

Computer Science Models & Approaches for Elementary, Middle, and High School Settings in Minnesota



Northern Lights Collaborative
for Computing Education

UNIVERSITY OF MINNESOTA





Introduction

The Minnesota Department of Education defines computer science (CS) as “... the study of computers and algorithmic processes, including their principles, hardware and software designs, implementation, and impact on society. Computer science education should focus on teaching students how to think computationally and create new technologies, not simply use technology.”

This document offers guidance and inspiration to advance computer science (CS) learning for **all students**, in accordance with Minnesota’s state strategic plan for the sustained growth of CS education across all K–12 districts and charter schools ([Minn. Stat. §120B.241](#)). Our goal is to balance breadth and accessibility - providing enough detail to support decision making but making it accessible enough for those new to CS education.

A K–12 CS pathway offers students opportunities at the elementary, middle, and high school levels to build foundational knowledge and progressively deepen their CS skills. Drawing from multiple sources—including interviews with 11 Minnesota schools implementing K–12 CS pathways, examples from project schools partnering on a National Science Foundation grant to develop pathways in rural and tribal settings, and insights from CS education research—this document highlights a range of models and approaches for CS education.

These elements represent components of a quality, equitable CS learning experience across all three grade band models for K-12 students:

- **Progression:** Knowledge and skills spiral upward in complexity, but identity, equity, and problem-solving remain central.
- **Pedagogy:** Active, hands-on, collaborative learning with developmentally appropriate opportunities for self-expression and choice.
- **Support:** Strong PD, communities of practice, curriculum resources and materials, and administrator backing.
- **Equity:** Access, representation, and culturally relevant teaching.
- **Connections:** Links to real-world issues, community, technology use, and responsible digital citizenship.



Dimensions	Elementary (K-5)	Middle (6-8)	High School (9-12)
Purpose for CS	Spark curiosity, build foundational knowledge and skills for all students in computational thinking and technology	Foster skills and persistence, deepen knowledge and skills for all students	Provide depth, readiness, workplace skills, connect students with future career opportunities
Student Experience	Playful, exploratory, engaging, hands-on, integrated, cross-disciplinary, unplugged/plugged activities	Structured, project-based, hands-on, relevant, address social influences	Emphasis on choice, focus, and career pathways
Equity	Universal access, culturally relevant curriculum, diverse role models	Active recruitment of underrepresented students, multiple entry points to CS for students	Open access to advanced opportunities, active recruitment and support for underrepresented groups
CS Goals	Students begin to develop problem-solving strategies, see mistakes as part of learning, persist through challenges, and develop a sense of belonging	Students learn to analyze problems and test different solutions, apply CS to real-world issues, feel a sense of belonging, and grow their CS identity	Students apply advanced problem-solving strategies to real-world CS challenges, and maintain a stable CS identity



Elementary School

Approaches and Considerations for Staffing and Content

	Specialists	General Classroom Teachers
Commonly Taught By	Library/media, STEM/STEAM, technology specialists	Grade K - 5 classroom teachers <i>May have push-in support from digital innovation specialists, tech integrationists, etc.</i>
Considerations	• When specialists see all students during their rotations, all students receive CS instruction • Easier to provide professional development to a limited set of specialists	• Large community of support • CS can be discussed as part of grade level PLCs
	• If the specialist leaves the district, may be difficult to find a replacement with CS skills • If a school uses pull-outs during specialist time for CS education, students with disabilities will not receive CS instruction • Instruction time is limited to specialist rotations and may only be for a portion of the year (e.g., 30 minutes / week for 8 weeks)	• Challenging to have all teachers agreeing to teach CS • Challenging to provide professional development for all teachers • Elementary teachers struggle to teach standards that are not assessed (e.g., social studies) and focus on math, ELA and to some degree science

Curriculum & Materials Examples

Note that our list includes curriculum and tools that meet one or more of the following criteria: (1) part of the CSTA accredited list for PD and curriculum, (2) mentioned as used by model schools in interviews or by Pathways grant participating schools, (3)



evidence-based). Also, an asterisk (*) indicates that there is a cost for the device and/or curricular resource and an (I) denotes integrated curriculum (versus standalone).

	Grades K-2	Grades 3-5
Robotics and Physical Computing	Bee Bot*, Code and Go Mouse*, Primo Cubetto*, Vex 123*, Sphero indi*, Kibo*	Sphero Bolt*, Vex 123* and Go*, Edison Robot*, micro:bit*, Makey Makey*
Programming Environments	ScratchJr	Scratch Code.org App Lab, Web Lab, Game Lab Microsoft MakeCode
Curricular Resources	Barefoot Computing BootUp Code.org CS Fundamentals KinderLab Robotics(*) NLC K-8 Integrated CS Repository (I) Tiny Techies	Barefoot Computing BootUp Code.org CS Fundamentals Code Savvy Byte-Sized Challenge Experience CS (I) NLC K-8 Integrated CS Repository (I)
Unplugged	Barefoot Computing Code.org Unplugged CSUnplugged.org	Barefoot Computing Code.org Unplugged CSUnplugged.org



Middle School

Approaches and Considerations for Staffing and Content

	Standalone Courses (Elective or Required)	CS Integration
Commonly Taught By	• Specialists - STEM/STEAM, technology, library/media • Business or Math educators	Varies, but all new Minnesota standards have CS integration
Considerations	• All students may receive CS instruction if the course is required • Easier to provide professional development to a limited set of educators • Block-based scheduling may help create space for CS	• If CS is integrated in a required course (e.g., Math, Science, Social Studies), then all students receive CS instruction • Students may see applications of CS in multiple contexts • Does not require adding new courses or electives to student and teacher schedules
	• If the specialist leaves the district, may be difficult to find a replacement with CS skills • If a school uses pull-outs during specialist time for CS education, students with disabilities will not receive CS instruction • Instruction time is limited to specialist rotations and may only be for a portion of the year (e.g., 30 minutes / week for 8 weeks)	• Challenging to have all grade level or subject area teachers agreeing to teach CS • Challenging to provide professional development for all grade level or subject area teachers • Number of hours of CS instruction may be limited within another course



Curriculum & Materials Examples

Note that our list includes curriculum and tools that meet one or more of the following criteria: (1) part of the CSTA accredited list for PD and curriculum, (2) mentioned as used by model schools in interviews or by Pathways grant participating schools, (3) research-backed). Also, an asterisk (*) indicates that there is a cost for the device and/or curricular resource and an (I) denotes integrated curriculum (versus standalone).

	Grades 6-8
Robotics and Physical Computing	Sphero Bolt*, Lego Prime*, Vex 123* and Go*, Edison Robot*, micro:bit*
Programming Environments	Code.org App Lab, Web Lab, Game Lab Microsoft MakeCode Scratch Snap!
Curricular Resources	Art + Code (Processing Foundation) (I)(*) BJC Sparks Bootstrap: Algebra (I) Bootstrap: Data Science (I) CodeHS* Code.org CS Discoveries CodeSavvy Byte Sized Challenge Scratch Creative Computing Experience CS (I) Science: Project GUTS (I)
Unplugged	CSUnplugged.org



High School

Approaches and Considerations for Staffing and Content

Note that our list includes curriculum and tools that meet one or more of the following criteria: (1) part of the CSTA accredited list for PD and curriculum, (2) mentioned as used by model schools in interviews or by Pathways grant participating schools, (3) research-backed.

	Required, Elective, and Advanced courses
Commonly Taught	• CTE (Career and Technical Education) pathways: Schools can embed CS into existing CTE pathways and leverage Perkins and state funds to offer CS • Advanced Placement Courses (e.g., Computer Science Principles, Computer Science A) • Concurrent Enrollment (also known as College in the Schools – CIS or CITS) • Electives (e.g., Cybersecurity, Networking, Mobile App Development)
Considerations	• CTE programs are often designed for diverse learners who may be underrepresented in CS courses • Perkins and state funds can be leveraged when CS is part of CTE programming • High schools often link CS to potential career paths • Offering HS CS courses with prerequisites can restrict access for students with limited prior CS experience or interest

Course & Curriculum Examples

	Grades 9-12
Rigorous Courses	<i>Advanced Placement (AP)</i> Courses: Computer Science Principles and Computer Science A Materials: College Board maintains lists of endorsed curriculum and professional development providers for both courses (CSP List, CSA List)



	Grades 9-12
	<i>Concurrent Enrollment (CE) / College in the Schools (CITS)</i> CS Principles: Southwest Minnesota State University Note: other Minnesota colleges may offer credit for other CS courses, including through articulation agreements (e.g., Lake Superior College High School Articulation) <i>International Baccalaureate (IB)</i> Learn more about IB computer science courses
Artificial Intelligence	See AI4K12 resources
Cybersecurity	See Cyber.org and TeachCyber College Board's AP Cybersecurity (launches in 2026-27)
Data Science	See Data Science for Everyone resources
Game and Interactive Media	Currently, there is no vetted curriculum repository for either pathway. See MDE CTE Frameworks in the Business-Information Technology Cluster and Trade and Industry-Communications Technology Cluster for guidance on courses and learning objectives. CSTA will have standards for these specialty areas in July 2026 (see Draft 3).
Programming	
X + CS	Currently, there is no vetted curriculum repository for these categories. CSTA will have standards for these specialty areas in July 2026 (see Draft 3).
Physical Computing	



Resources

The resources listed here are widely used in Minnesota schools and were selected for their alignment with equitable and inclusive teaching practices. Their inclusion does not imply endorsement or recommendation by the Northern Lights Collaborative for Computer Science Education.

1. [Minnesota's Computer Science State Strategic Plan](#)
2. NLC's Overviews of Computer Science in:
 - a. [Elementary Settings](#)
 - b. [Middle School Settings](#)
 - c. [High School Settings](#)
3. [CSTA's Reimaging CS Pathways](#) (for high school to college)
4. NLC's [CS Learning AID Checklist](#) for integrated CS lessons
5. [CSTA K-12 Standards](#) (for students)
6. [Universal Design for Learning in Computer Science Education](#)
7. [Culturally Responsive-Sustaining Computer Science Education: A Framework](#)
8. [CAPE Framework](#)
9. [MA Digital Literacy & Computer Science Curriculum Guide](#) (includes an extensive review of curricula at elementary, middle, and high school levels)



This work was developed by [Northern Lights Collaborative for Computing Education](#) at [UMN](#) in collaboration with Cooperative Ventures. This material is based upon work supported by the National Science Foundation under Award No. 2401154. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

