KANSAS COMPUTER SCIENCE STANDARDS

Understanding the Standards

KANSAS STATE DEPARTMENT OF EDUCATION

Kansas leads the world in the success of each student.

Jan. 16, 2026

Contents

3 **Overview**
3 Five Topics of Computer Science
5 Progression through the Grade Levels and Standard Identifications

6 **Course Considerations**
6 Standalone or Integrated?
6 Computer Science as a Standalone Course
7 Computer Science Integrated into Other Courses
7 Overall

8 **Integration Methodology**
8 Integrating Computer Science into the P–12 Curriculum: An Introduction for Educators and Leaders
9 Integrating Computer Science in High School Math, Science, and Engineering
9 Mathematics
9 Science
10 Engineering

11 **Vertical Alignment**
11 Computing Systems
12 Networks and the Internet
13 Data and Analysis
14 Algorithms and Programming
15 Impacts of Computing

Overview

The *P-12 Kansas Computer Science Standards*, adopted April 16, 2019, provide expectations for student learning in five core concept areas: computing systems, networks and the internet, data analysis, algorithms and programming, and impacts of computing. These foundational areas span from early grades through high school, forming a cohesive progression in computational thinking, and all are vital for building technology skills needed for the 21st century work force.

Five Topics of Computer Science

1. Computing Systems (CS)

This strand introduces students to the components and interplay of hardware and software—from basic device operation to troubleshooting system issues.

- In early grades, learners identify input/output devices, power on computers, and use simple apps.
- As they advance, they explore system architecture, abstractions, and device layering.
- At advanced levels, they evaluate hardware-software trade-offs and develop troubleshooting guidelines.

2. Networks and the Internet (NI)

Students develop an understanding of how devices interconnect, how data packets traverse networks, and the evolving landscape of cybersecurity.

- Basic concepts include how information is transmitted and received across networked devices.
- Progressing through grade bands, learners examine protocols, encryption, topology, scalability, and reliability.
- They also study cybersecurity threats, privacy, and strategies to balance usability with protection.

3. Data Analysis (DA)

This focus area emphasizes collecting, organizing, analyzing, and interpreting data to make informed decisions.

- Younger students learn basic charting and tool selection for simple datasets.
- Middle-level learners refine methods to represent data, ensure reliability, and use computational tools to interpret patterns.
- At secondary levels, they build and evaluate models, create visualizations, and critically assess storage and quality trade-offs of different data.

4. Algorithms and Programming (AP)

This focus area engages students in writing, analyzing, and refining code.

- Early grades begin with sequencing, loops, and event-driven structures.
- Students advance toward modular design, conditionals, data structures, debugging, and collaboration.
- Upper-level learners evaluate multiple algorithmic solutions, examine efficiency, and engage with AI and classic algorithms like sorting and search.

5. Impacts of Computing (IC)

The final strand addresses how computing affects society, ethics, accessibility, and the environment.

- Students explore digital citizenship, intellectual property, and ethical online behavior from early grades.
- They then examine equity, career perspectives, global impacts of automation, and evolving technologies like social media and AI.
- At higher levels, learners tackle bias, privacy concerns, and design inclusive technology for diverse populations.

These five strands are reinforced by seven overarching practices: fostering inclusive culture, problem definition, abstraction, test-driven development, iteration, collaboration, and communication—with each concept area woven into a comprehensive Pre-K-12 progression[1] (Kansas State Department of Education).

1 KSDE. *Kansas Computer Science Model Standards with Description*. https://drive.google.com/file/d/1hlXKO46jY0YabI53RzCz53qvGww-WV2X/view

Progression through the Grade Levels and Standard Identifications

Each of the five topics, along with its subtopics, is divided into standards by education levels. From pre-K through fifth grade, enough granularity exists to suggest grade-level progression, with one column on the standards progression spreadsheet per grade. These standards are identified with a four-field alphanumeric identifier in which the first field is the grade level (Pre-K, K, 1, etc.), the second identifies the topic (CS for Computing Systems, NI for Networks and the Internet, etc.), the third identifies the subtopic, and the fourth is a numeric identifier for its place within the subtopic (most are 01; some are 02, etc.). Thus, the second standard in the third grade Program Development subtopic of Algorithms and Programming would be identified as 3.AP.PD.02.

Middle grades, usually sixth through eighth, are where significant topical exploration occurs with a great deal of variance of sequence, so those standards are grouped as one grade band. Standards are identified by the same four fields, but in these grades the first field is middle grades (MG).

High school students, generally ninth through 12th grades, have the opportunity to learn computer science concepts in two standard grade bands. The first, L1, consists of computer science concepts all high school students should know regardless of their future career fields. The second band, L2, contains concepts which are specific to one or more career fields specific to the computer science and information technology fields. These standards, identified with the same four fields but starting with L2, overlap closely with the career and technical education competencies for the digital technology cluster.

This framework empowers students in Kansas to become thoughtful creators and critical users of technology, equipping them with both the technical and ethical skills necessary to thrive in tomorrow's increasingly digital world.

Grade Level	Grade Signifier
Pre-Kindergarten	PK
Kindergarten	K
Grades 1-5 (per grade)	1, 2, 3, 4, 5
Grades 6-8 (Middle Grades)	MG
High School (Level)	L1, L2

Course Considerations

Standalone or Integrated?

Computer science and its underlying computational thinking skillset have been called, among other things, some of the most fundamental skills of the 21st century. Thus, as Kansas schools prepare schedules and curricula to lead the world in the success of each student, they must thoughtfully consider how to deliver computer science (CS) content—whether as a standalone course or integrated across subjects like math, science, literature, art, or others. Both approaches offer distinct advantages and disadvantages.

COMPUTER SCIENCE AS A STANDALONE COURSE

Advantages:

1. **Depth and Rigor**

   A dedicated course allows for deeper exploration of core CS concepts—algorithms, programming, data structures, cybersecurity, and digital citizenship—without competing for instructional time and student engagement with other content areas.

2. **Clear Curriculum Pathways**

   Especially at the secondary level, standalone CS courses (like Advanced Placement Computer Science Principles or Introduction to Coding) provide students with structured opportunities to build upon skills over time, preparing them for college or careers in tech.

3. **Dedicated Expertise**

   Some schools can assign specialized teachers with CS education background, which likely presents students with more thorough exposure to the topics in both breadth and depth.

Disadvantages:

1. **Scheduling and Access Barriers**

   At the secondary level, students may not be able to fit an extra CS class into their schedule—especially in high school where electives compete for space. This can lead to equity gaps.

2. **Staffing Challenges**

   Hiring or developing qualified CS teachers remains a challenge, particularly in rural or under-resourced schools.

3. **Risk of Isolation**

   Treating CS as separate from other subjects may limit students' understanding of how it connects to real-world applications across disciplines.

COMPUTER SCIENCE INTEGRATED INTO OTHER COURSES

Advantages:

1. Authentic Application

   Integration makes CS feel relevant. For example, using code to simulate biological systems in science, or exploring algorithms in math, or robots to simulate a dance routine, helps students see CS as a real-world problem-solving tool.

2. Increased Access

   Embedding CS in core subjects ensures that all students—regardless of background or schedule—get exposed to computational thinking and its component skills.

3. Interdisciplinary Learning

   Integrating CS fosters collaboration between departments and strengthens students' critical thinking and creativity across domains. Imagine analyzing Frankenstein through the lens of artificial intelligence ethics!

Disadvantages:

1. Diluted Focus

   Integration risks surface-level coverage of CS concepts. Teachers may touch on ideas like loops or data visualization, but students may not gain fluency in them.

2. Professional Development Needs

   Teachers in other subjects may feel unprepared to teach CS content. Without robust training and curricular support, integration can feel flat or tentative.

3. Assessment and Accountability

   It's harder to measure student growth in CS when it is blended into other subjects. This can impact program evaluation and improvement efforts.

Overall

There is no one-size-fits-all solution. At the elementary level, integration can be a powerful way to build foundational CS thinking while keeping learning playful and connected. At the secondary level, offering both standalone and integrated opportunities can provide a more comprehensive offering—ensuring all students gain exposure, while those with interest can dive deeper.

Educators and leaders work to design a system that honors the diverse needs of all students, supports teachers with professional learning, and prepares all learners to thrive in a world increasingly shaped by technology. The very nature of the rapidly changing technology field ensures that educators must continue this conversation together as a team—balancing vision with practicality—always keeping students at the center of decisions.

Integration Methodology

Integrating Computer Science into the P–12 Curriculum: An Introduction for Educators and Leaders

As the digital landscape reshapes how we live, learn, and work, computer science (CS) has emerged as a foundational literacy, essential not just for aspiring technologists, but for all high school graduates regardless of chosen career field. The core importance of CS lies in a concept called "computational thinking," which embodies a technical approach to problems through a conceptualized algorithmic lens. Granted the importance of computational thinking, the *K–12 Computer Science Framework*[2], developed through national collaboration, emphasizes computer science is not solely a standalone discipline. Rather, it is a flexible, interdisciplinary field that can be meaningfully integrated across subjects to deepen learning, empower creativity, and prepare students for a rapidly evolving world.

In alignment with this framework, the *Kansas Computer Science Model Standards* provide a clear and adaptable structure for embedding computer science concepts and practices into a wide range of classrooms. Organized around five key concept areas—computing systems, networks and the internet, data and analysis, algorithms and programming, and impacts of computing—and supported by cross-cutting practices such as problem-solving, collaboration, and ethical reasoning, these standards offer practical entry points for educators in all disciplines.

For teachers and curriculum leaders, integrating CS begins with recognizing natural intersections between content areas. Mathematics classes can explore algorithmic thinking and data visualization; science instruction can benefit from modeling and simulation; language arts can embrace digital storytelling and logical structure. Even social studies and the arts can incorporate CS by examining the societal impacts of technology, or by using technical tools for analysis and creative expression. This interdisciplinary approach not only reinforces academic content but also fosters critical thinking, digital fluency, and real-world relevance.

Moreover, the Kansas CS standards encourage equitable access to computer science by promoting early exposure and gradual progression from elementary through high school. This scaffolding ensures that students develop core skills over time, whether through dedicated CS courses or integrated experiences within existing subjects. By leveraging this framework, educators can personalize integration strategies based on student needs, available resources, and schoolwide goals.

In embarking upon a journey toward CS integration with other subject areas, collaboration is key. Engage with colleagues across grade levels and disciplines, explore professional development opportunities, and experiment with small, meaningful changes. Whether embedding CS practices into a literature unit or launching a cross-curricular STEM project, the curriculum team is not just teaching content—they are preparing students to thrive in a world shaped by algorithm, computation, and creativity.

2 https://k12cs.org/wp-content/uploads/2016/09/K–12-Computer-Science-Framework.pdf

Integrating Computer Science in High School Math, Science, and Engineering

With Classroom Examples for Kansas Educators

MATHEMATICS

Computer science and mathematics are natural partners in logic, precision, and structure. The Kansas Computer Science Standards—especially those focused on algorithms and programming, data and analysis, and abstraction—complement the high school math curriculum by offering students real-world applications of mathematical reasoning.

Specific examples of integration in math include:

1. **Writing basic block-based or text-based programs to solve quadratic equations**, allowing students to explore how changing coefficients affect the graph and solutions.
2. **Using spreadsheets or data science tools** to analyze real-world datasets in statistics or algebra, reinforcing concepts like mean, median, standard deviation, and linear regression.
3. **Creating recursive algorithms to model geometric patterns or sequences**, connecting recursion to exponential growth or fractals.
4. **Applying modular arithmetic in cryptography exercises**, linking number theory with real-world applications in cybersecurity.
5. **Building logic circuits in a virtual simulator** to reinforce conditional reasoning and Boolean logic.

Through these and similar exercises, students develop a deeper understanding of math concepts while practicing computational thinking skills like decomposition and pattern recognition. This supports math achievement and prepares students for technical fields where math and code go hand-in-hand.

SCIENCE

The integration of computer science in science classrooms enriches inquiry by enabling modeling, simulation, and large-scale data analysis. Kansas standards encourage using technology not only to support scientific understanding but also to facilitate students to become creators of scientific tools and models.

Specific examples of integration in science include:

1. **Using a computerized modeling environment to simulate ecological systems**, such as predator-prey dynamics or disease spread, reinforcing biology and environmental science concepts.
2. **Programming temperature sensors with microcontrollers** to collect real-time climate or lab data and graph it using Python or spreadsheet tools.
3. **Analyzing real-world data sets from sources like NASA, NOAA, or local weather stations**, teaching students to clean, process, and interpret large sets of climate or space data.
4. **Simulating projectile motion or orbital mechanics with physics engines or code**, connecting Newtonian motion with computational models.
5. **Using computer vision to measure chemical reactions or physical changes in time-lapse experiments**, introducing image analysis and automation into lab work.

These experiences let students go beyond passive observation to actively creating and manipulating scientific models, fostering deeper understanding of core concepts and aligning with the data and analysis, computing systems, and impacts of computing strands in the Kansas standards.

ENGINEERING

High school engineering courses naturally align with computer science through problem-solving, iteration, and system design. Integrating CS deepens engineering projects, improves technical precision, and opens access to automation, robotics, and advanced manufacturing—all critical areas in today's workforce.

Specific examples of integration in engineering include:

1. **Designing and programming robotic arms or autonomous vehicles using platforms like VEX, LEGO EV3, or Arduino**, where students apply sensor data and control flow.
2. **Using CAD software integrated with scripting** to design customizable 3D models based on parameters students define through code.
3. **Programming microcontrollers to collect and respond to environmental data**, such as temperature, light, or movement, for smart-home or environmental monitoring systems.
4. **Simulating structural stress and testing designs in physics-based software**, allowing students to predict and optimize their prototypes before building physical models.
5. **Creating logic flowcharts and decision trees as part of the engineering design process**, helping students break down problems and document their thinking using CS-aligned practices like abstraction and algorithmic design.

These projects build skills outlined in both engineering and CS frameworks, including creating computational artifacts, refining solutions, and understanding computing systems. They also empower students to design innovations that respond to real-world needs—whether mechanical, environmental, or digital.

By embedding computer science in existing math, science, and engineering courses, Kansas educators can make their classrooms more dynamic, hands-on, and future-ready. These integrations not only reinforce academic standards but also prepare students for a workforce that increasingly demands both domain expertise and computational fluency. Through thoughtful, creative application of the Kansas Computer Science Standards, schools can help students bridge the gap between theory and practice, unlocking new paths for learning, innovation, and career readiness.

Vertical Alignment

Below is a suggested vertical alignment approach for each of the five topics in the Computer Science (CS) standards. General best practices apply to all. For example, ensure the participation of involved teachers from all grade levels and topics involved. Move from general topics to specific learning objectives. Third, remember that this is an iterative process that must be revisited to identify, using observations as well as available data, learning gaps.

Computing Systems

Elementary

Begin with foundational knowledge—helping students understand what computers are and how we interact with them. Students in early grades can identify basic hardware (keyboard, mouse, screen) and understand input/output processes. Activities can include disassembling old devices, drawing labeled diagrams, or using educational software to explore system parts.

Middle School

In middle grades, students expand their understanding of computing systems to include networks, peripheral devices, storage, and software interaction. Teachers can introduce system layering (hardware → OS → apps), basic troubleshooting, and digital citizenship tied to responsible use of devices. Learners can conduct comparative research on different types of computing devices, including tablets, desktops, and embedded systems.

High School

High school students deepen their understanding by exploring system architecture, abstraction, and performance trade-offs. They may build or configure computers, evaluate various operating systems, or analyze the role of computing systems in cybersecurity and innovation. Students should understand the implications of hardware/software choices and apply that knowledge in capstone or design-based projects.

ALIGNMENT STRATEGY

Ensure vertical alignment by mapping each grade band's focus: move from **identification and use (K–5) → systems analysis and troubleshooting (6–8) → technical design, optimization, and ethical use (9–12)**. Provide professional development on how computing systems support inquiry and productivity in all subjects.

Networks and the Internet

Elementary

Introduce the idea that computers can connect and share information. Use real-world analogies like a school intercom system or postal mail to explain how messages travel. Engage students with basic vocabulary such as "internet," "Wi-Fi," and "online safety," and help them identify devices that communicate wirelessly.

Middle School

In these grades, explore how data travels through the internet, using concepts such as packets, protocols, and IP addresses. Teachers can introduce simple diagrams of how local networks and the internet operate, and conduct basic simulations of how messages are broken down and reassembled. Digital security practices and awareness of cyber threats are essential components.

High School

Students at this level can investigate deeper aspects of networks—like DNS resolution, bandwidth, encryption, firewalls, and routing. They can also examine societal and economic impacts of global connectivity and evaluate technologies like IPv6, VPNs, and the cloud. High school learners may also simulate attacks and defenses in controlled cybersecurity environments.

ALIGNMENT STRATEGY:

Ensure progression from **understanding connectivity and digital communication (K–5) → network structure and function (6–8) → internet architecture, security, and global impact (9–12)**. Offer sample units on email packet travel, cyber hygiene, or internet history to help bridge topics between grades.

Data and Analysis

Elementary

Start with collecting and organizing simple data sets (e.g., favorite fruits, classroom weather charts). Students learn to use bar graphs, pictographs, and tables to represent and interpret information. This stage builds foundational skills in identifying patterns and basic inferencing.

Middle School

Introduce more complex data gathering and analysis, including surveys, measurement tools, and digital data logging. Students can use spreadsheets or coding platforms to create charts and analyze patterns. They can explore how data is stored and retrieved, and begin questioning reliability and bias in data.

High School

Students should be able to gather, clean, process, and visualize large data sets. Data tools can and should be utilized. Students also study issues like data privacy, ethical data use, and the influence of data science on decision-making in real-world contexts such as marketing, health, and politics.

ALIGNMENT STRATEGY:

Build from **data collection and basic visualization (K–5) → data manipulation, comparison, and simple tools (6–8) → data science methods, ethical considerations, and automation (9–12)**. Scaffold instruction so students progressively move from concrete to abstract data concepts.

Algorithms and Programming

Elementary

Start with unplugged activities using sequencing and pattern games, then transition to block-based coding (e.g., Scratch Jr., Code.org). Focus on loops, events, conditionals, and simple problem-solving using clear step-by-step instructions. The goal is to help students understand the logic behind programming.

Middle School

Introduce more sophisticated block coding and simple text-based programming. Teach abstraction, modular design, and debugging through scaffolded projects. Encourage students to work in pairs or teams to build programs that serve a function or solve a challenge, such as simple games or animations.

High School:

At this level, students should be proficient in writing, analyzing, and debugging code in one or more programming languages (e.g., Python, JavaScript). They explore algorithmic efficiency, recursion, object-oriented design, and data structures. Students also evaluate and compare algorithms for performance and complexity.

ALIGNMENT STRATEGY:

Ensure progression from **basic sequence and logic patterns (K–5) → structured programming with modularity and testing (6–8) → complex algorithm design and efficiency analysis (9–12)**. Encourage cross-discipline projects using code in science labs, math models, or robotics.

Impacts of Computing

Elementary

Teach students how technology affects their daily lives—communication, transportation, entertainment. Discuss online safety, respect, and how to be kind in digital spaces. Introduce the idea that computers are tools we control. Highlight diverse people who work in tech fields.

Middle School

Delve into broader discussions about how computing affects society, including issues of access, equity, and media influence. Introduce the concepts of bias in algorithms and ethical concerns such as plagiarism and responsible sharing. Students can analyze real-world examples of technology use in culture and economics.

High School

Students evaluate the global impacts of computing, from automation and surveillance to sustainability and the future of work. They research case studies and propose ethical responses to emerging issues in AI, privacy, and cybersecurity. Projects may include digital portfolios or debates on computing ethics.

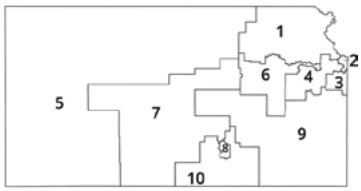
ALIGNMENT STRATEGY:

Design a progression from **personal impact and digital citizenship (K–5) → societal impact and ethical scenarios** (6–8) **→ global implications, equity, and philosophical questions (9–12)**. Use current events and media literacy strategies to keep lessons relevant and discussion-rich.

Kansas State Board of Education

BOARD MEMBERS

900 S.W. Jackson Street, Suite 600
Topeka, Kansas 66612-1212
(785) 296-3203
www.ksde.gov/board



DISTRICT 1



Danny Zeck
Vice Chair
Danny.Zeck@ksde.gov

DISTRICT 2



Melanie Haas
Melanie.Haas@ksde.gov

DISTRICT 3



Michelle Dombrosky
Michelle.Dombrosky@ksde.gov

DISTRICT 4



Connie O'Brien
Connie.O'brien@ksde.gov

DISTRICT 5



Cathy Hopkins
Chair
Cathy.Hopkins@ksde.gov

DISTRICT 6



Dr. Beryl A. New
Beryl.New@ksde.gov

DISTRICT 7



Dennis Hershberger
Dennis.Hershberger@ksde.gov

DISTRICT 8



Betty Arnold
Betty.Arnold@ksde.gov

DISTRICT 9



Jim Porter
Jim.Porter@ksde.gov

DISTRICT 10



Debby Potter
Debby.Potter@ksde.gov

SUCCESS DEFINED

A successful Kansas high school graduate has the

- Academic preparation,
- Cognitive preparation,
- Technical skills,
- Employability skills and
- Civic engagement

to be successful in postsecondary education, in the attainment of an industry recognized certification or in the workforce, without the need for remediation.

OUTCOMES

- Social-emotional growth
- Kindergarten readiness
- Individual Plan of Study
- Civic engagement
- Academically prepared for postsecondary
- High school graduation
- Postsecondary success

Kansans CAN

MISSION

To prepare Kansas students for lifelong success through rigorous, quality academic instruction, career training and character development according to each student's gifts and talents.

VISION

Kansas leads the world in the success of each student.

MOTTO

Kansans Can

COMMISSIONER OF EDUCATION



Dr. Randy Watson
Randy.Watson@ksde.gov

DEPUTY COMMISSIONER
Division of Fiscal and Administrative Services



Dr. Frank Harwood
Frank.Harwood@ksde.gov

DEPUTY COMMISSIONER
Division of Learning Services



Dr. Renee Nugent
Renee.Nugent@ksde.gov

The Kansas State Department of Education does not discriminate on the basis of race, color, religion, national origin, sex, disability or age in its programs and activities and provides equal access to the Boy Scouts and other designated youth groups. The following person has been designated to handle inquiries regarding the nondiscrimination policies: KSDE General Counsel, Office of General Counsel, KSDE, Landon State Office Building, 900 S.W. Jackson, Suite 102, Topeka, KS 66612, (785) 296-3201.

Kansas leads the world in the success of each student.

July 1, 2025

For more information, contact:

Dr. Stephen King
Education Program Consultant
Career, Standards and Assessment Services
(785) 296-7285
stephen.king@ksde.gov



Kansas State Department of Education
900 S.W. Jackson Street, Suite 102
Topeka, Kansas 66612-1212

https://www.ksde.gov