Arkansas Computer Science Standards for Grades 9-12

Computer Science Courses Levels 1 - 4

2016

(Updated 2018)

Arkansas Computer Science Standards for Grades 9-12

Introduction

The Arkansas Computer Science Standards for High School are designed to provide foundational 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 and Information; Algorithms and Programs; Computers and Communications; and Community, Global, and Ethical Impacts. 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, whether in computer science or in other fields.

Each semester course (level) may be taught with one of three emphases: programming/coding, networking/hardware, or information security. The choice of which emphasis to utilize is a local decision. When teaching the standards with an emphasis in a particular focus, the quantity and content of the standards do not change; the content delivery methods and student project areas shall reflect the chosen emphasis.

The Arkansas State Board of Education (SBE) does not place any pre-requisites on the Arkansas Computer Science 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. Though there are no specific course prerequisites, students enrolling in Advanced Programming, Advanced Networking, or Advanced Information Security should understand and be able to apply the content/concepts found within the Arkansas Computer Science Courses Levels 1 - 4.

The SBE and ADE authorize schools to enroll students across levels and emphases 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 Grades 9-12 begins during the 2017-2018 school year.

Course Titles: Computer Science Levels 1 - 4
Course/Unit Credit: 0.5 Credits per course/level

	Computer Science Level 1	**Computer Science Level 2**	**Computer Science Level 3**	**Computer Science Level 4**
Computer Science with Programming/Coding Emphasis	465010	465020	465030	465040
Mobile Application Development	465310	465320	465330	465340
Networking/Hardware Emphasis	465110	465120	465130	465140
Robotics	465510	465520	465530	465540
Information Security Emphasis	465210	465220	465230	465240

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 Computer Science levels; it is up to the local district to determine placement based on student ability.

Computer Science Practices

Students will exhibit proficiency in computer science through:

Perseverance - Students expect and persist in overcoming the challenges that occur when completing tasks. They recognize that making and correcting mistakes will take place during the learning process and problem solving.

Collaboration - Students effectively work and communicate with others 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.

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.

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 application programs, programming languages, or appropriate computing devices.

Communication - Students effectively communicate, using accurate and appropriate terminology, when explaining the task completion or problem solving strategies that were used. They recognize that good documentation is an ongoing part of the process, and when appropriate, provide accurate documentation of their work in a manner that is understandable to others.

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.

Problem Solving - Students exhibit proficiency in Computer Science through identifying and systematically solving problems (e.g., engineering design process). They recognize problem solving as an ongoing process.

Arkansas Computer Science Standards for High School

Strand	Content Cluster
Computational Thinking and Problem Solving	
	1. Students will analyze problem-solving strategies.
	2. Students will analyze connections between elements of mathematics and computer science.
Data and Information	
	3. Students will store and manipulate data through the use of computing devices.
	4. Students will analyze and interpret data through the use of computing devices.
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.
	8. Students will analyze resilient, reliable, and adaptable communication methods and systems used to transmit information among computing devices.
	9. Students will utilize appropriate hardware and software.
Community, Global, and Ethical Impacts	
	10. Students will analyze appropriate uses of technology and its social and global impacts.

Notes for the Computer Science Standards for High School document:

1. The examples given (e.g.,) are suggestions to guide the instructor.
2. The Practices are intended to be habits of mind for all students and were written broadly in order to apply to all grades. The Practices are not content standards and are not intended to be formally assessed but may be assessed formatively.
3. 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.
4. Notes found within the document are not approved 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 are subject to change as understandings of the standards evolve.

Strand: Computational Thinking and Problem Solving

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

THE GOAL FOR EACH STUDENT IS PROFICIENCY IN ALL REQUIREMENTS AT CURRENT AND PREVIOUS LEVELS			
Level 1	Level 2	Level 3	Level 4
CSL1.1.1 Leverage problem-solving strategies to solve problems of level-appropriate complexity	CSL2.1.1 Leverage problem-solving strategies to solve problems of level-appropriate complexity	CSL3.1.1 Leverage problem-solving strategies to solve problems of level-appropriate complexity	CSL4.1.1 Leverage problem-solving strategies to solve problems of level-appropriate complexity
NOTE: Some problem-solving strategies may include but are not limited to recursion, iteration, Agile method, 6-step engineering design process, and waterfall.			
CSL1.1.2 Compare and contrast multiple representations of problem-solving logic	CSL2.1.2 Analyze multiple representations of problem-solving logic	CSL3.1.2 Design multiple representations of problem-solving logic used to solve a problem of appropriate complexity	CSL4.1.2 Critique multiple representations of problem-solving logic used to solve a problem of appropriate complexity
NOTE: Some representation methods may include but are not limited to documentation, backlog, sprints, decision matrix, design brief, flowchart, and pseudocode.			
CSL1.1.3 Analyze and implement collaborative methods in problem solving of level-appropriate complexity	CSL2.1.3 Analyze and implement collaborative methods in problem solving of level-appropriate complexity	CSL3.1.3 Analyze and implement collaborative methods in problem solving of level-appropriate complexity	CSL4.1.3 Analyze and implement collaborative methods in problem solving of level-appropriate complexity
NOTE: Some implementation methods may include but are not limited to paired programming, distributive (divide & conquer), and redundant parallel.			
CSL1.1.4 Recognize processes and techniques for troubleshooting of level-appropriate complexity	CSL2.1.4 Recognize processes and techniques for troubleshooting of level-appropriate complexity	CSL3.1.4 Recognize processes and techniques for troubleshooting of level-appropriate complexity	CSL4.1.4 Recognize processes and techniques for troubleshooting of level-appropriate complexity
NOTE: Some processes and techniques for troubleshooting may include but are not limited to tracing; debugging; identification/removal of malware; and error-classification including syntax, logic, runtime, and off-by-one errors.			
CSL1.1.5 Decompose a problem of level-appropriate complexity into more simple, solvable parts	CSL2.1.5 Decompose a problem of level-appropriate complexity into more simple, solvable parts	CSL3.1.5 Decompose a problem of level-appropriate complexity into more simple, solvable parts	CSL4.1.5 Decompose a problem of level-appropriate complexity into more simple, solvable parts
NOTE for CSL1.1.5 through CSL4.1.5: Solvable parts may include but are not limited to methods, functions, and subroutines with and without parameters.			

Strand: Computational Thinking and Problem Solving

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

THE GOAL FOR EACH STUDENT IS PROFICIENCY IN ALL REQUIREMENTS AT CURRENT AND PREVIOUS LEVELS.			
Level 1	Level 2	Level 3	Level 4
CSL1.2.1 Interpret logical expressions using Boolean operators (e.g., AND, NOT, OR, XOR)	CSL2.2.1 Interpret logical expressions using short-circuit evaluation	CSL3.2.1 *Continuation of this standard is not specifically included or excluded*	CSL4.2.1 *Continuation of this standard is not specifically included or excluded*
CSL1.2.2 Classify the types of information that can be stored as variables (e.g., Booleans, characters, integers, floating points, strings)	CSL2.2.2 *Continuation of this standard is not specifically included or excluded*	CSL3.2.2 *Continuation of this standard is not specifically included or excluded*	CSL4.2.2 *Continuation of this standard is not specifically included or excluded*
CSL1.2.3 Identify mathematical concepts (e.g., random number generation, vocabulary) related to computer science	CSL2.2.3 Recognize the similarities and differences between mathematics and computer science algorithms	CSL3.2.3 Demonstrate basic encryption (e.g., block cipher, Caesar cipher)	CSL4.2.3 *Continuation of this standard is not specifically included or excluded*
CSL1.2.4 *This standard is not specifically required until Level 2*	CSL2.2.4 Discuss the concept of abstraction	CSL3.2.4 Analyze the concepts of abstraction as modeling and abstraction as encapsulation	CSL4.2.4 Use the concepts of abstraction as modeling and abstraction as encapsulation
CSL1.2.5 *This standard is not specifically required until Level 2*	CSL2.2.5 Perform simple operations with $base_{10}$, $base_2$, and $base_{16}$ numbers	CSL3.2.5 *Continuation of this standard is not specifically included or excluded*	CSL4.2.5 Perform simple operations with $base_{10}$, $base_2$, $base_{8,}$ and $base_{16}$ numbers
NOTE for CSL2.2.5 and CSL4.2.5: Some operations may include but are not limited to addition, subtraction, and conversion.			
CSL1.2.6 Demonstrate operator (e.g., +, -, /, %, concatenation) precedence in expressions and statements	CSL2.2.6 Demonstrate operator (e.g., math, pow, sqrt) precedence in expressions and statements	CSL3.2.6 *Continuation of this standard is not specifically included or excluded*	CSL4.2.6 *Continuation of this standard is not specifically included or excluded*
NOTE for CSL1.2.6 through CSL4.2.6: Some examples of operator precedence and assignment may include but are not limited to inside-out, order of operations, and x = 1 is not the same as 1 = x.			

Strand: Data and Information

Content Cluster 3: Students will store and manipulate data through the use of computing devices.

<table>
<tr><td colspan="4">THE GOAL FOR EACH STUDENT IS PROFICIENCY IN ALL REQUIREMENTS AT CURRENT AND PREVIOUS LEVELS.</td></tr>
<tr><td>Level 1</td><td>Level 2</td><td>Level 3</td><td>Level 4</td></tr>
<tr><td>CSL1.3.1
Define, store, and manipulate primitive data</td><td>CSL2.3.1
Define, store, and manipulate linear data</td><td>CSL3.3.1
Define, store, and manipulate structured data and objects</td><td>CSL4.3.1
Create a program to store and manipulate various data</td></tr>
<tr><td colspan="4">NOTE for CSL1.3.1 through CSL4.3.1:
Primitive data can include, but is not limited to, bool, char, double, float, int.
Linear data can include, but is not limited to, arrays, lists, strings, vectors.
Structured data can include, but is not limited to, arrays, classes, linked lists, multidimensional arrays, structs, user-defined classes.
Objects can include, but are not limited to, constructors, data members, methods, pass-by-value/pass-by-reference parameters.
Defining and storing can include, but are not limited to, modifiers such as final, private, protected, public.
Manipulating data can include, but is not limited to, arranging (including stacking and queuing), casting, rearranging, sorting.</td></tr>
<tr><td>CSL1.3.2
Compare and contrast level-appropriate numeric and non-numeric data representations</td><td>CSL2.3.2
Compare and contrast level-appropriate numeric and non-numeric data representations</td><td>CSL3.3.2
Compare and contrast level-appropriate numeric and non-numeric data representations</td><td>CSL4.3.2
Compare and contrast level-appropriate numeric and non-numeric data representations</td></tr>
<tr><td colspan="4">NOTE for CSL1.3.2 through CSL4.3.2: Topics could include, but are not limited to, analog vs. digital, ASCII/Unicode, bar codes, compression, encoding, light/pixels, size of file vs. data types vs. storage needed, sound wave/sampling.</td></tr>
</table>

Strand: Data and Information

Content Cluster 4: Students will analyze and interpret data through the use of computing devices.

THE GOAL FOR EACH STUDENT IS PROFICIENCY IN ALL REQUIREMENTS AT CURRENT AND PREVIOUS LEVELS.			
Level 1	Level 2	Level 3	Level 4
CSL1.4.1 *This standard is not specifically required until Level 2*	CSL2.4.1 Analyze the degree to which a computer model accurately represents an actual situation (e.g., Conway's Game of Life, population growth, predator-prey)	CSL3.4.1 Critique techniques for creating models, simulations, and generating random numbers to be used for data analysis	CSL4.4.1 Create various models and simulations as predictors for probabilistic scenarios (e.g., flip a coin, random walker, roll a die) and/or real-world scenarios (e.g., city population, predator-prey)
CSL1.4.2 Examine the ability of computing technology to create and process Big Data	CSL2.4.2 Determine an appropriate visual representation for given data	CSL3.4.2 Compare and contrast multiple visual representation tools for given data	CSL4.4.2 *Continuation of this standard is not specifically included or excluded*
NOTE for CSL1.4.2 through CSL4.4.2: Visual representation tools may include, but are not limited to, spreadsheets, Google Analytics, Python libraries, and other programming language libraries.			
CSL1.4.3 *This standard is not specifically required until Level 2*	CSL2.4.3 Implement algorithms to perform data analysis (e.g., longest string, maximum, mean, minimum, range)	CSL3.4.3 *Continuation of this standard is not specifically included or excluded*	CSL4.4.3 *Continuation of this standard is not specifically included or excluded*

Strand: Algorithms and Programs

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

THE GOAL FOR EACH STUDENT IS PROFICIENCY IN ALL REQUIREMENTS AT CURRENT AND PREVIOUS LEVELS.			
Level 1	Level 2	Level 3	Level 4
CSL1.5.1 Construct and evaluate simple expressions using relational and logical operators	CSL2.5.1 Construct and evaluate compound expressions using relational and logical operators	CSL3.5.1 *Continuation of this standard is not specifically included or excluded*	CSL4.5.1 *Continuation of this standard is not specifically included or excluded*
CSL1.5.2 Design and implement algorithms that use sequence and selection including nested ifs (e.g., if, if/else, if/else if, switch-case)	CSL2.5.2 Design and implement algorithms that use sequence, selection, and iteration including nested loops (e.g., for, for each, while, do while)	CSL3.5.2 Design and implement algorithms that use sequence, selection, iteration and recursion	CSL4.5.2 *Continuation of this standard is not specifically included or excluded*
CSL1.5.3 Illustrate the flow of execution of a program including branching and looping	CSL2.5.3 Illustrate the flow of execution of an increasingly complex program including branching and looping	CSL3.5.3 Critically analyze classic search and sort algorithms in different contexts, adapting as appropriate	CSL4.5.3 *Continuation of this standard is not specifically included or excluded*
CSL1.5.4 Evaluate the qualities of level-appropriate algorithms	CSL2.5.4 Evaluate the qualities of level-appropriate algorithms	CSL3.5.4 Evaluate the qualities of level-appropriate algorithms	CSL4.5.4 Evaluate the qualities of level-appropriate algorithms
NOTE for CSL1.5.4 through CSL4.5.4: Evaluation tools can include, but are not limited to, a code review and test cases. Qualities can include correctness, usability, readability, efficiency, portability, and scalability.			
CSL1.5.5 Utilize a systematic approach to detect structural and logic errors	CSL2.5.5 Utilize a systematic approach to detect structural and logic errors	CSL3.5.5 Utilize a systematic approach to detect structural and logic errors	CSL4.5.5 Utilize a systematic approach to detect structural and logic errors

Strand: Algorithms and Programs

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

THE GOAL FOR EACH STUDENT IS PROFICIENCY IN ALL REQUIREMENTS AT CURRENT AND PREVIOUS LEVELS.			
Level 1	Level 2	Level 3	Level 4
CSL1.6.1 Create programs to solve problems of level-appropriate complexity applying best practices of program design and format (e.g., descriptive names, documentation, indentation, whitespace)	CSL2.6.1 Create programs to solve problems of level-appropriate complexity applying best practices of program design and format (e.g., descriptive names, documentation, indentation, whitespace)	CSL3.6.1 Create programs to solve problems of level-appropriate complexity applying best practices of program design and format (e.g., descriptive names, documentation, indentation, whitespace)	CSL4.6.1 Create programs to solve problems of level-appropriate complexity applying best practices of program design and format (e.g., descriptive names, documentation, indentation, whitespace)
NOTE for CSL1.6.1 through CSL4.6.1: Problems of varying complexity can include, but are not limited to, encoding, encryption, finding minimum/maximum values, identifying prime numbers, searching and sorting, and solving the Towers of Hanoi.			
CSL1.6.2 Utilize functions/methods/procedures to input, output, and manipulate data with and without parameters	CSL2.6.2 Determine the scope of variables declared in functions/methods/procedures and control structures	CSL3.6.2 Determine the scope of variables and functions/methods/procedures declared in objects (e.g., public, private, encapsulation)	CSL4.6.2 Determine the scope of variables and functions/methods/procedures defined in abstract classes and interfaces (e.g., encapsulation, inheritance, polymorphism)
NOTE for CSL1.6.2 through CSL4.6.2: In conjunction with standards CSL1.3.1 through CSL4.3.1, the goal is to introduce and implement object-oriented programming.			
CSL1.6.3 Create a program that reads from standard input and writes to standard output	CSL2.6.3 Create a program that reads from a file and writes to a file	CSL3.6.3 *Continuation of this standard is not specifically included or excluded*	CSL4.6.3 *Continuation of this standard is not specifically included or excluded*
CSL1.6.4 *This standard is not specifically required until Level 4*	CSL2.6.4 *This standard is not specifically required until Level 4*	CSL3.6.4 *This standard is not specifically required until Level 4*	CSL4.6.4 Explain advantages and disadvantages of various software life cycle processes (e.g., Agile, spiral, waterfall) by participating on software project teams

Strand: Computers and Communications

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

THE GOAL FOR EACH STUDENT IS PROFICIENCY IN ALL REQUIREMENTS AT CURRENT AND PREVIOUS LEVELS.			
Level 1	Level 2	Level 3	Level 4
CSL1.7.1 *This standard is not specifically required until Level 2*	CSL2.7.1 Characterize how software and/or hardware is used in industry (e.g., business, government, medical, military, sports)	CSL3.7.1 *Continuation of this standard is not specifically included or excluded*	CSL4.7.1 Utilize software and/or hardware to solve various industry-based problems
CSL1.7.2 Identify desired technical and soft skills (e.g., collaboration, communication, problem solving, teamwork) that can be enhanced by computer science	CSL2.7.2 Discuss technical and soft skills honed by computer science	CSL3.7.2 Demonstrate technical and soft skills honed by computer science	CSL4.7.2 Demonstrate technical and soft skills honed by computer science
CSL1.7.3 Discuss diverse careers that are influenced by computer science and its availability to all regardless of background	CSL2.7.3 Analyze a historical timeline of computers and technology	CSL3.7.3 Explore advancing and emerging technologies (e.g., Artificially Intelligent Agents, Robotics, Internet of Things [IoT])	CSL4.7.3 Explain how cutting-edge technology may affect the way business is conducted in the future (e.g., eCommerce, entrepreneurship, payment methods, business responsibilities)

Strand: Computers and Communications

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

THE GOAL FOR EACH STUDENT IS PROFICIENCY IN ALL REQUIREMENTS AT CURRENT AND PREVIOUS LEVELS.			
Level 1	Level 2	Level 3	Level 4
CSL1.8.1 Utilize networks to perform level-appropriate tasks	CSL2.8.1 Utilize networks to perform level-appropriate tasks	CSL3.8.1 Utilize networks to perform level-appropriate tasks	CSL4.8.1 Utilize networks to perform level-appropriate tasks
CSL1.8.2 Discuss the role of internet service providers (ISP) in providing connectivity	CSL2.8.2 Discuss the hierarchical nature of networks, subnetworks, and the Internet	CSL3.8.2 Analyze how the nature of networks allow for a continual increase in the number of devices	CSL4.8.2 Research projects that utilize the power created through the networking of computers to solve level-appropriate problems
CSL1.8.3 Compare and contrast local area networks (LAN) and wide area networks (WAN)	CSL2.8.3 Identify various common topologies utilized in network implementations	CSL3.8.3 Analyze the tradeoffs of implementing various common topologies	CSL4.8.3 Analyze the tradeoffs of implementing increasingly complex topologies
CSL1.8.4 *This standard is not specifically required until Level 2*	CSL2.8.4 Identify digital and physical methods used to secure networks	CSL3.8.4 Discuss digital and physical methods used to secure networks	CSL4.8.4 Design a practical, efficient, and secure network solution (e.g., small office network)
CSL1.8.5 Identify common network protocols (e.g., DNS, HTTP/HTTPS, SMTP/POP/IMAP, Telnet/SSH)	CSL2.8.5 Compare and contrast common network protocols (e.g., DNS, HTTP/HTTPS, SMTP/POP/IMAP, Telnet/SSH)	CSL3.8.5 Analyze the Open Systems Interconnect (OSI) Model layers 1-7	CSL4.8.5 Map network operations to the OSI Model

Strand: Computers and Communications

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

THE GOAL FOR EACH STUDENT IS PROFICIENCY IN ALL REQUIREMENTS AT CURRENT AND PREVIOUS LEVELS.			
Level 1	Level 2	Level 3	Level 4
CSL1.9.1 Compare and contrast computer programming paradigms and languages (e.g., text-based, visual, high-level, low-level, object-oriented)	CSL2.9.1 Compare and contrast the tradeoffs between compiled and interpreted languages	CSL3.9.1 Discuss considerations when programming for multiple computing platforms (e.g., desktop, mobile, web)	CSL4.9.1 *Continuation of this standard is not specifically included or excluded*
CSL1.9.2 Discuss version control and Integrated Development Environments (IDE)	CSL2.9.2 Use the debugger in an IDE	CSL3.9.2 Use collaboration tools in a group software project (e.g., cloud-based software)	CSL4.9.2 Use version control systems
CSL1.9.3 Classify layers of software (e.g., applications, drivers, operating systems) within various platforms	CSL2.9.3 *Continuation of this standard is not specifically included or excluded*	CSL3.9.3 *Continuation of this standard is not specifically included or excluded*	CSL4.9.3 *Continuation of this standard is not specifically included or excluded*
CSL1.9.4 Identify hardware components (e.g., input/output devices, internal organization of a computer, storage devices) of computing technology within various platforms	CSL2.9.4 *Continuation of this standard is not specifically included or excluded*	CSL3.9.4 *Continuation of this standard is not specifically included or excluded*	CSL4.9.4 *Continuation of this standard is not specifically included or excluded*

Strand: Community, Global, and Ethical Impacts

Content Cluster 10: Students will analyze appropriate uses of technology and its social and global impacts

THE GOAL FOR EACH STUDENT IS PROFICIENCY IN ALL REQUIREMENTS AT CURRENT AND PREVIOUS LEVELS.			
Level 1	Level 2	Level 3	Level 4
CSL1.10.1 Categorize the risks associated with the utilization and implementation of digital technology • Legal • Physical • Psychological • Social NOTE: Legal issues include but are not limited to access, AFTRA, copyright, FAA, FCC, hacking, intellectual property, licensure, local computer-use policy, piracy, and plagiarism.	CSL2.10.1 Discuss the effects associated with the use of social media (e.g., global communication, hiring, incarceration, termination)	CSL3.10.1 Explain conflicting issues related to creating and enforcing cyber-related laws and regulations (e.g., ethical challenges, policy vacuum, privacy vs. security, unintended consequences)	CSL4.10.1 Formulate solutions that address the risks associated with extensive use and implementation of digital technology
CSL1.10.2 Discuss issues related to personal security	CSL2.10.2 Identify components of a digital footprint (e.g., active and passive data) and the lasting impact	CSL3.10.2 Explore the inverse relationship between online privacy and personal security (e.g., convenience and accessibility, data mining, digital marketing, online wallets, theft of personal information)	CSL4.10.2 *Continuation of this standard is not specifically included or excluded*
CSL1.10.3 *This standard is not specifically required until Level 2*	CSL2.10.3 *Continuation of this standard is not specifically included or excluded*	CSL3.10.3 Describe the beneficial and intrusive aspects of advancing and emerging technologies (e.g., Artificially Intelligent Agents, IoT, Robotics, self-aware, Skynet)	CSL4.10.3 Identify the ethical and moral implications encountered in managing and curating knowledge (e.g., harvesting, information overload, knowledge management reposting, sharing, summarizing)

Appendix 1: Computer Science: Mobile Applications Development

This appendix contains exceptions that apply to the teaching of Mobile Applications Development under the High School Computer Science standards. Students enrolled in Computer Science Mobile Applications Development at any level, must receive instruction in all High School Computer Science Standards within the CS level to which the MAD course appends. The following exceptions apply to the standard indicated and modify the requirements of that standard only, all other standards within that level must be taught as presented above, and any additional standards specific to MAD will be listed at the end of the exceptions.

High School Computer Science Level 1: Mobile Applications Development Level 1	
CSL1.5.3	To meet this standard, a visual programming environment should be used.
CSL1.6.1	To meet this standard, multiple applications for Android or iOS should be developed.
CSL1.7.2	To meet this standard, skills for employment and various roles (e.g., developer, graphic designer, project manager, team leader, quaility assurance) required by app development companies should be identified.
CSL1.9.2	To meet this standard an introductory discussion must include: • A basic visual programming environment (e.g, Scratch, Alice), • Android based visual and drag-and-drop programming environment (e.g., App Inventor), and • IOS based visual and drag-and-drop programming environment (e.g., App Lab, Game Salad).
CSL1.9.3	To meet this standard, top apps/genres must be compared.
CSL1.9.4	To meet this standard, Android or IOS devices and their components (e.g., sensors, input/output, interface elements) must be included.
CSL1.11.1 *(addition)*	Explore the Apple or Android developer website and determine steps to become a developer.
High School Computer Science Level 2: Mobile Applications Development Level 2	
CSL2.5.2	To meet this standard, both drag and drop and text-based programming paradigms should be used.
CSL2.6.1	To meet this standard, applications should be development for a different platform than was used in Level 1 (e.g., iOS vs. Android) or at least two platforms if MAD Level 1 was not taken.
CSL2.9.2	To meet this standard, a text-based IDE must be used (e.g., Eclipse, xCode).
High School Computer Science Level 3: Mobile Applications Development Level 3	
For all applicable CSL3 standards	In order to meet the standards of Level 3, a text-based mobile application environment should be used for either an Android or iOS platform.

High School Computer Science Level 4: Mobile Applications Development Level 4	
CSL4.1.5	To meet this standard, the "is-a" and "has-a" object oriented concepts must be explored.
CSL4.3.1	To meet this standard, implement: • A model class for tracking user input, and • A controller and viewer for application.
CSL4.6.1	To meet this standard: • Identify touch events (e.g., begin, canceled, end, move), • Create touch-event applications, and • Create code to clear screen in application.
CSL4.11.1 *(addition)*	Find and use the appropriate APIs and documentation to create various basic mobile applications for iOS or Android devices.

Appendix 2: Computer Science: Robotics

This appendix contains exceptions that apply to the teaching of Robotics under the High School Computer Science standards. Students enrolled in Computer Science Robotics at any level, must receive instruction in all High School Computer Science Standards within the CS level to which the Robotics course appends. The following exceptions apply to the standard indicated and modify the requirements of that standard only, all other standards within that level must be taught as presented above, and any additional standards specific to Robotics will be listed at the end of the exceptions.

High School Computer Science Level 1: Robotics Level 1	
CSL1.1.1	To meet this standard, focus must be on creating the plans, drawings and algorithms that describe the product, process or system that will be implemented.
CSL1.1.4	(e.g., com port, computer interface, driver installations, hardware, micro-controller interface, system disconnect, wiring)
CSL1.7.3	To meet this standard, focus must be on careers in robotics
CSL1.9.2	To meet this standard, choose an IDE that correlates to the Robot's system language (e.g., RobotC, C++, JAVA, C#, Python)
CSL1.9.4	To meet this standard, describe and discuss microcontrollers and their varied uses (e.g., Lego brick, VEX ARM, Arduino)
High School Computer Science Level 2: Robotics Level 2	
CSL2.1.4	(e.g., com port, computer interface, driver installations, hardware, micro-controller interface, system disconnect, wiring).
CSL2.2.6	Demonstrate operator (e.g., math, pow, sqrt) precedence in expressions and statements as correlated to movement of the robot.
CSL2.3.1	Define, store, and manipulate linear data through sensor data.
CSL2.4.1	(e.g., crowd dynamic studies, look for patterns through sensory feedback).
CSL2.6.2	NOTE for CSL2.6.2: Additional sensors may be necessary to increase functionality (e.g., light, sound, temperature).
CSL2.6.3	NOTE for CSL2.6.3 Additional sensors may be necessary to keep a log file.
CSL2.7.2	Discuss technical and soft skills honed by computer science as related to robotics
CSL2.8.1	(e.g., Bluetooth, additional sensors/components may be required, Wi-Fi)
CSL2.8.4	To meet this standard, discussions must include circuit pathways and logic.
CSL2.10.1	Discuss the effects associated with the use of social media and robotic technology (e.g., drones, global communication, hiring, incarceration, privacy issues termination)

High School Computer Science Level 3: Robotics Level 3	
CSL3.3.1	To meet this standard, progressive use of sensors and open-loop and closed-loop controls must be included.
CSL3.6.2	To meet this standard, additional sensors shall be used to increase functionality.
CSL3.7.2	Discuss technical and soft skills honed by computer science and any specific to the field of robotics
CSL3.10.1	Explain conflicting issues related to creating and enforcing laws and regulations regarding the use of computer science and any specific to the field of robotic technology (e.g., self-driving cars, technological unemployment, and environmental considerations)
CSL3.10.3	(e.g., Artificially Intelligent Agents, IoT, Robotics, self-aware, Skynet, smart homes, and constant digital monitoring)
High School Computer Science Level 4: Robotics Level 4	
CSL4.3.1	To meet this standard, progressive use of sensors and open-loop and closed-loop controls must be included.
CSL4.4.1	To meet this standard, real world scenarios shall include autonomous versus user control.
CSL4.6.2	To meet this standard, additional sensors shall be used to increase functionality.
CSL4.7.2	Discuss technical and soft skills honed by computer science and any specific to the field of robotics

Contributors

The following people contributed to the development of this document:

Stephany Alhajjaj – Little Rock School District	Lori Kagebein – Wonderview School District
Jeff Anderson – Rogers Public Schools	Jeff Matocha – Ouachita Baptist University
Brent Burgin – Dassault Falcon Jet	Daniel Moix – Arkansas School for Mathematics, Sciences, and the Arts
Kristian Cartwright – Fayetteville Public Schools	Larry Morell – Arkansas Tech University
Kevin Collins – Alma School District	David Nance – Arkansas Department of Education
Cecil Cossey – Hamburg School District	Thad Nipp – Alma School District
Ty Davis – Springdale Public Schools	Anthony Owen – Arkansas Department of Education
Jennifer Feltmann – Berryville Public Schools	Kenneth Powell – Metova Federal
Carl Frank – Arkansas School for Mathematics, Sciences, and the Arts	Jerry Prince – EAST Initiative
Charles Gardner – Cyber Innovation Center	Kimberly Raup – Conway Public Schools
Tammy Glass – Spring Hill School District	Sandra Rhone – Mineral Springs School District
Tommy Gober – Cyber Innovation Center	Linda Riley – Wonderview School District
Joel Gordon – Arkansas Regional Innovation Hub	Nicholas Seward – Arkansas School for Mathematics, Sciences, and the Arts
Marilyn Harris – Virtual Arkansas	Tom Simmons – El Dorado Public Schools
Andy Hostetler – Jonesboro Public Schools	Dustin Summey – Virtual Arkansas
Tim Johnston – Arkansas Department of Career Education	Travis Taylor – Little Rock School District
Linda Joplin – Fort Smith Public Schools	Karma Turner – Lake Hamilton School District

The following people contributed to the development of the Robotics Levels 3 and 4 Appendix:

Dametra Conway – Texarkana School District	Anthony Meeker – Hope School District
Kelly Griffin – Southeast Arkansas Education Cooperative	Anthony Owen – Arkansas Department of Education
Tim Johnston – Arkansas Department of Career Education	Jigish Patel – Northwest Arkansas Education Cooperative
Sandra Leiterman – University of Arkansas at Little Rock	Jerry Prince – EAST Initiative
Chris Lynch – Arkansas Department of Career Education	Claire Small – Springdale School District
Blake Matthews – Bentonville School District	