Arkansas Computer Science and Computing Standards

High School Cybersecurity

2020

Arkansas Computer Science and Computing Standards for High School Cybersecurity

Introduction

The Arkansas Computer Science and Computing Initiative standards for high school courses are designed to provide understandings of concepts in computer science that are necessary for students to function in an ever-changing technological world. Through these standards, students will explore, apply, and move toward mastery in skills and concepts related to Computational Thinking and Problem Solving; Data, Information, and Security; Algorithms and Programs; Computers and Communications; and Professionalism and Impacts of Computing. These standards help students learn to accomplish tasks and solve problems independently and collaboratively. These standards give students the tools and skills needed to be successful in college and careers including computer science, computing, and other fields.

State developed pathways within the Arkansas Computer Science and Computing Initiative all begin with common year-one standards which allow for consistency across the state and all schools. These common standards address the basic knowledge and skills needed for any student entering a technology-based field.

The course standards have been grouped into one-credit (typically yearly) standards to afford the classroom educator additional flexibility in their curriculum choices; however, the course codes remain based on one-half credit (typically semester). Each state-developed pathway will have three credits (six pathway specific course codes) worth of Computer Science Flex Credit (465XXX) course codes.

The Arkansas State Board of Education (SBE) does not place any prerequisites on the Arkansas Computer Science and Computing Initiative high school courses, but allows for schools to place students in any of the courses based on ability and desire. The Arkansas Department of Education (ADE) recommends that districts develop and formally adopt a written policy outlining placement protocols. Evaluation tools and placement criteria will be the responsibility of the local districts.

The SBE and ADE authorize schools to enroll students across levels in the same sections of the master schedule (a.k.a. stacking) as long as the number of students does not exceed Standards of Accreditation maximums and/or ratios and the school can reasonably assure a high-quality educational experience for all students within that section.

Implementation of the Arkansas Computer Science Standards for High School Cybersecurity begins during the 2021-2022 school year.

Course Title: Cybersecurity
Course/Unit Credit: 1 credit per listed course code

	Cybersecurity Year 1	**Cybersecurity Year 2**	**Cybersecurity Year 3 - Advanced**
Cybersecurity	465270	465280	465290

Teacher Licensure: Please refer to the Course Code Management System (https://adedata.arkansas.gov/ccms/) for the most current licensure codes.
Grades: 9-12
Prerequisites: There are no ADE established course prerequisites for any of the Arkansas Computer Science and Computing Initiative high school courses; it is up to the local district to determine placement based on student ability.

Computer Science and Computing Practices

Students exhibit proficiency in computer science and computing through:

Communication - Students effectively communicate, using accurate and appropriate terminology, when explaining the task completion or problem solving strategies used. They recognize that creating good documentation is an ongoing and important part of the communication process.

Collaboration - Students productively work with others while ensuring multiple voices are heard and considered. They understand that diverse thoughts may lead to creative solutions and that some problems may be best solved collaboratively.

Storytelling - Students creatively combine multimedia tools, such as graphics, animations, and videos with research, writing, and oral presentations to create ethical, data-driven stories.

Professionalism - Students embrace professionalism by demonstrating skills and behaviors necessary for success in technical careers.

Ethics and Impact - Students comprehend the ramifications of actions prior to taking them. They are aware of their own digital and cyber presence and its impact on other individuals and society.

Inclusion - Students encourage diversity in the field of computer science and computing regardless of race, ethnicity, gender, or other differences.

Learning by Failure - Students reflect upon and critique their work while embracing a willingness to seek feedback and constructive instruction from teachers and peers. They utilize the feedback to continually improve current projects, educational experiences, knowledge, and confidence.

Perseverance - Students expect difficulties and persist in overcoming challenges that occur when completing tasks. They recognize making and correcting mistakes is necessary for the learning process while problem solving.

Understanding - Students recognize patterns, utilize tools, and apply problem solving strategies to build understanding, find solutions, and successfully deliver high-quality work.

Patterns - Students understand and utilize the logical structure of information through identifying patterns and creating conceptual models. They decompose complex problems into simpler modules and patterns.

Problem Solving - Students exhibit proficiency through the process of identifying and systematically solving problems. They recognize problem solving is an ongoing process.

Research - Students purposefully gather information and seek to expand their knowledge through various methods and mediums. They embrace the practice of gaining knowledge to develop novel approaches for solving problems and addressing issues they have not previously encountered, in addition to merely searching for answers.

Tools - Students evaluate and select tools to be used when completing tasks and solving problems. They understand that appropriate tools may include, but are not limited to, their mind, pencil and paper, manipulatives, software applications, programming languages, or appropriate computing devices.

Arkansas Computer Science and Computing Standards for High School Cybersecurity

Strand	Content Cluster
Computational Thinking and Problem Solving	
	1. Students will analyze and utilize problem-solving strategies.
	2. Students will analyze and utilize connections between concepts of mathematics and computer science.
Data, Information, and Security	
	3. Students will analyze and utilize data through the use of computing devices.
	4. Students will analyze and utilize concepts of cybersecurity.
Algorithms and Programs	
	5. Students will create, evaluate, and modify algorithms.
	6. Students will create programs to solve problems.
Computers and Communications	
	7. Students will analyze the utilization of computers within industry.
	8. Students will analyze communication methods and systems used to transmit information among computing devices.
	9. Students will utilize appropriate hardware and software.
Professionalism and Impacts of Computing	
	10. Students will analyze the impacts of technology and professionalism within the computing community.
	11. Students will demonstrate understanding of storytelling with data and appropriately communicate about technical information.

Understanding the Arkansas Computer Science and Computing Standards Documents:

- This Arkansas Department of Education curriculum standards document is intended to assist in district curriculum development, unit design, and to provide a uniform, comprehensive guide for instruction.
- The goal for each student is proficiency in all academic standards for the course/year in which the student is enrolled.
- The Practice Standards are intended to be habits of mind for all students and were written broadly in order to apply to all grades/levels. The Practice Standards are not content standards and are not intended to be formally assessed.
- Notes (NOTE:) and examples given (e.g.,) found within the document are not mandated by the Arkansas State Board of Education, but are provided for clarification of the standards by the Arkansas Department of Education and/or the standards drafting committee. The notes and examples given are subject to change as understandings of the standards evolve.
- Within the high school documents, the numbering for standards is read as: Course Abbreviation - Year - Content Cluster - Standard. Example: “CSPG.Y1.2.3” would be Computer Science Programming - Year 1 - Content Cluster 2 - Standard 3.
- Within the Coding Block document, the numbering for standards is read as: Course Abbreviation - Content Cluster - Standard. Example: “CSCB.1.2” would be Coding Block, Content Cluster 1, Standard 2.
- Within the K-8 Computer Science Standards documents, the numbering for standards is read as: Course Abbreviation - Grade - Content Cluster - Standard. Example: “CSK8.G1.2.3” would be K-8, Grade 1, Content Cluster 2, Standard 3.
- Ancillary documents and supporting information may be released to assist in further understanding of the standards with possible classroom implementation strategies included.

“Research” and Learning

The Arkansas Department of Education Office of Computer Science recognizes that the use of the term “research” as an action verb within academic standards is not mainstream, though not unheard of, and exists as a measurable objective within other Arkansas K-12 academic standards. The members of the internal team, composed of the State Director of Computer Science and nine state-wide Computer Science Specialists, discussed this at length amongst ourselves and with many committee members. While there existed varying opinions for various reasons, the internal team opted to keep “research” as an action verb within the standards for the following reasons:

1. The internal team believes that this use of “research” and the skill-building activities students will undertake while performing said research will produce students that have a skillset which industry representatives have identified as missing from workers entering technical job fields.
2. As the field of Computer Science and Computing is ever changing and growing, professionals and students within this field must conduct informal research on an almost daily basis to maintain relevant knowledge and skills.
3. The use of “research” within this document does not determine classroom implementation; however, it is used to indicate that the student should take individual and active efforts to seek out knowledge to develop novel approaches for solving problems and addressing issues they have not previously encountered, in addition to merely searching for answers.
4. The use of “research” should not infer that a student should be required to do an extensive qualitative or quantitative research project from the use of “research” anywhere in this document; however, a more formal research project is not prohibited if the teacher feels it is appropriate.

Strand: Computational Thinking and Problem Solving

Content Cluster 1: Students will analyze and utilize problem-solving strategies.

Year 1	Year 2	Year 3 - Advanced
CSCS.Y1.1.1 Leverage problem-solving strategies to solve problems of level-appropriate complexity	CSCS.Y2.1.1 Leverage problem-solving strategies to solve problems of level-appropriate complexity CSCS Y2: Extend problem-solving strategies to include an understanding of adversarial thinking	CSCS.Y3.1.1 Leverage adversarial thinking and risk concepts to solve complex cybersecurity problems
NOTE: Problem-solving strategies that encompass computational thinking include, but are not limited to, abstraction, algorithm development, decomposition, and pattern recognition. NOTE CSCS Y2-Y3: Problem-solving strategies may include, but are not limited to modeling and decomposing system attacks. Adversarial thinking includes, but is not limited to, analysis of problems an attacker faces when determining ways to access, damage, or disrupt a computer system.		
CSCS.Y1.1.2 Analyze and utilize multiple representations of problem-solving logic used to solve problems of appropriate complexity	CSCS.Y2.1.2 Analyze and utilize multiple representations of problem-solving logic used to solve problems of appropriate complexity	CSCS.Y3.1.2 Explore and demonstrate tactics adversaries use to respond to system defenses to accomplish an objective
NOTE: Representations may include, but are not limited to, backlog, decision matrix, design brief, documentation, fault tree analysis, flowchart, pseudocode, and sprints.		
CSCS.Y1.1.3 Analyze and utilize collaborative methods in problem solving of level-appropriate complexity	CSCS.Y2.1.3 Analyze and utilize collaborative methods in problem solving of level-appropriate complexity	CSCS.Y3.1.3 Explore and utilize level-appropriate collaborative methods used to operate an organization at various scales (e.g., local, regional, national, global)
NOTE: Collaborative methods may include, but are not limited to, distributive (divide and conquer), paired programming, and redundant parallel. NOTE CSCS Y2-Y3: Essential collaboration skills include, but are not limited to, reading and writing documentation. Collaborative problem-solving includes, but is not limited to, contributing to open-source software. Collaborative tools and methods may include, but are not limited to, cloud technologies (e.g., Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS)), security information and event management systems, software-defined networking, ticket tracking systems, and version control systems.		

CSCS.Y1.1.4 Analyze and utilize level-appropriate troubleshooting strategies for hardware and software	CSCS.Y2.1.4 Analyze and utilize level-appropriate troubleshooting strategies for hardware and software	CSCS.Y3.1.4 Research and implement forensic investigation and intrusion detection techniques to detect adversarial behavior
NOTE CSCS Y2-Y3: Troubleshooting strategies for software includes, but is not limited to, reverse engineering. Detecting adversarial behavior techniques may include, but is not limited to, threat hunting and basic troubleshooting steps triggering cybersecurity incident response scenarios.		

Strand: Computational Thinking and Problem Solving

Content Cluster 2: Students will analyze and utilize connections between concepts of mathematics and computer science.

<table>
<tr><th>Year 1</th><th>Year 2</th><th>Year 3 - Advanced</th></tr>
<tr><td>CSCS.Y1.2.1
Interpret relational and logical expressions of level-appropriate complexity using comparison and Boolean operators</td><td>CSCS.Y2.2.1
Interpret compound expressions using multiple relational and logical operators</td><td><i>Continuation of this standard is not specifically included or excluded</i></td></tr>
<tr><td colspan="3">NOTE:
Boolean operators include AND, OR, NOT, and XOR.
Comparison operators may include, but are not limited to, <, >, and !=.</td></tr>
<tr><td>CSCS.Y1.2.2
Classify the types of information that can be stored as variables and analyze the appropriateness of each (e.g., Booleans, characters, integers, floating points, strings)</td><td><i>Continuation of this standard is not specifically included or excluded</i></td><td><i>Continuation of this standard is not specifically included or excluded</i></td></tr>
<tr><td>CSCS.Y1.2.3
Analyze how computer science concepts relate to the field of mathematics</td><td>CSCS.Y2.2.3
Research and implement level-appropriate common cryptography algorithms and concepts such as random number generation and hashing functions</td><td><i>Continuation of this standard is not specifically included or excluded</i></td></tr>
<tr><td colspan="3">NOTE:
Concepts may include, but are not limited to, different division methods (e.g., integer, long, modular), random number generation, domain, maximum, mean, minimum, mode, and range.</td></tr>
<tr><td>CSCS.Y1.2.4
Discuss and apply concepts of abstraction</td><td>CSCS.Y2.2.4
Analyze and utilize concepts of abstraction as modeling and abstraction as encapsulation</td><td><i>Continuation of this standard is not specifically included or excluded</i></td></tr>
<tr><td colspan="3">NOTE:
Abstraction is the process of reducing information and detail to facilitate focus on relevant concepts and functionality (displaying only essential information while hiding the details).</td></tr>
</table>

CSCS.Y1.2.5 Perform operations of level-appropriate complexity with binary, decimal, and hexadecimal numbers	CSCS.Y2.2.5 Perform operations of level-appropriate complexity with binary, octal, decimal, and hexadecimal numbers CSCS Y2: Perform data encoding and decoding operations between various encoding formats (e.g., American Standard Code for Information Interchange (ASCII), Base64, Unicode Transformation Format - 8 Bit (UTF-8))	*Continuation of this standard is not specifically included or excluded*
NOTE: Operations may include, but are not limited to, addition, subtraction, multiplication, division, and conversion.		
CSCS.Y1.2.6 Demonstrate operator precedence in expressions and statements	CSCS.Y2.2.6 Interpret the security impacts of misinterpreting or misunderstanding proper application of the order of operations	*Continuation of this standard is not specifically included or excluded*
NOTE: Operators include, but are not limited to, addition, subtraction, division, modulus division, concatenation, square root, and exponentiation. Operator precedence may include, but is not limited to, inside-out, order of operations, and the understanding that the assignment statement of "x = 1" is not the same as "1 = x."		
This standard is not specifically required until Year 2	CSCS.Y2.2.7 Explore classical and modern uses of steganography	CSCS.Y3.2.7 Demonstrate the use of steganography in a program or a digital file (e.g., audio, document, image, video)

Strand: Data, Information, and Security

Content Cluster 3: Students will analyze and utilize data through the use of computing devices.

<table>
<tr><th>Year 1</th><th>Year 2</th><th>Year 3 - Advanced</th></tr>
<tr><td>CSCS.Y1.3.1
Define, store, access, and manipulate level-appropriate data (e.g., primitive, linear)</td><td>CSCS.Y2.3.1
Create programs to store, access, and manipulate level-appropriate data (e.g., structured data, objects)</td><td><i>Continuation of this standard is not specifically included or excluded</i></td></tr>
<tr><td colspan="3">NOTE:
Primitive data may include, but is not limited to, Boolean, character, double, float, and integer.
Linear data may include, but is not limited to, arrays, lists, strings, and vectors.
Structured data may include, but is not limited to, arrays, classes, linked lists, maps, multidimensional arrays, and structs.
Objects may include, but are not limited to, constructors, data members, and methods.
Defining, storing, and accessing may include, but are not limited to, type declaration, variables, and modifiers (e.g., final, pass-by-value, pass-by-reference parameters, private, protected, public).
Manipulating data may include, but is not limited to, arranging (e.g., queuing, stacking), bit manipulation, casting, rearranging, and sorting.</td></tr>
<tr><td>CSCS.Y1.3.2
Define and discuss different examples of level-appropriate quantitative and qualitative data</td><td>CSCS.Y2.3.2
Define and discuss different examples of level-appropriate quantitative and qualitative data</td><td><i>Continuation of this standard is not specifically included or excluded</i></td></tr>
<tr><td><i>This standard is not specifically required until Year 2</i></td><td>CSCS.Y2.3.3
Research, discuss, and create level-appropriate programs to model and simulate probabilistic and real-world scenarios</td><td><i>Continuation of this standard is not specifically included or excluded</i></td></tr>
<tr><td colspan="3">NOTE:
Probabilistic scenarios may include, but are not limited to, flipping a coin, random walkers, and rolling dice.
Real-world scenarios may include, but are not limited to, city population and predator-prey.</td></tr>
<tr><td>CSCS.Y1.3.4
Analyze, utilize, and visually represent level-appropriate data</td><td>CSCS.Y2.3.4
Analyze, utilize, and visually represent level-appropriate static and dynamic data

CSCS Y2:
Utilize security event and incident management (SEIM) platforms (e.g., Elastic Stack, Graylog, Splunk) or network traffic analysis tools (e.g., NetworkMiner, Wireshark) and analyze their ability to graphically represent the data they collect</td><td>CSCS.Y3.3.4
Utilize SEIM platforms (e.g., Elastic Stack, Graylog, Splunk) or network traffic analysis tools (e.g., NetworkMiner, Wireshark) and analyze their ability to graphically represent the data they collect</td></tr>
</table>

NOTE: Visual representation tools may include, but are not limited to, analytics reports, graphical representations, programming language libraries, and spreadsheets. Dynamic data may include, but are not limited to, network traffic, real-time weather data, sensor statuses, stock market valuations, and system status.		
CSCS.Y1.3.5 Perform level-appropriate data analysis using computing tools	CSCS.Y2.3.5 Perform level-appropriate data analysis using computing tools	CSCS.Y3.3.5 Perform level-appropriate data analysis using computing tools
NOTE: Analysis may include, but is not limited to, maximum values, mean values, minimum values, ranges, and string comparisons. NOTE CSCS Y2-Y3: Data may include, but is not limited to, authentication request logs, network traffic logs, physical location logs, program execution logs, and web server and file server access logs. Analysis may include, but is not limited to, identifying users of a server based on uniquely identifying features (e.g., hash, internet protocol (IP) address) and cross-referencing those features with lists of malicious entities or threat intelligence services.		

Strand: Data, Information, and Security

Content Cluster 4: Students will analyze and utilize concepts of cybersecurity.

<table>
<tr><th>Year 1</th><th>Year 2</th><th>Year 3 - Advanced</th></tr>
<tr><td>CSCS.Y1.4.1
Identify the five pillars of cybersecurity and evaluate the relevance of each pillar to computer science concepts</td><td>CSCS.Y2.4.1
Apply the five pillars of cybersecurity as applicable to level-appropriate computer science concepts</td><td>CSCS.Y3.4.1
Research and describe the origins of Operational Security (OPSEC) programs and the role OPSEC plays in both offensive and defensive security programs</td></tr>
<tr><td colspan="3">NOTE:
Additional concepts and key terms of the five pillars of cybersecurity (confidentiality, integrity, availability, non-repudiation, and authentication) may include, but are not limited to, access control paradigms, accountability, authorization, least-privilege, and need-to-know.</td></tr>
<tr><td>CSCS.Y1.4.2
Research and describe different roles within the hacking community (e.g., white hat, black hat, gray hat hacking), including positive and negative motivations, significant impacts, and social stereotypes</td><td><i>Continuation of this standard is not specifically included or excluded</i></td><td>CSCS.Y3.4.2
Identify and research the various local and regional cybersecurity communities</td></tr>
<tr><td colspan="3">NOTE:
White hat hacking may include, but is not limited to, bug bounty programs and contracted penetration testing. A significant impact example may include, but is not limited to, Charlie Miller’s compromisation of Fiat Chrysler vehicles.
Black hat hacking may include, but is not limited to, the unauthorized processes of accessing systems to destroy, compromise, or steal data and deny access to services or systems. A significant impact example may include, but is not limited to, Behzad Mesri’s alleged theft of data from Home Box Office (HBO) and subsequent ransom demands.
Gray hat hacking may include, but is not limited to, unauthorized processes of accessing systems to report, correct, and draw attention to security vulnerabilities. A significant example of gray hat hacking is intentionally not included; students and teachers are encouraged to explore and discuss the nuances of “right versus wrong” and motivations within this community, including nation-state actions.

NOTE CSCS:
Research includes, but is not limited to, separating the knowledge of an action/task (e.g., hacking) from the results of this action/task.
Hacking may include, but is not limited to, leveraging the understanding (e.g., creator, outsider) of a system for novel or potentially unexpected outcomes.

NOTE CSCS Y3:
Research may include, but is not limited to, attending local cybersecurity meetings, conferences, capture the flag (CTF) events, or other educational and networking events.</td></tr>
<tr><td>CSCS.Y1.4.3
Research and describe the impacts of ransomware, trojans, viruses, and other malware</td><td>CSCS.Y2.4.3
Research and describe common attacks on hardware, software, and networks</td><td>CSCS.Y3.4.3
Recommend and implement level-appropriate mitigations to common attacks on hardware, software, and networks</td></tr>
</table>

NOTE:
Common hardware attacks may include, but are not limited to, clones, hardware trojans, and side-channel attacks.
Common software attacks may include, but are not limited to, buffer overflows, deployment errors, software bugs, and Structured Query Language (SQL) and command injection.
Common network attacks may include, but are not limited to, man-in-the-middle attacks, packet sniffing, protocol abuse, and spoofing of media access control (MAC) or internet protocol (IP) addresses.

NOTE CSCS Y2-Y3:
Researching malware includes, but is not limited to, understanding the different classes of malware (e.g., potentially unwanted programs, ransomware, rootkits, trojans, viruses, worms) and the reasoning for its application by an adversary.
Mitigations may include, but are not limited to, configuring file permissions, configuring host-based and network-based firewalls, and using encryption technology for network communications.
Mitigation strategies include, but are not limited to, reducing the potential vulnerabilities caused by social engineering of humans, which is an attack vector present in all systems including hardware, software, and/or networks.

CSCS.Y1.4.4 Explain implications related to identification and responsible reporting of a vulnerability versus exploitation	CSCS.Y2.4.4 Research and describe ethical and unethical methods of disclosing vulnerabilities and the concepts of agency, consent, and permission	*Continuation of this standard is not specifically included or excluded*
This standard is not specifically required until Year 2	CSCS.Y2.4.5 Identify the purposes, common processes, and desired and undesired outcomes of cybersecurity assessments	CSCS.Y3.4.5 Perform and document a level-appropriate cybersecurity assessment against an application or system

NOTE CSCS Y2-Y3:
Cybersecurity assessment examples may include, but are not limited to, full- or limited-scope penetration tests, hardware analysis, and source code audits.
Undesired outcomes include, but are not limited to, false senses of security that may result from improperly conducted assessments.

Strand: Algorithms and Programs

Content Cluster 5: Students will create, evaluate, and modify algorithms.

Year 1	Year 2	Year 3 - Advanced
CSCS.Y1.5.1 Design and implement level-appropriate algorithms that use iteration, selection, and sequence	CSCS.Y2.5.1 Design and implement level-appropriate algorithms that use iteration, recursion, selection, and sequence	CSCS.Y3.5.1 Design and implement algorithms that solve level-appropriate, student-identified problems
CSCS.Y1.5.2 Illustrate the flow of execution of algorithms in level-appropriate programs including branching and looping	CSCS.Y2.5.2 Illustrate the flow of execution of algorithms in level-appropriate programs including branching, looping, and function	*Continuation of this standard is not specifically included or excluded*
NOTE: Illustrations may include, but are not limited to, flowcharts and pseudocode. NOTE CSCS Y2: Illustration tools may include, but are not limited to, Ghidra, IDA Starter, Radare2, and x64dbg.		
CSCS.Y1.5.3 Evaluate the qualities of level-appropriate student-created and non-student-created algorithms	CSCS.Y2.5.3 Evaluate the qualities of level-appropriate student-created and non-student-created algorithms including classic search and sort algorithms	*Continuation of this standard is not specifically included or excluded*
NOTE: Evaluation tools may include, but are not limited to, code review and test cases. Qualities may include, but are not limited to, correctness, efficiency, exception handling, input/data/model validation, portability, readability, scalability, and usability.		
CSCS.Y1.5.4 Use a systematic approach to detect and resolve errors in a given algorithm	CSCS.Y2.5.4 Use a systematic approach to detect and resolve errors in a given algorithm	CSCS.Y3.5.4 Utilize a systematic approach to identify and mitigate common security errors in code (e.g., buffer overflows, cleartext password handling, input validation)

Strand: Algorithms and Programs

Content Cluster 6: Students will create programs to solve problems.

Year 1	Year 2	Year 3 - Advanced
CSCS.Y1.6.1 Create programs using procedures to solve problems of level-appropriate complexity	CSCS.Y2.6.1 Create programs to solve problems of level-appropriate complexity	CSCS.Y3.6.1 Create programs to solve problems of level-appropriate complexity that obtain data from external sources
NOTE: “Procedures” is considered interchangeable with “functions” for meeting this standard. Problems may include, but are not limited to, encoding, encryption, finding minimum/maximum values, identifying prime numbers, searching and sorting, and solving classic computer science tasks such as The Towers of Hanoi problem. NOTE CSCS Y3: External sources of data may include, but are not limited to, the automation of downloading files, web-based application programming interfaces (API), or web-scraping.		
CSCS.Y1.6.2 Discuss and apply best practices of program design and format (e.g., descriptive names, documentation, indentation, user experience design, whitespace)	CSCS.Y2.6.2 Discuss and apply best practices of program design and format (e.g., descriptive names, documentation, indentation, user experience design, whitespace) CSCS Y2: Discuss the vulnerabilities of not applying best practices of program design, format, and distribution	*Continuation of this standard is not specifically included or excluded*
NOTE CSCS Y2: Discussion may include, but is not limited to, how an adversary can leverage well documented code or executables with debugging symbols.		
CSCS.Y1.6.3 Determine the scope and state of variables declared in procedures and control structures over time	*Continuation of this standard is not specifically included or excluded*	*Continuation of this standard is not specifically included or excluded*
NOTE: “Procedures” is considered interchangeable with “functions” for meeting this standard.		
CSCS.Y1.6.4 Create programs of level-appropriate complexity that read from standard input, write to standard output, read from a file, write to a file, and append to a file	CSCS.Y2.6.4 Create programs that read from, write to, and append to a file of level-appropriate complexity that includes structured data	*Continuation of this standard is not specifically included or excluded*

NOTE: Standard input and output is platform-specific. Standard input and output on personal computers may include, but are not limited to, a keyboard and terminal. Standard input and output on mobile application devices may include, but are not limited to, touchscreen and speakers. Standard input and output on robots may include, but are not limited to, sensors and servos. Structured data refers to any representation of data which can be interpreted by an external or separate computing system including, but not limited to, comma-separated values (CSV), JavaScript Object Notation (JSON), Extensible Markup Language (XML), and other line-based text documents.		
CSCS.Y1.6.5 Use a systematic approach to detect logic, runtime, and syntax errors within a program	CSCS.Y2.6.5 Use a systematic approach to detect logic, runtime, and syntax errors within a program	CSCS.Y3.6.5 Use a systematic approach to detect logic, runtime, and syntax errors within a program
This standard is not specifically required until Year 2	CSCS.Y2.6.6 Perform operations that manipulate files using a hex editor	CSCS.Y3.6.6 Perform level-appropriate tasks that alter the execution of a program, subvert protections, or otherwise manipulate a file
NOTE CSCS Y2-Y3: Methods of alteration and subversion may include, but are not limited to, attacks that result in predetermined outcomes, code injection, or patching an executable.		

Strand: Computers and Communications

Content Cluster 7: Students will analyze the utilization of computers within industry.

Year 1	Year 2	Year 3 - Advanced
CSCS.Y1.7.1 Identify hardware and software specific to carrying out the mission of regional industries	CSCS.Y2.7.1 Utilize hardware and/or software to solve level-appropriate industry-based problems	*Continuation of this standard is not specifically included or excluded*
CSCS.Y1.7.2 Research advancing and emerging technologies (e.g., artificially intelligent agents, blockchain, extended reality, Internet of Things (IoT), machine learning, robotics)	CSCS.Y2.7.2 Research cutting-edge technology and its effects on the way business may be conducted in the future (e.g., blockchain, business responsibilities, eCommerce, entrepreneurship, payment methods, virtual currencies)	*Continuation of this standard is not specifically included or excluded*

Strand: Computers and Communications

Content Cluster 8: Students will analyze communication methods and systems used to transmit information among computing devices.

Year 1	Year 2	Year 3 - Advanced
CSCS.Y1.8.1 Utilize the command line to accomplish common network troubleshooting tasks at an introductory level	CSCS.Y2.8.1 Explain how information obtained from common network troubleshooting processes may be used for malicious purposes	CSCS.Y3.8.1 Identify potential mitigation strategies to prevent unnecessary information disclosure about the internal design or architecture of a network
NOTE: Common network troubleshooting tasks may include, but are not limited to, viewing internal IP address information (e.g., ipconfig /all); viewing external IP address information using an external service (e.g., ifconfig.me, myip.com, whatsmyip.com); validating communication with a remote system (e.g., ping); tracing path of communication to a remote system (e.g., traceroute); and releasing and renewing IP addresses (e.g., ipconfig /renew).		
CSCS.Y1.8.2 Research and describe common networking concepts at an introductory level	CSCS.Y2.8.2 Research and describe the following networking concepts and their relationship: ● Local IP and public IP and how they are assigned to individuals or organizations. ● Purpose of a MAC address ● Separation of network access (e.g., employee versus guest, staff versus student) ● Virtual private networks (VPN) and proxies	*Continuation of this standard is not specifically included or excluded*
NOTE: Networking concepts may include, but are not limited to, different types of networks (e.g., local area network (LAN), wide area network (WAN)); various common topologies; the role of a MAC address; local versus public IP and how they are assigned; Internet Protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6) addressing schemes; role of Domain Name System (DNS); the hierarchical nature of networks; purpose of virtual private networks (VPN); signal carriers for networks (e.g., copper, fiber optic, radio); purpose of firewalls; network access roles (e.g., employee versus guest, staff versus student); role of internet service providers (ISP); wireless connectivity; client-server relationship versus peer-to-peer (P2P); role of common internet protocols; and secure versus insecure protocols.		
CSCS.Y1.8.3 Research and describe modems, network interface cards, routers (e.g., consumer, industrial), switches, and wireless access points, and identify their purposes within a network	CSCS.Y2.8.3 Research and describe various types of network security and monitoring devices or concepts including, but not limited to, Access Control Lists (ACLs), firewalls, switch security, and WAN optimizers	CSCS.Y3.8.3 Research and describe network security and monitoring devices or concepts including, but not limited to, alerting versus logging, intrusion detection systems (IDS), intrusion prevention systems (IPS), and wireless intrusion detection systems (WIDS)

CSCS.Y1.8.4 Describe the importance of creating and using common rules for communication and the utilization of common network protocols including the relationship between client and server	CSCS.Y2.8.4 Research and describe the flow of common internet traffic by using a protocol analyzer (e.g., NetworkMiner, Wireshark, Zeek) to inspect how programs communicate over a network	CSCS.Y3.8.4 Analyze network traffic for suspicious or malicious activity using a protocol analyzer (e.g., NetworkMiner, Wireshark, Zeek)
NOTE: Discussions of common rules for communications may include, but are not limited to, the Open Systems Interconnection (OSI) Model and packet communication. Common network protocols may include, but are not limited to, DNS, Hypertext Transfer Protocol (HTTP)/Secure Hypertext Transfer Protocol (HTTPS), Simple Mail Transfer Protocol (SMTP)/Post Office Protocol (POP)/Internet Message Access Protocol (IMAP), and Telnet/Secure Shell (SSH). NOTE CSCS Y2-Y3: Analyzation may include, but is not limited to, identifying parties involved in communication, paths taken between those parties, and protocols used in communication.		

Strand: Computers and Communications

Content Cluster 9: Students will utilize appropriate hardware and software.

Year 1	Year 2	Year 3 - Advanced
CSCS.Y1.9.1 Compare and contrast computer programming paradigms (e.g., functional, imperative, object-oriented)	CSCS.Y2.9.1 Visually distinguish and identify level-appropriate source code from various programming languages and operating systems (e.g., assembly, Bash, C/C++, Java, JavaScript, PowerShell, Python)	CSCS.Y3.9.1 Create a functionally equivalent program of level-appropriate complexity in two or more programming languages
CSCS.Y1.9.2 Research, describe, and utilize at an appropriate level: ● debugging strategies ● integrated development environments (IDE) ● source-code editors ● version control strategies	CSCS.Y2.9.2 Use collaboration tools and version control systems in a group software project of appropriate complexity	*Continuation of this standard is not specifically included or excluded*
CSCS.Y1.9.3 Classify layers of software (e.g., applications, drivers, firmware, operating systems) utilized within various platforms (e.g., Android, ChromeOS, iOS, Linux, macOS, Windows)	CSCS.Y2.9.3 Research and describe techniques utilized by antivirus software to protect a system	CSCS.Y3.9.3 Research and describe tactics utilized by malware to resist removal from a system
CSCS.Y1.9.4 Identify and describe the purpose of hardware components within various personal computing platforms	CSCS.Y2.9.4 Research and describe strategies to limit the impacts of maliciously crafted hardware (e.g., BadUSB devices, hardware keyloggers, network implants)	*Continuation of this standard is not specifically included or excluded*
NOTE: Hardware components include, but are not limited to, central processing units (CPU), chassis, cooling components, graphics cards, input/output devices, memory, motherboards, power supplies, and storage devices.		

Strand: Professionalism and Impacts of Computing

Content Cluster 10: Students will analyze the impacts of technology and professionalism within the computing community.

<table>
<tr><th>Year 1</th><th>Year 2</th><th>Year 3 - Advanced</th></tr>
<tr><td>CSCS.Y1.10.1
Research and describe the risks and risk mitigation strategies associated with the utilization and implementation of social media and other digital technology implications</td><td>CSCS.Y2.10.1
Research and describe the various components of a threat model</td><td>CSCS.Y3.10.1
Identify and construct threat models</td></tr>
<tr><td colspan="3">NOTE:
Risks include, but are not limited to, cyberbullying, identity theft, impersonation, and social engineering attacks.
Implications may include, but are not limited to, employability, legal, physical, psychological, and social access.

NOTE CSCS Y2-Y3:
Threat model types include, but are not limited to, application, computing systems, organizational, and personal.</td></tr>
<tr><td><i>This standard is not specifically required until Year 2</i></td><td>CSCS.Y2.10.2
Research and describe issues related to creating and enforcing cyber-related laws and regulations (e.g., ethical challenges, policy vacuum, privacy versus security, unintended consequences)</td><td>CSCS.Y3.10.2
Research and describe the laws governing cybercrime and data security (e.g., Computer Fraud and Abuse Act of 1984 (CFAA), Digital Millennium Copyright Act (DMCA), General Data Protection Regulation (GDPR), Health Insurance Portability and Accountability Act of 1996 (HIPAA), Payment Card Industry Data Security Standard (PCI DSS), wire fraud laws)</td></tr>
<tr><td>CSCS.Y1.10.3
Research and describe the potential benefits associated with the utilization and implementation of social media and other digital technologies</td><td><i>Continuation of this standard is not specifically included or excluded</i></td><td><i>Continuation of this standard is not specifically included or excluded</i></td></tr>
<tr><td colspan="3">NOTE:
Potential benefits may include, but are not limited to, brand building, crowdsourcing, personal promotion awareness, and project funding.</td></tr>
<tr><td>CSCS.Y1.10.4
Research and describe the relationship between access and security (e.g., active and passive data, convenience, data mining, digital marketing, online wallets, privacy, theft of personal information)</td><td>CSCS.Y2.10.4
Identify the ethical implications encountered in the curation, management, and monetization of data (e.g., harvesting, information overload, knowledge management repositories, sharing, summarizing)</td><td>CSCS.Y3.10.4
Discuss ethical implications encountered in the cybersecurity industry that relate to intellectual property, non-compete clauses, and non-disclosure agreements</td></tr>
</table>

This standard is not specifically required until Year 2	CSCS.Y2.10.5 Explain advantages and disadvantages of various software life cycle processes (e.g., Agile, spiral, waterfall)	*Continuation of this standard is not specifically included or excluded*
CSCS.Y1.10.6 Research the history of computing devices and their impact on society	CSCS.Y2.10.6 Research the history of the hacking, phreaking, and general cybersecurity communities	*Continuation of this standard is not specifically included or excluded*
NOTE CSCS Y2: Historical topics may include, but are not limited to, historical periods and incidents within the hacking and phreaking community (e.g., crypto wars of the 90s, the golden era of hacking, post-Snowden era, post-Stuxnet era); exploring prominent figures, their exploits, and the ramifications stemming from those exploits; and the reasoning for the various labels or sub-groups that have emerged (e.g., bio-hackers, privacy hackers, social engineers, white hat).		
CSCS.Y1.10.7 Research and identify diverse careers and career opportunities (e.g., accessibility, availability, demand) that are influenced by computer science and the technical and soft skills needed for each	CSCS.Y2.10.7 Demonstrate industry-relevant technical and soft skills	CSCS.Y3.10.7 Create and maintain a professional digital portfolio comprised of self-created work

Strand: Professionalism and Impacts of Computing

Content Cluster 11: Students will demonstrate understanding of storytelling with data and appropriately communicate about technical information.

Year 1	Year 2	Year 3 - Advanced
CSCS.Y1.11.1 Communicate basic technical information effectively to diverse audiences including, but not limited to, non-technical audience members	CSCS.Y2.11.1 Communicate level-appropriate technical information effectively to diverse audiences including, but not limited to, non-technical audience members	CSCS.Y3.11.1 Communicate level-appropriate technical information effectively to diverse audiences including, but not limited to, non-technical audience members
NOTE: Technical information may include, but is not limited to, collecting or collected data, computing hardware, cyber hygiene, networking concepts, programming paradigms, and troubleshooting concepts.		
CSCS.Y1.11.2 Describe and utilize the concepts of storytelling with data	CSCS.Y2.11.2 Describe and utilize the concepts of storytelling within forensic investigations and incident response	CSCS.Y3.11.2 Describe and utilize the concepts of storytelling within forensic investigations and incident response
NOTE: Storytelling concepts may include, but are not limited to, identifying the knowledge level of the intended audience; developing a compelling narrative; creating appealing visualizations appropriate for the intended audience and that enhance the narrative; remaining objective and avoiding biases; and avoiding the censoring of data.		
CSCS.Y1.11.3 Describe the following common types of data bias: ● confirmation bias ● confounding variables ● outliers ● overfitting/underfitting ● selection bias	CSCS.Y2.11.3 Identify common types of bias in technical reports and how each can be used for exploitation	CSCS.Y3.11.3 Correct for or mitigate common types of bias in technical reports
CSCS.Y1.11.4 Compare and contrast causation and correlation	*Continuation of this standard is not specifically included or excluded*	CSCS.Y3.11.4 Correct for misinterpretations between causation and correlation
CSCS.Y1.11.5 Compare and contrast interpreting data, inferring using data, and implicating with data	CSCS.Y2.11.5 Interpret data, from the perspective of a business penetration test report or forensic timeline, to draw inferences and implications about system security	CSCS.Y3.11.5 Interpret data, from the perspective of a business risk assessment or cybersecurity assessment, to draw inferences and implications about system security

Contributors

The following people contributed to the development of this document:

Dr. Stephen Addison - Professor and CNSM Dean; University of Central Arkansas	**Mark McDougal** - K-12 Account Executive for Arkansas and Oklahoma; Apple Education
Scott Anderson - Executive Director; Forge Institute - Arkansas Cyber Alliance	**Mickey McFetridge** - Director of Federal Programs and Professional Learning; Fayetteville School District
Josh Baugh - Senior InfoSec Analyst; Entergy	**Dr. Josh McGee** - Chief Data Officer and Associate Director of Office for Education Policy; State of Arkansas and University of Arkansas
Garin Bean - Teacher; Cedarville Public Schools	**Ben Mcilmoyle** - Developer Advocate; Unity Technologies
Kimberly Bertschy - Program Coordinator, Networking and Cybersecurity; Northwest Arkansas Community College	**Deborah McMillan** - EAST Facilitator; Arkadelphia School District
John Black - Computer Specialist/Cyber Range Manager; University of Central Arkansas	**Eli McRae** - Statewide Computer Science Specialist; Arkansas Department of Education Office of Computer Science
Sarah Burnett - STEM Project Coordinator; Arkansas Tech University	**Alex Moeller** - Statewide Computer Science Specialist; Arkansas Department of Education Office of Computer Science
Julia Cottrell - K-8 STEM Coordinator; Van Buren School District	**Daniel Moix** - Director, STEM Pathways; Arkansas School for Mathematics, Sciences, and the Arts
Dr. Miles Dyson - Director of Special Projects; Cyberdyne Systems	**Adam Musto** - STEM Program Coordinator; Arkansas Division of Career and Technical Education
Jake Farmer - Teacher; Arkansas Arts Academy	**Allison Nicholas** - Director of Recruiting; Metova Inc.
Carl Frank - Teacher; Arkansas School for Mathematics, Sciences, and the Arts	**Anthony Owen** - State Director of Computer Science; Arkansas Department of Education Office of Computer Science
Jim Furniss - Statewide Computer Science Specialist; Arkansas Department of Education Office of Computer Science	**Dr. Elizabeth Parker** - Director of Financial and Statistical Analysis; Dillards
Tammy Glass - Statewide Computer Science Specialist; Arkansas Department of Education Office of Computer Science	**Kimberly Raup** - Teacher; Conway Public Schools
Tommy Gober - Curriculum Development Specialist; CYBER.ORG	**Ryan Raup** - Teacher; Conway Public Schools
Keith Godlewski - Teacher; Rogers Public Schools	**Stacy Reynolds** - Teacher; McGehee School District
Sean Gray - Teacher; Marion School District	**Mike Rogers** - Senior Director Maintenance and Refrigeration; Tyson Foods
Kelly Griffin - Statewide Computer Science Lead Specialist; Arkansas Department of Education Office of Computer Science	**Christy Ruffin** - Teacher; Lake Hamilton School District

John Hart - Statewide Computer Science Specialist; Arkansas Department of Education Office of Computer Science	**Jordan Sallis** - Cyber Intelligence Manager; GlaxoSmithKline
John Hightower - Department Head Computer Science and Engineering; University of Arkansas at Fort Smith	**Leslie Savell** - Statewide Computer Science Specialist; Arkansas Department of Education Office of Computer Science
Philip Huff - Assistant Professor of Cybersecurity and Director of Cybersecurity Research; University of Arkansas at Little Rock	**Dr. Karl Schubert** - Professor of Practice and Associate Director, Data Science Program; University of Arkansas
Grant Hurst - Teacher; North Little Rock School District	**Amanda Seidenzahl** - Director of Regional Workforce Grants; University of Arkansas at Fort Smith
Chris Jennings - Teacher; Valley View Public Schools	**Nicholas Seward** - Teacher; Arkansas School for Mathematics, Sciences, and the Arts
Lori Kagebein - Statewide Computer Science Specialist; Arkansas Department of Education Office of Computer Science	**Dr. Thilla Sivakumaran** - Vice Chancellor of Global Engagement and Outreach; Arkansas State University
Michael Karr - Makerspace Program Coordinator; National Park College	**Courtney Speer** - Technology Coach; Nettleton School District
David Kersey - Executive Director; PIXEL: A School for Media Arts	**Joel Spencer** - STEAM Magnet Coordinator; Little Rock School District
Catherine Leach - Associate Professor; Henderson State University	**Zackary Spink** - Statewide Computer Science Specialist; Arkansas Department of Education Office of Computer Science
Sandra Leiterman - Managing Director; UA Little Rock Cyber Gym	**Emily Torres** - Policy Development Coordinator; Arkansas Department of Education Office of Computer Science
Rhaelene Lowther - Associate Professor of Art: Game Art, Animation, and Simulation; Southern Arkansas University	**Morgan Warbington** - Program Advisor; Arkansas Department of Education Office of Computer Science
Gerri McCann - Teacher; Manila School District	**Bill Yoder** - Executive Director; Arkansas Center for Data Sciences
Amy McClure - Course Implementation Specialist; Virtual Arkansas	**Bradford Young** - Teacher; Mountain Home School District