games, gather peer feedback, and refine code for clarity and accessibility. The unit culminates in a showcase where students document their design process, justify their use of specific algorithms, and reflect on how their game's rules create an engaging user experience.

A comprehensive introduction to game design at the middle school level should include:

1. **Algorithms and Logic:** Building game features, including rules, scoring, and win conditions.
2. **Collaborative Programming:** Writing clean and reusable code and working effectively in teams.
3. **Testing and Iteration:** Using structured testing and improving the game based on user feedback.
4. **Inclusive Design:** Designing the game to meet the needs of the player and making adjustments to ensure it is accessible to all players.

Alternative Unit Designs with a Programming Focus

Alternative Unit	Description
App Development	Students design a functional mobile app that stores user data using variables of multiple data types. They create flowcharts for app navigation, use procedures to handle repeated user interface events, and test systematically with user feedback to improve accessibility. They also analyze how app algorithms might introduce bias and modify their design to be more inclusive.
Web Design	Students build an interactive website using variables, conditionals, and functions to respond to user clicks or form inputs. They structure code with reusable procedures, document borrowed assets with proper attribution, and refine the site based on peer usability testing. They also examine how web tracking raises ethical issues around data privacy and modify their site to reduce unnecessary data collection.
Visual Programming	Students create an animated story or interactive simulation using visual blocks, designing algorithms with loops, conditionals, and variables to control character movement and scoring. They collaborate in pairs using inclusive practices, organize scripts into custom blocks, and test edge cases systematically. They modify their project to remove unintended bias and add accessibility features like slower animation speeds.

Game Design Unit Learning Progression

Phase of Learning	Identifier	Standard	Application in Game Design
1. Algorithms and Logic	MS-ALG-PS-01	Design an algorithm that includes variables of multiple data types to solve a problem or express an idea.	Design a game algorithm that uses at least two data types to track game progress and express win/loss conditions.
	MS-ALG-PS-02	Model a given algorithm with a flowchart or pseudocode that includes a combination of control structures and procedures.	Create a flowchart or pseudocode for their game's program that includes loops, conditionals, and at least one procedure.
	MS-ALG-PS-03	Verify the accuracy of an algorithm for given inputs.	Manually trace through their game algorithm with specific test inputs to verify that the output matches expected behavior.
	MS-PRO-VD-16	Use variables of multiple data types to store, access, and manipulate data within a program.	Create and modify variables for score, health, level number, and player name in their game, reading and updating these variables during events like collecting power-ups.
2. Collaborative Programming	MS-PRO-PD-12	Use a procedure to structure code for clarity and reusability.	Restructure repetitive game code into reusable procedures and explain how this improves code clarity.
	MS-PRO-PD-13	Use reference documentation in program development.	Consult online or teacher-provided documentation to implement a new game feature like a timer or random spawning.
	MS-PRO-PD-14	Justify the importance of attribution and intellectual property when developing computing technologies.	Add comments or a credits screen to their game that attributes any borrowed assets like sounds or sprites and explain why this matters.
	MS-PRO-PD-15*	Develop a program utilizing inclusive collaboration practices.	Work in a pair or small group using pair programming and inclusive feedback to build one shared game project.

Phase of Learning	Identifier	Standard	Application in Game Design
3. Testing and Iteration	MS-PRO-RD-17	Analyze the roles of iteration, selection, variables, and procedures in a segment of code.	Examine a peer's game code and correctly identify one example of iteration, one selection, one variable, and one procedure, explaining each role.
	MS-PRO-TR-19*	Use systematic strategies to test, refine, and document changes to a computing technology to meet the intended purpose.	Maintain a simple test log, running at least 5 test cases on their game (including edge cases like negative score), documenting bugs and fixes before final submission.
	MS-PRO-TR-20*	Refine a computing technology based on user feedback to improve its usability and accessibility.	Collect feedback from two playtesters using a structured form and make at least two usability improvements.
	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.	Analyze how choosing to collect only "win rate" data in their game and ignoring player frustration or quit rate could lead to misleading conclusions about whether the game is fun.

** Inclusive design standards are integrated within phases 2 and 3 rather than treated as a separate phase.*

Physical Computing / Robotics

Physical computing makes abstract concepts tangible by connecting code to the physical world, rewarding the persistence and teamwork that larger projects demand.

In a 4–6 week unit, students design, build, test, and improve projects combining hardware (sensors, motors, microprocessors, circuit boards) with software, using a block-based or text-based coding language to control motion, sound, lights, or data. They test their projects, gather peer feedback, and refine their code for clarity and accessibility, documenting their design process and reflecting on how the project meets requirements and delivers value to the user.

A comprehensive introduction to physical computing at the middle school level should include:

1. **Understanding Physical Digital Devices:** Learning how various input and output devices work and how to program them to act and collect data.
2. **Algorithms, Programming, and Logic:** Learning to control the actions and reactions of various devices.
3. **Testing and Iteration:** Using structured testing and improving the device based on performance and user feedback.
4. **Collaborative Design:** Larger projects need teams, whether in a class, a club, or the real world, and limited space and equipment make collaboration essential.
5. **Inclusive Design:** Designing physical projects without bias, designing devices to meet the needs of the user, and making adjustments to ensure the device is accessible to all users.

Alternative Unit Designs with a Physical Computing Focus

Alternative Unit	Description
AI and Robotics	Students program robots to complete tasks using features like voice, facial, color, or object recognition for decision-making, and may use AI tools to generate, test, and debug code. The unit should include a discussion of bias and accessibility issues that AI-enabled detection can improve or worsen. Students can research the latest innovations in AI, machine learning, and robotics.
Microprocessors	Students are introduced to computing hardware, electronics, and coding through hands-on projects, using microprocessors with sensors and basic circuits to build interactive devices like robots, environmental monitors, smart devices, or game controllers.
E-Textiles or Paper Circuits	Students combine computer science with textile or paper arts, creating interactive, wearable, or decorative electronics by embedding sensors, microcontrollers, and light-up components into fabric or paper using conductive materials, and learning programming concepts like loops and variables through a creative, tangible project.

Physical Computing / Robotics Unit Learning Progression

Phase of Learning	Identifier	Standard	Application in Robotics / Physical Computing
1. Understanding Physical Digital Devices	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.	Work with sensors (input) and actuators (output) and explain how robots interact with the physical world.
	MS-PRO-VD-16	Use variables of multiple data types to store, access, and manipulate data within a program.	Store sensor readings (distance, light, color, etc.) to control robot behavior.
2. Algorithms, Programming, and Logic	MS-ALG-PS-01	Design an algorithm that includes variables of multiple data types to solve a problem or express an idea.	Define rules for robot behavior (if sensor detects X → robot does Y).
	MS-ALG-PS-02	Model a given algorithm with a flowchart or pseudocode that includes a combination of control structures and procedures.	Plan robot logic before coding (input → decision → action).
	MS-PRO-PD-12	Use a procedure to structure code for clarity and reusability.	Create reusable code blocks or functions for robot behaviors.
	MS-PRO-PD-13	Use reference documentation in program development.	Use guides to correctly program sensors, motors, and components.
	MS-PRO-RD-17	Analyze the roles of iteration, selection, variables, and procedures in a segment of code.	Explain how the robot continuously reads input and responds in real time.
	MS-ALG-PS-03	Verify the accuracy of an algorithm for given inputs.	Test specific scenarios to confirm correct robot behavior.

Phase of Learning	Identifier	Standard	Application in Robotics / Physical Computing
3. Testing and Iteration	MS-PRO-TR-19	Use systematic strategies to test, refine, and document changes to a computing technology to meet the intended purpose.	Troubleshoot code, wiring, and logic when the robot doesn't behave as expected.
	MS-PRO-TR-20	Refine a computing technology based on user feedback to improve its usability and accessibility.	Refine the robot after testing and peer feedback.
4. Collaborative Design	MS-PRO-PD-15	Develop a program utilizing inclusive collaboration practices.	Work in teams, share roles, and contribute to a shared build.
	MS-PRO-PD-14	Justify the importance of attribution and intellectual property when developing computing technologies.	Credit sources and reflect on the design process.
5. Inclusive Design	MS-SYS-IM-36	Collaborate to improve the design of a computing system to meet the needs of diverse users.	Adjust design for usability, including clear signals, responsiveness, and accessibility.

Artificial Intelligence Unit

Teaching about artificial intelligence and machine learning is no longer separate from teaching computer science. AI is integrated throughout the standards because of its impact on the way we interact with computers and program them.

In a 4–6 week unit, students investigate how AI systems work, where they appear in daily life, and how they are built. They begin by identifying AI in familiar tools and classifying the types of problems it solves. Students then train and test simple machine learning models using age-appropriate platforms, examining how training data shapes model behavior and where bias can enter. They critique AI-generated content for quality and accuracy, and evaluate the trade-offs of using AI in a real-world scenario. The unit culminates in a presentation where students recommend whether and how AI should be used for a specific task, supporting their position with evidence from their own model testing.

A comprehensive introduction to AI at the middle school level should include:

1. **Introduction to AI and Machine Learning:** Recognizing types of AI and common applications, identifying when AI is being used, and critically evaluating what it produces.
2. **Training and Refining AI Models:** Analyzing, improving, and critiquing AI-generated content and machine learning models.
3. **Critical Evaluation of AI Tools:** Assessing both the quality of what AI tools produce and the ramifications of using AI as part of a solution.
4. **Impacts of AI on People and Society:** Examining changes in careers, education, data privacy, creative expression, and social interaction, for good and bad.

Alternative Unit Designs with an AI Focus

Alternative Unit	Description
AI and Robotics	Students design, build, test, and improve projects combining hardware, such as sensors, motors, microprocessors, and circuit boards, with software, using a coding language to control motion, sound, lights, or data. They test their projects, gather peer feedback, and refine their code for clarity and accessibility, documenting their process and adjusting designs to be accessible to everyone.
AI and Cybersecurity	Students explore how AI has impacted cybersecurity, from spam filtering to automated recommendations across websites, search engines, ads, texts, and email. They investigate how cybercriminals use AI to launch more sophisticated, personalized, large-scale attacks and explore tools for real-time threat detection and automated response. Students can research the impacts and possible biases of using AI to analyze typing styles, voice patterns, and login behaviors for authentication, along with the data privacy concerns raised by AI systems that require access to large amounts of sensitive data.
Coding with AI	Students use age-appropriate tools to develop algorithms and programs for a variety of tasks, giving feedback at each step and evaluating the quality of what is produced. Students learn to attribute AI-generated code, images, and other content they use.

Artificial Intelligence Unit Learning Progression

Phase of Learning	Identifier	Standard	Application in AI
1. Understanding AI and ML	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.	Learn about different approaches to solving problems using computing technology and discuss when an AI solution is appropriate and beneficial, and when it may not be.
	MS-PRO-PD-14	Justify the importance of attribution and intellectual property when developing computing technologies.	Give appropriate credit to AI-generated content in schoolwork and online interactions.
	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.	Investigate the many different kinds of data stored online (traffic, health, finances, education, original art and music, social preferences, etc.) and discuss who the data belongs to and what the benefits and consequences are of their use by AI.
2. Train and Refine AI Models	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.	Identify where a simple AI predictor tool gives incorrect results and improve the model by adding additional training data.
	MS-SYS-IM-36	Collaborate to improve the design of a computing system to meet the needs of diverse users.	Suggest AI-powered solutions to problems encountered by diverse users.
	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.	Trace, in a small group, how a decision at one stage of a data project, such as collection, labeling, or cleaning, could bias an AI model's output and lead to misleading conclusions.

Phase of Learning	Identifier	Standard	Application in AI
3. Critically Evaluate AI Tools	MS-ALG-ML-08	Evaluate the features and limitations of a machine learning model.	Evaluate AI-generated art, music, videos, writing, or other products to describe their strengths and possible drawbacks.
	MS-PRO-RD-18	Analyze AI-generated code for accuracy and usability in a programming project.	Analyze AI-generated code for its readability, correctness, and efficiency.
	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.	Work in teams to envision an AI solution and outline its capabilities, limitations, and possible environmental impacts.
	MS-SYS-SE-33	Evaluate common types of cyber attacks and preventions.	Examine how private or sensitive data online is protected, and how it can still be compromised by good and bad actors.
4. Impacts of AI on People and Society	MS-SOC-HI-38	Compare the roles of individuals, communities, organizations, and governments in shaping computing technologies across major eras in computing history.	Explore the evolution of AI over the past 50 years and connect it to the broader evolution of computing hardware and software.
	MS-SOC-ET-42	Debate ways an emerging technology impacts the social, cultural, and environmental issues in local communities.	Debate the impacts of AI on the local community: energy, cost, environment, jobs, and fair access to resources and AI tools.
	MS-SOC-HU-43	Analyze how the decisions humans make when using computing technologies have ethical and social consequences.	Uncover applications of AI that affect the world and identify how misuse or biased data could have consequences.
	MS-SOC-CE-44	Analyze how workers in different careers use computational thinking to solve real-world problems.	Investigate a variety of careers that make use of AI and discuss the pros and cons of including AI in the workforce across different fields.
	MS-SOC-CE-45	Evaluate how automation in technology can create or replace jobs and change how people work.	Report on careers AI is making more efficient, then report on the AI tools most likely to replace human workers.

Cybersecurity Unit

Cybersecurity and digital literacy are intertwined, and a foundational understanding of both is essential for everyone. Cybersecurity focuses on the threats to our sensitive or private information and how we protect it.

In a 4–6 week unit, students examine how data is stored, transmitted, and protected, and what happens when those protections fail. They begin by mapping their own digital footprints and identifying the kinds of personal data that need protection. Students then explore common threats such as phishing and weak authentication, practice encrypting and decrypting messages, and analyze a real or simulated security incident to trace how it occurred and how it could have been prevented. The unit culminates in students designing a security plan for a realistic scenario, such as protecting a school club's member data, and justifying their choices in terms of both protection and usability.

A comprehensive introduction to cybersecurity at the middle school level should include:

1. **Introduction to Cybersecurity:** Core vocabulary, awareness of physical and digital forms of data storage and protection, and an understanding of how data is stored and communicated.
2. **Digital Responsibility:** Digital footprints, ethics, and responsible behavior online.
3. **Threats and Protections:** Cyber threats and threat prevention, encryption, and data and system security, including authentication, passwords, firewalls, hardware, and software.
4. **Impacts of Data Security:** Awareness of careers and the implications for freedom of speech and society.

Alternative Unit Designs with a Cybersecurity Focus

Alternative Unit	Description
Systems and Networks	Students gain a thorough understanding of the hardware and software that make up computing systems and devices, which is essential for a true understanding of cybersecurity.
Codebreaking and Cyberdefense	Students learn about historical codebreakers and modern cybersecurity experts, and how prime numbers, probability, and algebra form the bedrock of data encryption and network security.
Digital Forensics and Ethical Hacking	Students learn about cybersecurity in crime investigations, and careers focused on finding vulnerabilities in data security. With a focus on the human element, students learn about social engineering, phishing, baiting, pretexting, and manipulation, discovering that human behavior is often the weakest link in security. Students also evaluate everyday smart devices, like smart speakers or watches, for potential security flaws.

Cybersecurity Unit Learning Progression

Phase of Learning	Identifier	Standard	Application in Cybersecurity
1. Introduction to Cybersecurity	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.	Present on a given industry (such as health care, entertainment, or education) and explain what data and devices are used, which data is important to protect with cybersecurity measures, and why.
	MS-SYS-NT-34	Model how information in a network is broken down into packets, transmitted between devices, and reassembled.	Demonstrate how information flows across the internet and how the system is made as reliable and foolproof as possible.
	MS-SYS-NT-35	Explain how the resilience of the internet depends on interconnected devices and their roles and functions within the network.	Identify the names and functions of essential devices on the internet involved with the storage and transfer of information.
	MS-ALG-ML-06	Hypothesize how a machine learning model generates classifications or predictions.	Examine how automated systems, such as spam detectors for email or text, determine what to filter out and where they appear to be working, and how they could be improved.
2. Digital Responsibility	MS-SOC-HU-43	Analyze how the decisions humans make when using computing technologies have ethical and social consequences.	Evaluate what consequences there can be because of how decisions are made about how to use various websites, games, or social platforms.
	MS-DAT-DC-22	Explain how data and its associated metadata can be used to answer questions.	Explore how much can be learned about a person by information posted online, including location information included in photos and other hidden metadata.

Phase of Learning	Identifier	Standard	Application in Cybersecurity
3. Threats and Protections	MS-SYS-SE-33	Evaluate common types of cyber attacks and preventions.	Report on currently popular and historic cyber attacks, their consequences, and possible preventions, and explore different methods for protecting data, including encryption, authentication, firewalls, and passwords, along with the trade-offs of added security vs. inconvenience or costs.
	MS-SYS-SE-32	Explain the effects of not using the CIA triad when working with data.	Explain what can go wrong if any of the three parts of the CIA triad are not applied: data remains private (confidentiality), accurate and unaltered (integrity), and accessible to authorized users (availability).
	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.	Debate who gets to decide what data can be collected and who has the right to access it, along with possible consequences of making the wrong decisions.
4. Impacts of Data Security	MS-SOC-HI-38	Compare the roles of individuals, communities, organizations, and governments in shaping computing technologies across major eras in computing history.	Explore the history of cybersecurity and how decisions made about the structure and openness of the internet impact data security.
	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.	Report on data security breaches and discuss what can be done to improve cybersecurity with emerging technologies such as AI.
	MS-SOC-CE-44	Analyze how workers in different careers use computational thinking to solve real-world problems.	Identify the skills and knowledge needed for various careers in cybersecurity.

Model 3: Interdisciplinary Integration

In this model, computer science standards and practices are addressed deliberately within core content instruction. Many middle schools already integrate computing into subjects such as science, mathematics, social studies, or ELA. This model focuses on strengthening and formalizing those practices so that integrated instruction intentionally addresses computer science standards rather than relying on incidental technology use.

Is This Model Right for Your School?

This model fits schools where integration already occurs informally, scheduling limits standalone CS courses, content teachers are ready and interested, or the school is building CS capacity over time.

- **Main strength – access through existing structures:** Integration reaches all students within courses they already take and builds interdisciplinary collaboration.
- **Main challenge – rigor and coherence:** Without explicit standards alignment and cross-department coordination, integrated CS can dissolve into general technology use or fragment across classrooms.

Schools using this model should avoid short-term projects standing in for long-term standards, overestimating how much CS learning is already occurring, or placing responsibility on individual teachers without support.

What Interdisciplinary Integration Is and Is Not

Interdisciplinary integration is:	Interdisciplinary integration is *not*:
✔ Planned alignment between CS standards and content goals, advancing learning in both disciplines ✔ Repeated engagement with computational practices over time ✔ A means of expanding access while maintaining rigor	✘ Occasional or one-off activities ✘ General technology use (digital literacy) without standards-aligned learning goals ✘ Assuming CS learning occurred simply because a computer was used

High-quality integration depends on several conditions: explicit identification of the CS standards being addressed, collaboration among content teachers, CS educators, and instructional leaders, PD focused on standards and pedagogy, and time and structures for joint planning and revision. In many cases, modest adjustments in planning and documentation can significantly improve alignment to standards.

Debunking Common Myths About Interdisciplinary Integration

- **Myth:** Any technology use counts as computer science.
  Reality: CS requires explicit focus on computing concepts. Using an AI writing or image tool in another subject only becomes CS learning when the lesson addresses how the tool works.
- **Myth:** Integration makes computer science optional.
  Reality: Integrated models still hold CS as required learning when planned intentionally.
- **Myth:** Content teachers must become CS experts overnight.
  Reality: Content teachers need PD, planning time, and collaboration with CS colleagues.
- **Myth:** Integration is easier than standalone CS.
  Reality: Integration takes its own planning, coordination, and instructional leadership.

Building on Existing Standards Alignment

Many educators are already teaching content that naturally aligns with computer science practices, even if it has not been labeled as such. Schools should build on existing standards, connections, and instructional models rather than inventing new structures from scratch.

The standards were intentionally designed to align with other widely adopted frameworks, including:
- Common Core State Standards for English Language Arts
- Common Core State Standards for Mathematics
- Next Generation Science Standards

These alignments reflect shared practices such as problem solving, data analysis, modeling, argumentation, and communication. Schools can use the crosswalks to:
- Identify where CS standards already align with existing units
- Strengthen lessons by making CS learning goals explicit
- Reduce duplication by leveraging naturally overlapping practices

Planning Resources

Three resources support planning in this model: a crosswalk table for each grade, connecting that grade's standards to their aligned math, ELA, science, and other subject-area standards. Each table follows below. For easier viewing and more details, check out the interactive web display.

Grade 6 Progression Table with Interdisciplinary Connections

Identifier	Standard	Math	ELA	Science	Other
MS-ALG-PS-01	Design an algorithm that includes variables of multiple data types to solve a problem or express an idea.	6.EE.B.6	W.6.2; SL.6.1	MS-ETS1-2; MS-ETS1-3	Arts: VA:Cr2.1.6a
MS-ALG-PS-02	Model a given algorithm with a flowchart or pseudocode that includes a combination of control structures and procedures.	7.EE.B.3	W.6.2	MS-ETS1-1	Arts: MA:Cr2.1.8
MS-ALG-ML-06	Hypothesize how a machine learning model generates classifications or predictions.	8.SP.A.1; 7.SP.C.5	RL.7.1; RL.7.3	MS-LS2-2; MS-LS1-4	
MS-ALG-IM-09	Evaluate which human-centered design principles are present or missing in an existing computing technology.	7.SP.A.1	SL.8.1	MS-LS2-5	Arts: VA:Cr1.1.8a
MS-PRO-PD-12	Use a procedure to structure code for clarity and reusability.	6.EE.A.2	W.7.2.D	MS-PS1-4	Arts: MA:Cr2.1.7
MS-PRO-PD-14	Justify the importance of attribution and intellectual property when developing computing technologies.	6.SP.B.5.C; 8.G.A.5	W.8.8; W.7.8	MS-ETS1-3; MS-ETS1-4	Social Studies: D2.Civ.7.6-8; Arts: VA:Cr2.1.8a
MS-PRO-PD-15	Develop a program utilizing inclusive collaboration practices.	6.SP.B.5.A; 7.SP.C.7.A	SL.7.1; SL.8.1.C	MS-ETS1-1; MS-ETS1-2	Social Studies: D2.Civ.7.6-8
MS-PRO-VD-16	Use variables of multiple data types to store, access, and manipulate data within a program.	6.SP.B.5.c; 7.RP.A.3	W.6.2; SL.7.1	MS-ETS1-2; MS-ETS1-3	Social Studies: D2.Eco.3.6-8; Arts: VA:Cr3.1.8a

Grade 6 Progression Table with Interdisciplinary Connections

Identifier	Standard	Math	ELA	Science	Other
MS-DAT-DC-22	Explain how data and its associated metadata can be used to answer questions.	6.SP.B.5.B	RST.6-8.7		
MS-DAT-DC-23	Use a computational tool to sort, filter, group, and summarize structured data.	6.SP.B.4; 7.SP.A.1	WHST.6-8.7; SL.8.5	MS-LS2-1; MS-ESS3-5	Arts: VA:Cr2.3.8a; Social Studies: D2.Eco.11.6-8
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.	6.SP.B.5.A; 7.RP.A.2	RI.6.7; W.7.2	MS-ETS1-1; MS-ETS1-2	Social Studies: D2.Eco.2.6-8
MS-SYS-IM-36	Collaborate to improve the design of a computing system to meet the needs of diverse users.	6.SP.B.5.C; 7.RP.A.2	SL.6.1; W.7.2	MS-ETS1-2; MS-ETS1-3	Social Studies: D4.6.6-8; Arts: VA:Cr1.1.6a
MS-SOC-HI-38	Compare the roles of individuals, communities, organizations, and governments in shaping computing technologies across major eras in computing history.	6.SP.B.4; 7.RP.A.2	RH.6-8.6; WHST.6-8.1	MS-ESS3-3; MS-ETS1-1	Arts: VA:Cn11.1.7a
MS-SOC-HU-43	Analyze how the decisions humans make when using computing technologies have ethical and social consequences.	6.SP.B.5.A; 7.RP.A.3	SL.6.1; W.6.1	MS-ETS1-2; MS-ESS3-3	
MS-SOC-CE-44	Analyze how workers in different careers use computational thinking to solve real-world problems.	6.SP.B.5.c; 7.EE.B.4	W.7.2; SL.8.1.A	MS-ETS1-4; MS-ESS3-3	Social Studies: D2.Civ.13.6-8; Arts: VA:Cr2.1.7a

Grade 7 Progression Table with Interdisciplinary Connections

Identifier	Standard	Math	ELA	Science	Other
MS-ALG-PS-03	Verify the accuracy of an algorithm for given inputs.	8.F.A.1	RST.6-8.3	MS-ETS1-3	
MS-ALG-PS-05	Use an AI tool to generate outputs that assist in solving a computational problem.	7.EE.B.3	RST.6-8.8		
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.	7.SP.A.1		MS-ETS1-3	Arts: VA:Cr2.3.8a
MS-ALG-IM-10	Examine evidence of beneficial and harmful impacts, ethical issues, and biases of algorithms encountered in daily life.	6.SP.B.5.C; 7.SP.C.8	W.7.1; RI.6.6	MS-ETS1-3; MS-ETS1-2	Arts: VA:Re.7.2.6a; Health and PE: PE.2.12.15
MS-PRO-PD-13	Use reference documentation in program development.	7.EE.B.4; 6.SP.B.5.C	W.8.2.A	MS-ETS1-3	Arts: VA:Cr3.1.8a
MS-PRO-RD-17	Analyze the roles of iteration, selection, variables, and procedures in a segment of code.	8.F.A.2; 7.EE.B.3	L.6.6	MS-ESS2-5	
MS-PRO-TR-19	Use systematic strategies to test, refine, and document changes to a computing technology to meet the intended purpose.	6.SP.B.5.C; 8.F.A.1	W.7.2; SL.7.1	MS-ETS1-3; MS-ETS1-4	Social Studies: D2.Civ.9.6-8; Arts: VA:Cr3.1.8a
MS-DAT-DC-24	Analyze options to address a data quality issue.	6.SP.B.5.C; 7.SP.A.2	W.7.1.b; SL.8.1.A	MS-ESS2-5; MS-LS1-5	Arts: VA:Pr5.1.7a; Social Studies: D2.His.9.6-8

Grade 7 Progression Table with Interdisciplinary Connections

Identifier	Standard	Math	ELA	Science	Other
MS-DAT-DI-26	Create data visualizations to show how different design choices can impact the interpretation of the same data.	6.SP.B.5.B	RI.8.7	MS-LS2-2	
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.	6.SP.B.5.A	RI.6.8	MS-ETS1-1	Social Studies: D2.Civ.1.6-8
MS-SYS-SE-32	Explain the effects of not using the CIA triad when working with data.	6.RP.A.3; 7.SP.C.7	SL.7.1; W.8.1	MS-ETS1-2; MS-ETS1-3	Social Studies: D4.6.6-8; Arts: VA:Cr1.1.6a
MS-SYS-NT-34	Model how information in a network is broken down into packets, transmitted between devices, and reassembled.	6.RP.A.1	RST.6-8.3	MS-PS4-3	
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.	6.SP.B.4; 7.RP.A.3	SL.8.1; W.7.2	MS-ETS1-1; MS-ESS3-4	Social Studies: D4.6.6-8
MS-SOC-HI-39	Analyze intended and unintended impacts of a historical computing technology on society and the environment.	6.RP.A.3.C; 7.SP.A.2	W.7.1.A	MS-ESS3-4; MS-LS2-4	Arts: VA:Re.7.2.7a
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.	7.EE.B.3; 6.SP.B.5.C	WHST.6-8.1; RI.6.8	MS-ESS3-3; MS-ETS1-2	Social Studies: D2.Civ.3.6-8

Grade 8 Progression Table with Interdisciplinary Connections

Identifier	Standard	Math	ELA	Science	Other
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.	6.SP.A.1; 6.EE.B.6	W.6.1; RI.7.8	MS-ETS1-2; MS-ETS1-3	Social Studies: D2.Eco.7.6-8; Arts: MU:Re7.1.8a
MS-ALG-ML-08	Evaluate the features and limitations of a machine learning model.	6.SP.B.5.C	W.8.2.A	MS-ETS1-3	Arts: VA:Cr3.1.8a
MS-ALG-IM-11	Modify an algorithm to address a specific societal impact, ethical issue, or bias.	6.SP.B.5.B; 7.SP.C.8.C	RI.6.6; W.6.3	MS-ETS1-4	Arts: VA:Cr2.1.6a; Social Studies: D2.Civ.7.6-8
MS-PRO-RD-18	Analyze AI-generated code for accuracy and usability in a programming project.	8.SP.A.4	RI.7.4	MS-ETS1-3	
MS-PRO-TR-20	Refine a computing technology based on user feedback to improve its usability and accessibility.	6.SP.B.5.A; 7.RP.A.3	W.7.5; W.6.7	MS-ETS1-2; MS-ETS1-4	Social Studies: D2.Civ.12.6-8; Arts: VA:Cr3.1.8a
MS-DAT-DC-21	Evaluate how different levels of precision and granularity in data collection affect accuracy, storage, and analysis.	6.SP.B.5.C; 7.RP.A.2	SL.7.1; W.7.1	MS-ETS1-4; MS-PS4-1	Arts: VA:Cr2.3.7a; Social Studies: D2.His.13.6-8
MS-DAT-DI-25	Use a computational tool to identify relationships among variables in a dataset and make a classification or prediction.	8.SP.A.1	W.8.2.B	MS-LS2-1	Social Studies: D2.Civ.13.6-8
MS-DAT-DI-27	Summarize a data investigation process, including potential biases, limitations, and supporting evidence.	7.SP.A.1	W.8.1.B	MS-ESS3-5	Social Studies: D3.3.6-8

Grade 8 Progression Table with Interdisciplinary Connections

Identifier	Standard	Math	ELA	Science	Other
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.	7.SP.A.2	RI.7.8	MS-ETS1-2	Social Studies: D2.His.17.6-8
MS-SYS-HW-30	Examine differences between computing systems based on user needs, system requirements, and potential societal, environmental, and ethical impacts.	6.SP.B.4	RI.7.8; W.6.1	MS-ETS1-1; MS-ESS3-3	Social Studies: D2.Eco.1.6-8
MS-SYS-SE-33	Evaluate common types of cyber attacks and preventions.	6.SP.B.5.C; 7.SP.C.8	RI.6.8; W.7.1	MS-ETS1-1; MS-ETS1-2	Arts: VA:Cr2.1.7a; Arts: VA:Cn11.1.6a
MS-SYS-NT-35	Explain how the resilience of the internet depends on interconnected devices and their roles and functions within the network.	6.EE.B.8; 8.EE.B.5	RI.6.4	MS-PS4-3	
MS-SOC-ET-41	Evaluate how the decisions made while designing an emerging technology influence user experiences differently across different communities.	6.SP.B.5.C; 7.RP.A.3	SL.6.1; W.6.1	MS-ETS1-1; MS-ESS3-3	
MS-SOC-ET-42	Debate ways an emerging technology impacts the social, cultural, and environmental issues in local communities.	6.SP.B.5.A; 8.SP.A.1	RI.6.3; RI.8.2	MS-ETS1-1	
MS-SOC-CE-45	Evaluate how automation in technology can create or replace jobs and change how people work.	7.SP.A.1; 6.RP.A.3	W.8.1; RI.8.9	MS-ETS1-2; MS-ETS1-1	Social Studies: D2.Eco.7.6-8; Arts: VA:Cr2.2.8a

Appendix: Communication Tools for Middle School

Schools often find that one of the hardest parts of this work is not the planning itself, but how to talk about it with others. Families, teachers, and students are all trying to understand what's changing and what it means for them. Clear, consistent communication can make a big difference. It helps people feel informed, included, and more comfortable with the shift.

The tools below are meant to support those conversations. They are not scripts to follow exactly, but starting points that schools can adapt based on their community, tone, and needs.

Key Messages to Communicate

Before getting into specific audiences, it helps to be clear on a few core ideas. These messages tend to come up again and again, no matter who you are speaking with.

Question	Answer
Why does it matter?	Computer science helps students understand and shape the technology around them while building problem solving and creative thinking skills.
What is changing?	Computer science is becoming a consistent experience for all students across grades 6 to 8 rather than a one time or optional opportunity.
What will students experience?	Students will design, build, collaborate, and explore real world problems using technology.
What to expect during rollout?	Implementation will take time and may evolve as schools adjust schedules, curriculum, and support systems.

Messaging by Audience

Different groups will need different levels of detail and different entry points. The chart below is meant as a quick reference to help shape those conversations.

Audience	What They Need to Understand	Sample Language
Administrators	Structure, coherence across grades, and system level planning	Middle school computer science should be planned as a progression across grades 6 to 8. Schools may choose different models, but all students should have access to consistent, standards aligned learning experiences.
Teachers	What counts as computer science and how it connects to their work	Computer science includes problem solving, data, systems, and impact, not just coding. Many of these practices already exist in classrooms, and the goal is to make them more intentional and connected.
Families	Why this matters for their child	Students are not just using technology. They are learning how it works and creating with it. This builds skills that apply across many future paths.
Students	Engagement and belonging	You will design, build, and explore ideas using technology. You do not need experience. This is a space to create and try things out.
School Boards and District Leaders	Equity, long term vision, and access	This ensures all students have access to foundational computer science learning and prepares them for future opportunities while aligning to standards.

Ready to Use Communication Samples

Schools often need simple, adaptable language they can use to explain what middle school computer science is, why it matters, and what implementation will look like. The sample messages below are written for different audiences and are intended to be revised to fit local context, grade structures, and school voice.

Message to Families

Subject: What Computer Science Will Look Like in Middle School

Dear Families,

Our school is expanding middle school computer science so that all students have opportunities to learn how technology works, how it is designed, and how it shapes the world around them.

In middle school computer science, students do much more than use digital tools. They may design algorithms, build programs, work with data, explore how networks and systems function, and examine the benefits, risks, and real-world impacts of technology. This is different from passive screen time. Students are actively creating, testing, problem-solving, and thinking critically about how computing works and how it affects people and communities.

Computer science helps students build skills that matter across many areas of learning and life, including creativity, collaboration, communication, and problem solving. No prior experience is required. Our goal is for all students to see themselves as capable learners and creators in computer science.

We look forward to sharing more about what students will be learning and how this work will grow across the middle school years.

Sincerely,
[School or District Name]

Message to Teachers

Subject: Middle School Computer Science Rollout
Dear Colleagues, As we begin implementing middle school computer science across grades 6–8, it is important to remember that computer science is broader than coding alone. It includes problem solving, programming, data, systems, networks, and examining the social and ethical impacts of technology. Many of the practices connected to computer science are already present in classrooms, including analyzing information, designing solutions, interpreting data, troubleshooting, collaborating, and reflecting on how tools affect people. The goal is not to start from scratch. The goal is to make this learning more visible, intentional, and coherent across grade levels. As implementation moves forward, we will continue identifying where computer science concepts naturally connect to existing instruction and where additional support, planning, or collaboration may be helpful. This work will be strongest when it is shared across roles rather than carried by one person alone. Sincerely, [Teacher Name]

Message to Students

Subject: What You'll Be Doing in Computer Science
Dear Students, In middle school computer science, you might design a game, build something interactive, figure out how a system works, explore how apps and AI make decisions, or use data to answer questions about the world. You will have chances to test ideas, solve problems, work with others, and improve your work over time. You do not need to come in already knowing how to code or having any special experience. This is a place to learn, experiment, make mistakes, and build confidence. Computer science is for everyone, and there are many different ways to be successful in it. Sincerely, [Teacher Name]

Optional Tools Schools Can Use

Not every school will use all of these, but a few well chosen tools can make communication more visible and consistent.

- **Family information sessions**: Short sessions can help families understand what computer science includes, how it differs from general technology use, and what students will experience across the middle school years.
- **Student showcases or exhibitions**: Sharing student projects, investigations, or design work can make learning visible and help families and staff see computer science as creative, relevant, and connected to real-world thinking.
- **Staff planning and collaboration time**: Grade-level meetings, department meetings, or cross-role planning sessions can help teachers and leaders stay aligned on pacing, expectations, and support needs.
- **School website or newsletter updates**: Brief updates can keep families informed about what students are learning and why the work matters.
- **Student-facing visuals or announcements**: Posters, advisory announcements, or short orientation messages can help students understand that computer science is for everyone and that no prior experience is required.

csteachers.org/pk12standards/resources