south dakota
DEPARTMENT OF EDUCATION
Learning. Leadership. Service.

South Dakota

Computer Science Standards

Adopted April 14, 2025

Contents

Introduction

At no point in the future will the current generation of students use less technology than they do now.[1] Whether a student chooses a career in a technology field or not, technology and computer science (CS) are transforming all job sectors. So, how are we preparing students for this shift? By default, students are consumers of technology, but they also need to be creators. Being a creator means students are empowered to demonstrate their understanding of technology and to create new technologies that will drive innovation. This also means they'll be prepared for the jobs of the future.

In the spring of 2024, the Department of Education created the Computer Science Standards Committee to compile and recommend K-8 computer science standards, laying the foundation for a better understanding of the technology in students' everyday lives. This initiative aligns with the South Dakota Department of Education's aspiration for all students to leave the K-12 education system "College, Career, and Life Ready." It also integrates into the larger vision of other initiatives South Dakota has developed at the high school and collegiate levels, such as computer science Career and Technical Education (CTE) classes in high school. The Computer Science Standards Committee encompassed experienced K-12 educators and science professionals. Through many hours of discussion and collaboration, the team focused on ensuring best practices were embedded in the standards while maintaining rigor, attainability, and user-friendliness.

[1] Kirsten Baesler at CSEdCon 2021: Lessons Learned from State and District Computer Science Initiatives

A shared understanding of computer science is essential, as it encompasses various concepts and applications. Technically, as the Computer Science Teachers Association (CSTA) defines it, Computer Science is “the study of computers and algorithmic processes, including their principles, their hardware and software designs, their implementation, and their impact on society.” Although coding is an integral part of computer science, it is not the only aspect. Other aspects are artificial intelligence, networks, security, database systems, human-computer interaction, vision, and graphics. Put simply, computer science is about using/creating technology to solve problems, make connections, and/or create art or games, often in collaborative and creative ways.[2]

This comprehensive understanding of computer science underpins the development of educational standards that aim to equip students with the necessary skills and knowledge in this field. Various documents, including the 2017 Computer Science Teachers Association (CSTA) K-12 Computer Science Standards, the 2016 K-12 Computer Science Framework, and similar states’ Computer Science standards, inspired the formation of South Dakota’s K-8 Computer Science standards. Although 11 state standards were referenced, most adaptations came from Montana’s, Idaho’s, and North Dakota’s state standards. Finally, Dakota State University was consulted, as they are experts in the field.

The CSTA Computer Science Standards are designed to provide a comprehensive framework for K-12 computer science education. These standards are composed of two main components: concepts and practices. Concepts represent the core content areas of computer science, including algorithms and programming, computing systems, data and analysis, impacts of computing, and networks and the internet. Conversely, practices refer to the behaviors and thought processes that students should develop, such as computational thinking, problem-solving, and societal considerations. Each standard is written with a combination of a concept and a practice, ensuring that students acquire fundamental knowledge and develop essential skills for applying that knowledge in real-world contexts. This integrated approach provides a well-rounded education that prepares students for the evolving demands of the digital world.

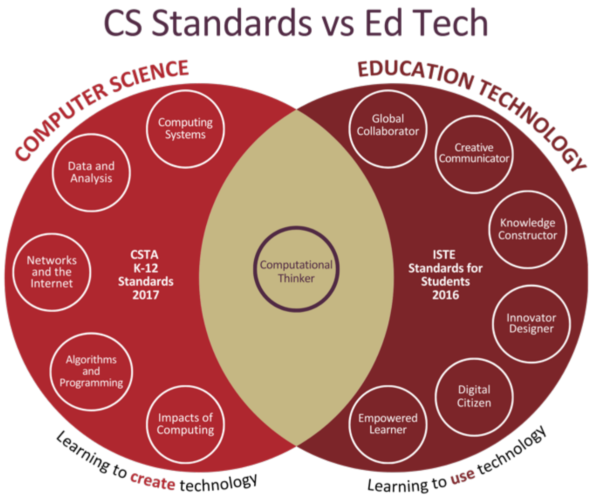
The K-12 Computer Science Framework describes the practices as follows: “The seven core practices of computer science describe the behaviors and ways of thinking that computationally literate students use to fully engage in today’s data-rich and interconnected world.”[3] The order does indicate a process to create a computational artifact or a computer program or action, but it is a cyclical process that can follow many paths. Practices three through six relate to computational thinking. Computational thinking (CT) is a skill set that involves solving problems, defining systems, and understanding human behavior by drawing on the concepts fundamental to computer science.[4] This includes skills like logical reasoning, pattern recognition, abstraction, algorithmic thinking, and decomposition. The essence of CT lies in its ability to enable learners to approach complex problems systematically, making it a vital skill in computer science and everyday life. This approach shows how computational thinking connects abstract ideas with practical problem-solving, useful in many areas like math, science, and even English-language arts.

In kindergarten, the CSTA K-12 Standards focus on basic concepts such as understanding what computers are and simple sequencing activities, often using "unplugged" lessons that don't require technology. By 8th grade, the standards advance to more complex topics, such as designing and analyzing algorithms, exploring the impacts of computing on society, and beginning to write code. These standards are designed to be age-appropriate, ensuring that students build a strong foundation in computer science through engaging and accessible activities.

[2] K–12 Computer Science Framework. (2016). Pages 13-14. Retrieved from http://www.k12cs.org.
[3] K–12 Computer Science Framework. (2016). Page 67. Retrieved from http://www.k12cs.org.
[4] Wing, J. M. (2006). Computational Thinking. Communications of the ACM, 49(3), 33-35. Retrieved from https://dl.acm.org/doi/10.1145/1118178.1118215

The SD Computer Science Standards will overlap slightly with the South Dakota Technology Standards, see *Figure 1* which was adopted from the Connecticut State Department of Education. The difference is that the SD Technology Standards focus more on using computers appropriately and safely. In contrast, computer science standards address how computers work and how to build the different aspects (software, internet, hardware, etc.) of technology. Districts will have the flexibility to determine how their teachers and various programs can best address the differences in the standards. The core concept of computational thinking overlaps both sets of standards and serves as an accessible framework for problem-solving in any subject area.

Figure 1



One goal of the standards is to explicitly indicate where artificial intelligence (AI) is referenced or where concepts and skills build into an understanding of AI. Even though AI has developed since its inception in the 1950s, it hasn't been until the last ten to fifteen years that AI has been woven into technologies, like recommendations in Netflix or Amazon Prime, that commonly interface directly with users. Then, with the release of OpenAI's large language model, ChatGPT, the need to better understand AI and its implications became more urgent. AI isn't simply a separate entity from computer science; it's more like an evolving branch woven into the field. It leverages core computer science principles like algorithms, data structures, and computational theory to design intelligent systems capable of learning, adapting, and making decisions without explicit programming. As AI advances, it will undoubtedly influence even more areas of computer science, blurring the lines between traditional programming and intelligent automation.[5] The AI4K12 framework[6], developed by the Association for the Advancement of Artificial Intelligence (AAAI) and the Computer Science Teachers Association (CSTA), inspired South Dakota's AI Core Ideas by providing a structured approach to integrating AI concepts into the K-8 SD SC standards. Thus, by incorporating the "AI Big Ideas," we want to ensure that students are not only technologically adept but also socially aware of the implications of AI. Standards that address the social impact of computer science and AI bring in the critical aspect of keeping "humans in the AI loop." That phrase emphasizes the role of human oversight, decision-making, and collaboration in AI systems to ensure effective and safe outcomes.

Students can learn how AI works through standards that balance theoretical understanding and practical application while adhering to individual district policies on AI use. These standards lay the foundation for introducing fundamental concepts of AI, such as machine learning and data analysis, through age-appropriate topics. Educators ensure that lessons comply with district guidelines by incorporating safe AI practices, emphasizing responsible use, and fostering critical thinking about AI's impact on society. This approach allows students to gain valuable insights into AI technology while respecting local educational policies and software license agreements. As AI continues to advance and permeate our daily lives, a solid foundation in AI literacy is essential in shaping a future generation that is well-informed, critical, and capable of leveraging AI for societal betterment. The other realm of district control is how these standards will be incorporated into students' learning. Each district will be unique, whether these standards are addressed by K-5 classroom teachers, library media specialists, or whatever other scenario works best for that district. Since the standards are divided into grade bands, or ranges, there is even more flexibility with what can be covered at each grade level.

[5] Rice University. (n.d.). Computer Science vs. Artificial Intelligence and Machine Learning. Retrieved from https://csweb.rice.edu/academics/graduate-programs/online-mcs/blog/computer-science-vsartificial-intelligence-and-machine-learning.
[6] AI4K12. (2020). Five Big Ideas in Artificial Intelligence. Retrieved from https://ai4k12.org/resources/big-ideas-poster/

Implementing the K-8 SD Computer Science Standards can be both challenging and rewarding. These standards are designed to equip students with the skills and knowledge necessary to thrive in a technology-driven world. While this may present new challenges for some educators, it is an invaluable opportunity to guide students toward becoming informed users and creators of technology. The resources and support available will aid in this transition, ensuring that teachers are well-prepared to integrate these standards effectively. Embracing this initiative and modeling the learning process alongside students fosters an environment of collaboration, continuous learning, and innovation. This prepares students for a future where they are ready to drive technological advancements and contribute meaningfully to society.

Computer Science Standards Committee

Name	**Employer**	**Position**
Alexandra Davison	Yankton School District	Elementary Technology Integrationist
Ben Sayler	Center for the Advancement of Math and Science Education - Black Hills State University	Director
Benjamin Benson	Sanford Research	Senior Research Education Programs Specialist
Carrie Henrickson	Ethan School District	PreK-12 Computer Educator
Carrie Tobin	Plankinton Elementary School	Elementary Computers/Title Math
Deana Castro	Rapid City Area Schools	IT Application Analyst
Fenecia Homan	Governors Cyber Academy - Dakota State University	Dean
Joy Lundgren	Rapid City Area School District	5th grade teacher
Laura Shumaker	Howard School District	Technology Teacher; 6-12
Marcia Torgrude	Rapid City Catholic School System	MS/HS Computer Science Instructor
Monica Pickard	Howard School District	Elementary Teacher
Sadia Risse	Bennett County Middle School	STEAM Teacher
Scott Binde	Tea Area School District	Student Success Coach
Scott Headrick	Flandreau School District	Computer Science Teacher; Assistant Technology Coordinator
Tamara Schmidt	Brookings School District	4th Grade math, science & tech teacher
Support Team		
Nicole Uhre-Balk	Compass	Learning Specialist/Committee Facilitator
Vera Van Winkle	Department of Education	DOE Project Lead

Key Concepts and Standards Overview

Computer Science
Computer science is the study of computers and algorithmic processes, including their principles, their hardware and software designs, their implementation, and their impact on society.

Concepts
Concepts in the CSTA K-12 computer science Standards refer to the fundamental areas of knowledge within computer science. These include the core principles and knowledge students need to understand to build a solid foundation in the field.

- **Computing Systems**
  Computing Systems involves the study of hardware and software that form the backbone of computer operations. This includes understanding how computers process information, the components that make up computing devices, and how software and hardware interact.

- **Networks and the Internet**
  Networks and the Internet cover the principles and structures of computer networks, including how data is transmitted and received over various types of networks. They also include the study of the Internet, its architecture, and the protocols that enable global communication.

- **Data and Analysis**
  Data and Analysis focuses on how data is collected, organized, and interpreted. Students learn about data representation, data structures, and the methods used to analyze and draw meaningful conclusions from data sets, which are crucial for informed decision-making.

- **Algorithms and Programming**
  Algorithms and Programming involve studying step-by-step procedures and coding to solve problems. This area teaches students how to design, write, and debug programs and how to use algorithms to perform tasks efficiently and effectively.

- **Impacts of Computing**
  Impacts of Computing examine the broader effects of technology on society, including societal, cultural, and economic implications. This concept encourages students to consider how computing influences the world and to develop responsible practices in using and developing technology.

Practices

The seven core computer science practices help students develop the skills and ways of thinking needed to succeed in our data-rich, connected world. These practices work together and follow a process for creating technology projects. This process starts with understanding different users and valuing their perspectives and ends with sharing results with a wide audience. Unlike other parts of the standards, these practices are not split by grade level but describe how students should improve from kindergarten through 12th grade.

Computational Thinking

Computational thinking (CT) is a problem-solving process involving designing solutions that capitalize on the power of computers and is essentially a core human ability. Any practice that has an asterisk (*) indicates that the practice relates to computational thinking.

1. **Creating accessible computing environments**
   Understanding personal, social, and educational contexts and considering the needs of a wide range of users during the design process is essential for developing accessible and beneficial computational products. Strategic incorporation of various perspectives creates learning environments poised for creation, idea-sharing, and collaboration.

2. **Collaborating around computing**
   Collaborative computing involves working in pairs or teams to complete a task. By asking for others' input and feedback, collaboration often leads to better results than working alone. It requires managing different perspectives, ideas, skills, and personalities. Students should use tools to help them work together effectively and create complex projects.

3. **Recognizing and defining computational problems***
   The ability to identify when and how to use computing is a skill that develops over time and is crucial in computing. Solving a problem with a computational approach involves defining the problem, breaking it down into parts, and assessing if a computational solution is suitable for each part.

4. **Developing and using abstractions, decomposition, pattern recognition, and algorithmic thinking***
   Developing and using abstractions, decomposition, pattern recognition, and algorithmic thinking are essential skills in computing. Abstraction involves simplifying complex systems by focusing on the main ideas. Decomposition means breaking down a problem into smaller, manageable parts. Pattern recognition helps identify similarities that can simplify problem-solving. Algorithmic thinking involves creating step-by-step instructions to solve problems efficiently. Together, these skills help students tackle complex tasks more effectively.

5. **Creating computational artifacts***
   Creating computational artifacts involves both creativity and problem-solving to develop prototypes, early models used to test and refine ideas, and solutions. Students make artifacts that are meaningful to them or helpful to their community. Computational artifacts, which are digital creations made using computing tools, can be made by modifying or creating new ones. Examples include programs, simulations, visualizations, digital animations, robotic systems, and apps.

6. **Testing and refining computational artifacts***
   Testing and refinement involve improving a computational project through repeated trials. This includes finding and fixing errors and ensuring the results match the goals. Students also adjust their work based on user feedback to enhance performance, reliability, usability, and accessibility.

7. **Communicating about computing**
   Communication in computer science involves expressing ideas and sharing them with others. Students explain how and why they use computation and its effects. They write clear comments, document their work, and use various media to share their ideas. Effective communication includes using precise language and thinking about the audience.

AI Core Ideas

The AI Core Ideas are key concepts designed to help students understand how artificial intelligence (AI) works within computer science. By exploring these concepts, students gain insights into how AI systems function, learn, and interact, preparing them to navigate and contribute to ever-advancing AI.

Artificial Intelligence

AI is a broad scientific field that encompasses the development of systems that can exhibit behaviors such as learning and decision-making. This encompasses tasks like logical reasoning, understanding language, and solving complex problems. “Artificial intelligence is based on the use of machine learning algorithms and technologies to give machines the ability to apply certain cognitive abilities and perform tasks on their own autonomously or semi-autonomously.”[7]

1. **Perception:** Computers interpret the world through various forms of data collection, including sensors, user inputs, and other databases. AI's notable success lies in advancing these machines to reliably process and analyze this diverse data, turning it into meaningful insights.

2. **Representation & Reasoning:** AI agents use data to construct the world into models that facilitate reasoning, addressing a core challenge of intelligence. Despite their ability to process complex issues, these agents' reasoning differs fundamentally from human thought processes.

3. **Learning:** Machine learning enables computers to discern patterns and gain knowledge from vast datasets. When paired with learning algorithms, this method has driven substantial advancements in AI, though it often depends on human-provided or machine-acquired "training data."

4. **Natural Interaction:** Intelligent agents are progressing towards natural human interaction, using advances in deep learning to understand and respond to speech, cultural context, and non-verbal cues.

5. **Societal Impact:** AI is transforming various facets of life, offering benefits in work, communication, and care, yet societal considerations are crucial to address potential issues and ensure service for all.

[7] Morandín-Ahuerma, F. (2022). What is Artificial Intelligence? *International Journal of Research Publication and Reviews, 3*(12), 1947-1951. Retrieved from www.ijrpr.com

How to Read the Standards

K-2.CS.02	Use appropriate terminology to identify simple hardware and software problems and apply strategies for solving these problems.

K-2 = Grade-band, **CS** = Computing Systems concept, **02** = standard number

Concepts & Subconcepts	Practices	AI Core Ideas
Computing Systems • Devices • Hardware & Software • Troubleshooting Networks and the Internet • Network Communication and Organization • Cybersecurity Data and Analysis • Storage • Collection, Visualization, and Transformation • Inference and Models Algorithms and Programming • Algorithms • Variables • Control • Modularity • Program Development Impacts of Computing • Culture • Social Interactions • Safety & Regulation	1. Creating accessible computing environments 2. Collaborating around computing **3. Recognizing and defining computational problems*** **4. Developing and using abstractions, decomposition, pattern recognition, and algorithmic thinking*** **5. Creating computational artifacts*** **6. Testing and refining computational artifacts*** 7. Communicating about computing ***Practices that refer to computational thinking**	1. Perception 2. Representation & Reasoning 3. Learning 4. Natural Interaction 5. Societal Impact

Grades K-2 Computer Science Standards

Computing Systems

Identifier	Standard	Subconcept	Practice Number	AI Big Idea
K-2.CS.01	Use suitable hardware/software to perform different tasks while recognizing individual's needs and preferences.	Devices, Hardware & Software	1, 7	1. Perception
K-2.CS.02	Use appropriate terminology to identify simple hardware and software problems and apply strategies for solving these problems.	Troubleshooting	6, 7	

Networks and the Internet

Identifier	Standard	Subconcept	Practice Number	AI Big Idea
K-2.NI.01	Explain what a password is, as well as its purpose.	Cybersecurity	7	

Data and Analysis

Identifier	Standard	Subconcept	Practice Number	AI Big Idea
K-2.DA.01	With guidance, draw conclusions, make predictions, and interpret data based on picture graphs or patterns with or without a computing device.	Inference & Models	4	2. Representation & Reasoning 3. Learning

Algorithms and Programming

Identifier	Standard	Subconcept	Practice Number	AI Big Idea
K-2.AP.01	Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks (with or without devices).	Algorithms	4	3. Learning
K-2.AP.02	Use problem-solving steps to break down a problem into smaller parts to identify patterns and possible solutions.	Variables, Control, Modularity	4, 5, 3	3. Learning
K-2.AP.03	Use the appropriate terms to explain the process of finding and correcting mistakes in an algorithm or program (ex. sequences, looping, bugs, debugging, variables).	Program Development	6, 7	2. Representation & Reasoning 5. Societal Impact
K-2.AP.04	Describe how computer systems can be designed to interact with humans.	Program Development	1, 5, 6, 7	3. Learning 4. Natural Interaction

Impacts of Computing

Identifier	Standard	Subconcept	Practice Number	AI Big Idea
K-2.IC.01	Describe past, present, and possible future impacts of technology on society.	Culture	7	5. Societal Impact
K-2.IC.02	Identify and explain how technologies are used in daily life, school, and the workforce.	Culture, Social Interactions, Safety & Regulation	1, 7	5. Societal Impact
K-2.IC.03	Describe the concept of a digital footprint and how to create online connections to maintain safety and balance with offline relationships.	Safety & Regulation	7	5. Societal Impact

Grades 3-5 Computer Science Standards

Concepts & Subconcepts	Practices	AI Core Ideas
Computing Systems ● Devices ● Hardware & Software ● Troubleshooting Networks and the Internet ● Network Communication and Organization ● Cybersecurity Data and Analysis ● Storage ● Collection, Visualization, and Transformation ● Inference and Models Algorithms and Programming ● Algorithms ● Variables ● Control ● Modularity ● Program Development Impacts of Computing ● Culture ● Social Interactions ● Safety & Regulation	1. Creating accessible computing environments 2. Collaborating around computing **3. Recognizing and defining computational problems*** **4. Developing and using abstractions, decomposition, pattern recognition, and algorithmic thinking*** **5. Creating computational artifacts*** **6. Testing and refining computational artifacts*** 7. Communicating about computing ***Practices that refer to computational thinking**	1. Perception 2. Representation & Reasoning 3. Learning 4. Natural Interaction 5. Societal Impact

Computing Systems

Identifier	Standard	Subconcept	Practice Number	AI Big Idea
3-5.CS.01	Describe how computing devices' internal and external parts function to form a system.	Devices	7	
3-5.CS.02	Model how hardware and software work together as a system to accomplish tasks.	Hardware & Software	4	1. Perception 2. Representation & Reasoning
3-5.CS.03	Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies.	Troubleshooting	6	

Networks and the Internet

Identifier	Standard	Subconcept	Practice Number	AI Big Idea
3-5.NI.01	Model how information is sent and received over a network.	Network Communication & Organization	4	
3-5.NI.02	Discuss real-world cybersecurity problems and how personal information can be protected.	Cybersecurity	3	2. Representation & Reasoning 3. Learning 5. Societal Impacts

Data and Analysis

Identifier	Standard	Subconcept	Practice Number	AI Big Idea
3-5.DA.01	Organize and visually present collected data to highlight relationships and support a claim.	Collection Visualization & Transformation	7	
3-5.DA.02	Use data to highlight patterns, propose cause-and-effect relationships, predict outcomes, or communicate an idea.	Inference & Models	3, 4, 7	2. Representation and Reasoning 3. Learning

Algorithms and Programming

Identifier	Standard	Subconcept	Practice Number	AI Big Idea
3-5.AP.01	Compare and refine multiple algorithms for the same task and determine which is the most appropriate.	Algorithms	3, 4, 6	3. Learning
3-5.AP.02	Create programs that use variables to store and modify data.	Variables	5	2. Representation and Reasoning
3-5.AP.03	Create programs that include sequences, events, loops, and conditionals.	Control	4, 5	
3-5.AP.04	Decompose (break down) problems into subproblems to facilitate the program development process, including incorporating portions of existing programs and proper documentation.	Modularity, Program Development	3, 5, 7	
3-5.AP.05	Discuss how voice assistants use algorithms to understand and respond to spoken commands, as well as factors that contribute to different responses.	Algorithms, Variables, Program Development	4, 5, 6, 7	4. Natural Interaction

3-5.AP.06	Use the iterative process to test and debug a program or algorithm to ensure it runs as intended.	Program Development	4, 6	2. Representation and Reasoning
3-5.AP.07	Collaborate with peers during all stages of program development.	Program Development	2	
3-5.AP.08	Describe choices made during program development using code comments, presentations, and demonstrations.	Program Development	7	

Impacts of Computing

Identifier	Standard	Subconcept	Practice Number	AI Big Idea
3-5.IC.01	Discuss computing technologies that have changed South Dakota and the world, and express how those technologies influence, and are influenced by, cultural practices.	Culture	3	5. Societal Impact
3-5.IC.02	Identify limitations and possible solutions relating to accessibility and usability of technology products for the varied needs and wants of users.	Culture	1	4. Natural Interaction 5. Societal Impact
3-5.IC.03	Collect and utilize varied perspectives to improve computational artifacts.	Social Interactions	1	5. Societal Impact
3-5.IC.04	Explain societal issues that relate to technology and describe the consequences of inappropriate use.	Safety & Regulation	7	5. Societal Impact

Grades 6-8 Computer Science Standards

<table>
<tr><th>Concepts & Subconcepts</th><th>Practices</th><th>AI Core Ideas</th></tr>
<tr><td>Computing Systems
Devices
Hardware & Software
Troubleshooting
Networks and the Internet
Network Communication and Organization
Cybersecurity
Data and Analysis
Storage
Collection, Visualization, and Transformation
Inference and Models
Algorithms and Programming
Algorithms
Variables
Control
Modularity
Program Development
Impacts of Computing
Culture
Social Interactions
Safety & Regulation</td><td>Creating accessible computing environments
Collaborating around computing
Recognizing and defining computational problems*
Developing and using abstractions, decomposition, pattern recognition, and algorithmic thinking*
Creating computational artifacts*
Testing and refining computational artifacts*
Communicating about computing
*Practices that refer to computational thinking</td><td>1. Perception
2. Representation & Reasoning
3. Learning
4. Natural Interaction
5. Societal Impact</td></tr>
</table>

Computing Systems

Identifier	Standard	Subconcept	Practice Number	AI Big Idea
6-8.CS.01	Recommend improvements to the design of computing devices, based on an analysis of how individuals interact with the devices.	Devices	3	2. Representation & Reasoning 3. Learning 4. Natural Interaction
6-8.CS.02	Design projects that combine hardware and software components to collect and exchange data.	Hardware & Software	5	1. Perception 2. Representation & Reasoning
6-8.CS.03	Systematically identify and fix problems with computing devices and their components.	Troubleshooting	6	

Networks and the Internet

Identifier	Standard	Subconcept	Practice Number	AI Big Idea
6-8.NI.01	Simulate the flow of information as packets on the Internet and networks.	Network communication & Organization	4	
6-8.NI.02	Explain how physical and digital security measures protect electronic information.	Cybersecurity	7	2. Representation & Reasoning 5. Societal Impacts
6-8.NI.03	Apply multiple methods of encryption to demonstrate how to transmit information securely.	Cybersecurity	4	2. Representation & Reasoning 5. Societal Impacts

Data and Analysis

Identifier	Standard	Subconcept	Practice Number	AI Big Idea
6-8.DA.01	Represent data using multiple encoding schemes.	Storage	4	2. Representation & Reasoning
6-8.DA.02	Collect data using computational tools and transform the data to make it more useful and reliable.	Collection Visualization & Transformation	6	
6-8.DA.03	Refine computational models based on the data they have generated.	Inference & Models	5, 4	

Algorithms and Programming

Identifier	Standard	Subconcept	Practice Number	AI Big Idea
6-8.AP.01	Use flowcharts and/or pseudocode to address complex problems as algorithms.	Algorithms	4, 4	
6-8.AP.02	Create clearly named variables that represent different data types and perform operations on their values.	Variables	5, 5	2. Representation & Reasoning
6-8.AP.03	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	Control	5, 5	
6-8.AP.04	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	Modularity	3	
6-8.AP.05	Create procedures with parameters to organize code and make it easier to reuse.	Modularity	4, 4	

6-8.AP.06	Seek and incorporate feedback from team members and users to refine a solution that meets user needs.	Program Development	1, 2	
6-8.AP.07	Incorporate existing code, media, and libraries into original programs, and give attribution.	Program Development	4, 5, 7	
6-8.AP.08	Systematically test and refine programs using a range of test cases.	Program Development	6	
6-8.AP.09	Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.	Program Development	2	
6-8.AP.10	Document programs in order to make them easier to follow, test, and debug.	Program Development	7	

Impacts of Computing

Identifier	Standard	Subconcept	Practice Number	AI Big Idea
6-8.IC.01	Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options in South Dakota and the world, as well as urban, rural, and reservation communities.	Culture, Social Interactions, Safety & Regulation	1, 2, 3, 7	5. Societal Impacts
6-8.IC.02	Discuss issues of accessibility in the design of existing technologies.	Culture	1.2	5. Societal Impacts
6-8.IC.03	Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.	Social Interactions	2, 5	
6-8.IC.04	Describe tradeoffs between allowing information to be public and keeping information private and secure.	Safety & Regulation	7	5. Societal Impacts