Arkansas Computer Science and Computing Standards

High School Computer Engineering

2020

Arkansas Computer Science and Computing Standards for High School Computer Engineering

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 and Computing Standards for High School Computer Engineering begins during the 2021-2022 school year.

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

	Computer Engineering Year 1	**Computer Engineering Year 2**	**Computer Engineering Year 3 - Advanced**
Computer Engineering	465470	465480	465490

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 Computer Engineering

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
CSCE.Y1.1.1 Leverage problem-solving strategies to solve problems of level-appropriate complexity	CSCE.Y2.1.1 Leverage problem-solving strategies to solve problems of level-appropriate complexity	CSCE.Y3.1.1 Leverage problem-solving strategies to solve problems of level-appropriate complexity
NOTE: Problem-solving strategies that encompass computational thinking include, but are not limited to, abstraction, algorithm development, decomposition, and pattern recognition.		
CSCE.Y1.1.2 Analyze and utilize multiple representations of problem-solving logic used to solve problems of appropriate complexity	CSCE.Y2.1.2 Analyze and utilize multiple representations of problem-solving logic used to solve problems of appropriate complexity	CSCE.Y3.1.2 Analyze and utilize multiple representations of problem-solving logic used to solve problems of appropriate complexity
NOTE: Representations may include, but are not limited to, backlog, decision matrix, design brief, documentation, fault tree analysis, flowchart, pseudocode, and sprints.		
CSCE.Y1.1.3 Analyze and utilize collaborative methods in problem solving of level-appropriate complexity	CSCE.Y2.1.3 Analyze and utilize collaborative methods in problem solving of level-appropriate complexity	CSCE.Y3.1.3 Analyze and utilize collaborative methods in problem solving of level-appropriate complexity
NOTE: Collaborative methods may include, but are not limited to, distributive (divide and conquer), paired programming, and redundant parallel.		
CSCE.Y1.1.4 Analyze and utilize level-appropriate troubleshooting strategies for hardware and software	CSCE.Y2.1.4 Analyze and utilize level-appropriate troubleshooting strategies for hardware and software CSCE Y2: Analyze issues associated with power supply distribution and discuss mitigation strategies	CSCE.Y3.1.4 Analyze and utilize level-appropriate troubleshooting strategies for hardware and software

Strand: Computational Thinking and Problem Solving

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

Year 1	Year 2	Year 3 - Advanced
CSCE.Y1.2.1 Interpret relational and logical expressions of level-appropriate complexity using comparison and Boolean operators	CSCE.Y2.2.1 Interpret Boolean expressions using the two-level gate forms of AND-OR, OR-AND, NAND-NAND, NOR-NOR and positive/negative/mixed-logic conventions	CSCE.Y3.2.1 Describe and design the structure/operation of arithmetic building blocks including adders, comparators, shift registers, and subtractors; basic latches (D, SR); and flip-flops (D, JK, T)
NOTE: Boolean operators include AND, OR, NOT, and XOR. Comparison operators may include, but are not limited to, <, >, and !=.		
CSCE.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)	*Continuation of this standard is not specifically included or excluded*	*Continuation of this standard is not specifically included or excluded*
CSCE.Y1.2.3 Analyze how computer science concepts relate to the field of mathematics	*Continuation of this standard is not specifically included or excluded*	*Continuation of this standard is not specifically included or excluded*
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.		
CSCE.Y1.2.4 Discuss and apply concepts of abstraction	CSCE.Y2.2.4 Analyze and utilize concepts of abstraction as modeling and abstraction as encapsulation	*Continuation of this standard is not specifically included or excluded*
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).		
CSCE.Y1.2.5 Perform operations of level-appropriate complexity with binary, decimal, and hexadecimal numbers	CSCE.Y2.2.5 Perform operations of level-appropriate complexity with binary, octal, decimal, and hexadecimal numbers	*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.		

CSCE.Y1.2.6 Demonstrate operator precedence in expressions and statements	*Continuation of this standard is not specifically included or excluded*	*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	CSCE.Y2.2.7 Describe and represent basic electrical quantities including, but not limited to charge, current, energy, power, and voltage and describe the relationships among them	*Continuation of this standard is not specifically included or excluded*
This standard is not specifically required until Year 2	CSCE.Y2.2.8 Provide examples of using mathematical models in circuit simulators	CSCE.Y3.2.8 Produce mathematical models to represent material properties of electronic devices
This standard is not specifically required until Year 2	CSCE.Y2.2.9 Solve problems of level-appropriate complexity using fundamental laws of electricity (e.g., Faraday, Kirchhoff, Ohms)	CSCE.Y3.2.9 Solve problems of level-appropriate complexity using fundamental laws of electricity (e.g., Faraday, Kirchhoff, Ohms)
This standard is not specifically required until Year 2	CSCE.Y2.2.10 Perform conversions of level-appropriate complexity	CSCE.Y3.2.10 Perform conversions of level-appropriate complexity
NOTE CSCE Y2-Y3: Conversions may include, but are not limited to, converting to and from the formats specified by the Institute of Electrical and Electronics Engineers (IEEE) 754 standard for floating-point arithmetic; converting from a physical quantity such as acceleration, pressure, and temperature to voltage or current; and voltage to binary and binary to voltage numerical conversions given encoding method, range, and reference voltage parameters.		

Strand: Data, Information, and Security

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

Year 1	Year 2	Year 3 - Advanced
CSCE.Y1.3.1 Define, store, access, and manipulate level-appropriate data (e.g., primitive, linear)	CSCE.Y2.3.1 Create programs to store, access, and manipulate level-appropriate data (e.g., structured data, objects)	*Continuation of this standard is not specifically included or excluded*
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.		
CSCE.Y1.3.2 Define and discuss different examples of level-appropriate quantitative and qualitative data	CSCE.Y2.3.2 Define and discuss different examples of level-appropriate quantitative and qualitative data	*Continuation of this standard is not specifically included or excluded*
This standard is not specifically required until Year 2	CSCE.Y2.3.3 Research, discuss, and create level-appropriate programs to model and simulate probabilistic and real-world scenarios	*Continuation of this standard is not specifically included or excluded*
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.		
CSCE.Y1.3.4 Analyze, utilize, and visually represent level-appropriate data	CSCE.Y2.3.4 Analyze, utilize, and visually represent level-appropriate static and dynamic data	*Continuation of this standard is not specifically included or excluded*
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 is not limited to, network traffic, real-time weather data, sensor statuses, stock market valuations, and system status.		
CSCE.Y1.3.5 Perform level-appropriate data analysis using computing tools	CSCE.Y2.3.5 Perform level-appropriate data analysis using computing tools	*Continuation of this standard is not specifically included or excluded*

NOTE:
Analysis may include, but is not limited to, maximum values, mean values, minimum values, ranges, and string comparisons.

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>CSCE.Y1.4.1
Identify the five pillars of cybersecurity and evaluate the relevance of each pillar to computer science concepts</td><td>CSCE.Y2.4.1
Apply the five pillars of cybersecurity as applicable to level-appropriate computer science concepts</td><td>CSCE.Y3.4.1
Apply the five pillars of cybersecurity as applicable to level-appropriate computer science concepts</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>CSCE.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>Continuation of this standard is not specifically included or excluded</td><td>Continuation of this standard is not specifically included or excluded</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.</td></tr>
<tr><td>CSCE.Y1.4.3
Research and describe the impacts of ransomware, trojans, viruses, and other malware</td><td>CSCE.Y2.4.3
Research and describe common attacks on hardware, software, and networks</td><td>CSCE.Y3.4.3
Replicate level-appropriate and common attacks on hardware, software, and networks</td></tr>
<tr><td colspan="3">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.</td></tr>
</table>

CSCE.Y1.4.4 Explain implications related to identification and responsible reporting of a vulnerability versus exploitation	*Continuation of this standard is not specifically included or excluded*	*Continuation of this standard is not specifically included or excluded*
This standard is not specifically required until Year 2	CSCE.Y2.4.5 Define cracking and contrast it to hacking	CSCE.Y3.4.5 Enumerate techniques to combat cracking attacks

Strand: Algorithms and Programs

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

Year 1	Year 2	Year 3 - Advanced
CSCE.Y1.5.1 Design and implement level-appropriate algorithms that use iteration, selection, and sequence	CSCE.Y2.5.1 Design and implement level-appropriate algorithms that use iteration, recursion, selection, and sequence	CSCE.Y3.5.1 Design and implement level-appropriate algorithms including, but not limited to, brute force, divide and conquer, and greedy algorithms
CSCE.Y1.5.2 Illustrate the flow of execution of algorithms in level-appropriate programs including branching and looping	CSCE.Y2.5.2 Illustrate the flow of execution of algorithms in level-appropriate programs including combinational and sequential logic circuits (e.g., arithmetic blocks, combinatorial gates, memory elements) and finite state machines	CSCE.Y3.5.2 Illustrate the flow of execution of algorithms in level-appropriate programs including high-impedance state and logic gate implementation including a tri-state buffer
NOTE: Illustrations may include, but are not limited to, flowcharts and pseudocode.		
CSCE.Y1.5.3 Evaluate the qualities of level-appropriate student-created and non-student-created algorithms	CSCE.Y2.5.3 Evaluate the qualities of level-appropriate student-created and non-student-created algorithms including classic search and sort algorithms CSCE Y2: Include evaluation of scheduling algorithms on system performance; algorithms used in application domains including control applications; discrete event simulation applications; encryption/decryption algorithms; and location-aware or mobile applications	CSCE.Y3.5.3 Evaluate the qualities of level-appropriate student-created algorithms and non-student-created algorithms including parallel algorithms
NOTE: Evaluation tools may include, but are not limited to, a 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.		
CSCE.Y1.5.4 Use a systematic approach to detect and resolve errors in a given algorithm	CSCE.Y2.5.4 Use a systematic approach to detect and resolve errors in a given algorithm	CSCE.Y3.5.4 Use a systematic approach to detect and resolve errors in a given algorithm

This standard is not specifically required until Year 2	CSCE.Y2.5.5 Explain how regular expressions are related to finite state machines and why this is important	CSCE.Y3.5.5 Generate a regular expression to represent a specified language

Strand: Algorithms and Programs

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

Year 1	Year 2	Year 3 - Advanced
CSCE.Y1.6.1 Create programs using procedures to solve problems of level-appropriate complexity	CSCE.Y2.6.1 Create programs to solve problems of level-appropriate complexity	CSCE.Y3.6.1 Create programs to solve problems of level-appropriate complexity and to demonstrate an understanding of machine-level operations including: ● a deterministic finite state machine to accept a simple language ● level-appropriate assembly language programs
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.		
CSCE.Y1.6.2 Discuss and apply best practices of program design and format (e.g., descriptive names, documentation, indentation, user experience design, whitespace)	CSCE.Y2.6.2 Discuss and apply best practices of program design and format (e.g., descriptive names, documentation, indentation, user experience design, whitespace)	CSCE.Y3.6.2 Apply best practices of program design and format (e.g., descriptive names, documentation, indentation, user experience design, whitespace)
CSCE.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.		
CSCE.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	CSCE.Y2.6.4 Create programs that read from, write to, and append to a file of level-appropriate complexity that includes structured data	CSCE.Y3.6.4 Create programs that perform input/output (I/O) to an external device or system that uses an asynchronous serial interface

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.		
CSCE.Y1.6.5 Use a systematic approach to detect logic, runtime, and syntax errors within a program	CSCE.Y2.6.5 Use a systematic approach to detect logic, runtime, and syntax errors within a program	CSCE.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	CSCE.Y2.6.6 Describe direct memory access (DMA) and how it is supported on target embedded systems	CSCE.Y3.6.6 Create programs that perform a sequence of I/O operations using DMA
This standard is not specifically required until Year 2	CSCE.Y2.6.7 Create programs that measure waveform characteristics including frequency and pulse width using hardware timers	CSCE.Y3.6.7 Create programs that use pulse width modulation for external device control
This standard is not specifically required until Year 2	CSCE.Y2.6.8 Describe the sampling theorem and related concepts of the aliasing and Nyquist frequency	CSCE.Y3.6.8 Demonstrate aliasing and implement anti-aliasing strategies

Strand: Computers and Communications

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

Year 1	Year 2	Year 3 - Advanced
CSCE.Y1.7.1 Identify hardware and software specific to carrying out the mission of regional industries	CSCE.Y2.7.1 Utilize hardware and/or software to solve level-appropriate industry-based problems	CSCE.Y3.7.1 Model and utilize hardware and/or software to solve level-appropriate industry-based problems
CSCE.Y1.7.2 Research advancing and emerging technologies (e.g., artificially intelligent agents, blockchain, extended reality, Internet of Things (IoT), machine learning, robotics)	*Continuation of this standard is not specifically included or excluded*	*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
CSCE.Y1.8.1 Utilize the command line to accomplish common network troubleshooting tasks at an introductory level	*Continuation of this standard is not specifically included or excluded*	*Continuation of this standard is not specifically included or excluded*
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).		
CSCE.Y1.8.2 Research and describe common networking concepts at an introductory level	*Continuation of this standard is not specifically included or excluded*	*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.		
CSCE.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	*Continuation of this standard is not specifically included or excluded*	*Continuation of this standard is not specifically included or excluded*
CSCE.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	*Continuation of this standard is not specifically included or excluded*	*Continuation of this standard is not specifically included or excluded*
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).		

Strand: Computers and Communications

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

Year 1	Year 2	Year 3 - Advanced
CSCE.Y1.9.1 Compare and contrast computer programming paradigms (e.g., functional, imperative, object-oriented)	CSCE.Y2.9.1 Discuss early programming languages, their key features, why early software was written in machine language and assembly language, and the relationship between the encoding of machine-level operations at the binary level and their representation in a symbolic assembly language	CSCE.Y3.9.1 Explain the need for a hardware description language (HDL) in digital system design
CSCE.Y1.9.2 Research, describe, and utilize at an appropriate level: ● debugging strategies ● integrated development environments (IDE) ● source-code editors ● version control strategies	CSCE.Y2.9.2 Use collaboration tools and version control systems in a group software project of appropriate complexity	CSCE.Y3.9.2 Use collaboration tools and version control systems in a group software project of appropriate complexity
CSCE.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)	CSCE.Y2.9.3 Describe techniques used in real time operating systems (RTOS) including coroutines, message passing, mutexes, preemptive versus cooperative scheduling, queues, semaphores, and tasks	CSCE.Y3.9.3 Create programs using either a state machine framework or a RTOS for sample embedded system applications
CSCE.Y1.9.4 Identify and describe the purpose of hardware components within various personal computing platforms	CSCE.Y2.9.4 Contrast the circuit properties of different kinds of non-volatile storage elements (e.g., flash memory, read-only memory)	CSCE.Y3.9.4 Discuss common types of mixed-signal circuits and applications, including analog-to-digital (A/D) and digital-to-analog (D/A) converters and sample-and-hold circuits
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.		
This standard is not specifically required until Year 3	*This standard is not specifically required until Year 3*	CSCE.Y3.9.5 Compare, contrast, and utilize design tools and tool flow that are useful for the creation and simulation of digital circuits and systems

This standard is not specifically required until Year 2	CSCE.Y2.9.6 Apply properties of circuits containing various combinations of capacitance (C), inductance (L), resistance (R) and elements including damping, steady-state and transient responses, and time constants	*Continuation of this standard is not specifically included or excluded*
This standard is not specifically required until Year 2	CSCE.Y2.9.7 Identify the purpose of and represent basic circuit elements including, but not limited to, capacitors, inductors, resistors, and transformers	CSCE.Y3.9.7 Explain characteristics and properties of electronic materials including: ● acceptors, donors, and doping ● conductivity and resistivity ● drift and diffusion currents ● electrons and holes ● mobility and diffusivity ● n-type and p-type materials
This standard is not specifically required until Year 3	*This standard is not specifically required until Year 3*	CSCE.Y3.9.8 Explain the reasons and strategies for different computer architectures including, but not limited to, a von Neumann machine and indicate strengths and weaknesses inherent in each
This standard is not specifically required until Year 2	CSCE.Y2.9.9 Contrast various elements of circuit models including dependent and independent sources as well as parallel and series elements	*Continuation of this standard is not specifically included or excluded*
This standard is not specifically required until Year 2	CSCE.Y2.9.10 Define important engineering constraints such as cost, performance, power, size, timing, and weight and their tradeoffs in the context of digital systems design	*Continuation of this standard is not specifically included or excluded*
This standard is not specifically required until Year 2	*This standard is not specifically required until Year 3*	CSCE.Y3.9.11 Create programs that use one or more external sensors for monitoring physical properties
This standard is not specifically required until Year 2	CSCE.Y2.9.12 Discuss how accelerators (e.g., digital signal processors, field-programmable gate arrays, graphics processing units) can be used to improve performance	CSCE.Y3.9.12 Discuss how memory performance metrics, including bandwidth, cycle time, interleaving, and latency are used to measure the effects of memory on overall system performance

This standard is not specifically required until Year 2	CSCE.Y2.9.13 Contrast parallel I/O versus serial I/O tradeoffs in terms of application, cost, throughput, and wiring cost	CSCE.Y3.9.13 Describe the appropriateness of different I/O configurations (input, open-drain, strong drive, tri-state, weak pullup/pulldown) available in general purpose I/O for a given target application
This standard is not specifically required until Year 2	CSCE.Y2.9.14 Describe data formatting and timing diagrams	CSCE.Y3.9.14 Describe interrupts and signaling levels used in embedded systems
This standard is not specifically required until Year 3	*This standard is not specifically required until Year 3*	CSCE.Y3.9.15 Discuss mechanisms for buffering data streams including, but not limited to, stacks and queues (e.g., first in first out, last in first out)

Strand: Professionalism and Impacts of Computing

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

Year 1	Year 2	Year 3 - Advanced
CSCE.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	*Continuation of this standard is not specifically included or excluded*	*Continuation of this standard is not specifically included or excluded*
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.		
This standard is not specifically required until Year 2	CSCE.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)	*Continuation of this standard is not specifically included or excluded*
CSCE.Y1.10.3 Research and describe the potential benefits associated with the utilization and implementation of social media and other digital technologies	*Continuation of this standard is not specifically included or excluded*	*Continuation of this standard is not specifically included or excluded*
NOTE: Potential benefits may include, but are not limited to, brand building, crowdsourcing, personal promotion awareness, and project funding.		
CSCE.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)	CSCE.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)	*Continuation of this standard is not specifically included or excluded*
This standard is not specifically required until Year 2	CSCE.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*
CSCE.Y1.10.6 Research the history of computing devices and their impact on society	*Continuation of this standard is not specifically included or excluded*	*Continuation of this standard is not specifically included or excluded*

CSCE.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	CSCE.Y2.10.7 Demonstrate industry-relevant technical and soft skills	*Continuation of this standard is not specifically included or excluded*
This standard is not specifically required until Year 2	CSCE.Y2.10.8 Identify the components of a quality professional digital portfolio	CSCE.Y3.10.8 Evaluate the quality and impact of a professional digital portfolio
This standard is not specifically required until Year 2	CSCE.Y2.10.9 Create and maintain a digital collection of self-created work	CSCE.Y3.10.9 Create and maintain a professional digital portfolio comprised of self-created work
NOTE CSCE Y2-Y3: Self-created works may include, but are not limited to, diagrams, media, project reports, and source code.		

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
CSCE.Y1.11.1 Communicate basic technical information effectively to diverse audiences, including but not limited to, non-technical audience members	*Continuation of this standard is not specifically included or excluded*	*Continuation of this standard is not specifically included or excluded*
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.		
CSCE.Y1.11.2 Describe and utilize the concepts of storytelling with data	*Continuation of this standard is not specifically included or excluded*	*Continuation of this standard is not specifically included or excluded*
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.		
CSCE.Y1.11.3 Describe the following common types of data bias: ● confirmation bias ● confounding variables ● outliers ● overfitting/underfitting ● selection bias	*Continuation of this standard is not specifically included or excluded*	*Continuation of this standard is not specifically included or excluded*
CSCE.Y1.11.4 Compare and contrast causation and correlation	*Continuation of this standard is not specifically included or excluded*	*Continuation of this standard is not specifically included or excluded*
CSCE.Y1.11.5 Compare and contrast interpreting data, inferring using data, and implicating with data	*Continuation of this standard is not specifically included or excluded*	*Continuation of this standard is not specifically included or excluded*

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