

APPENDIX E

Administrator Overview of the 2019 Computer Science Standards

This snapshot is designed to give the reader a quick overview of the Computer Science (CS) standards and benchmarks (K-12). The full Computer Science Standards document and other appendices can be found at edu.wyoming.gov/standards/computer-science.

There are five domains (core concepts), 16 standards (listed as bold headers), and 130 benchmarks broken out as follows:

- Grades K-2 (18)
- Grades 3-5 (23)
- Grades 6-8 (25)
- HS Level 1 (35)
- HS Level 2 (29)

Computer Science, as defined in the CS Standards document, is the study of computing principles, design, and applications (hardware & software); the creation, access, and use of information through algorithms and problem solving, and the impact of computing on society.

WYOMING 2019 COMPUTER SCIENCE DOMAINS & STANDARDS

Computing Systems	Networks & The Internet	Data Analysis	Algorithms & Programming	Impacts of Computing
CS.D—Devices CS.HS—Hardware & Software CS.T—Troubleshooting	NI.NCO—Network Communication & Organization NI.C—Cybersecurity	DA.S—Storage DA.CVT—Collection, Visualization, & Transformation DA.IM—Inference & Models	AP.A—Algorithms AP.V—Variables AP.C—Control AP.M—Modularity AP.PD—Program Development	IC.C—Culture IC.SI—Social Interactions IC.SLE—Safety, Law, & Ethics

Computing Systems: Devices

K-2	3-5	6-8	9-12 (Level 1)	9-12 (Level 2)
2.CS.D.01 Independently select and use a computing device to perform a variety of tasks for an intended outcome (e.g., create an artifact).	5.CS.D.01 Independently, describe how internal and external parts of computing devices function to form a system.	8.CS.D.01 Recommend improvements to the design of computing devices based on an analysis of how a variety of users interact with the device.	L1.CS.D.01 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.	

Computing Systems: Hardware & Software

K-2	3-5	6-8	9-12 (Level 1)	9-12 (Level 2)
2.CS.HS.01 Demonstrate and describe the function of common components of computing systems (hardware and software) (e.g. use a browser, search engine).	5.CS.HS.01 Model how information is translated, transmitted, and processed in order to flow through hardware and software to accomplish tasks.	8.CS.HS.01 Design and refine a project that combines hardware and software components to collect and exchange data.	L1.CS.HS.01 Explain the interactions between application software, system software, and hardware layers.	L2.CS.HS.01 Categorize the roles of operating system software.

Computing Systems: Troubleshooting

K-2	3-5	6-8	9-12 (Level 1)	9-12 (Level 2)
2.CS.T.01 Recognize computing systems might not work as expected and identify and effectively communicate simple hardware or software problems and implement solutions (e.g., app or program is not working as expected, no sound is coming from the device, caps lock turned on) and discuss problems with peers and adults.	5.CS.T.01 Identify hardware and software problems that may occur during everyday use, then develop, apply, and explain strategies for solving these problems.	8.CS.T.01 Systematically identify, resolve, and document increasingly complex software and hardware problems with computing devices and their components.	L1.CS.T.01 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and resolve errors.	L2.CS.T.01 Identify how hardware components facilitate logic, input, output, and storage in computing systems, and their common malfunctions.

Networks & The Internet: Network Communication & Organization

K-2	3-5	6-8	9-12 (Level 1)	9-12 (Level 2)
2.NI.NCO.01 Identify and describe that computing devices can be connected in a variety of ways (e.g., Bluetooth, Wi-Fi, home and school networks, the internet).	5.NI.NCO.01 Model and explain how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the internet, and reassembled at the destination.	8.NI.NCO.01 Model the role of protocols in transmitting data across networks and the internet (e.g. explain protocols and their importance to data transmission; model how packets are broken down into smaller pieces and how they are delivered).	L1.NI.NCO.01 Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.	L2.NI.NCO.01 Describe the issues that impact network functionality (e.g., bandwidth, load, latency, topology).

Networks & The Internet: Cybersecurity

K-2	3-5	6-8	9-12 (Level 1)	9-12 (Level 2)
2.NI.C.01 Explain what authentication factors are, why we use them, and apply authentication to protect devices and information (personal and private) from unauthorized access.	5.NI.C.01 Discuss real-world cybersecurity problems and identify and implement appropriate strategies for how personal information can be protected.	8.NI.C.01 Critique physical and digital procedures that could be implemented to protect electronic data/information.	L1.NI.C.01 Give examples to illustrate how sensitive data can be affected by malware and other attacks.	L2.NI.C.01 Compare ways software developers protect devices and information from unauthorized access.
		8.NI.C.02 Apply multiple methods of encryption to model the secure transmission of data.	L1.NI.C.02 Recommend cybersecurity measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts.	
			L1.NI.C.03 Compare various security measures, considering trade-offs between the usability and security of a computing system.	
			L1.NI.C.04 Explain trade-offs when selecting and implementing cybersecurity recommendations.	

Data Analysis: Storage

K-2	3-5	6-8	9-12 (Level 1)	9-12 (Level 2)
2.DA.S.01 With guidance, develop and modify an organizational structure by creating, copying, moving, and deleting files and folders.	5.DA.S.01 Justify the format and location for storing data based on sharing requirements and the type of information (e.g., images, videos, text).	8.DA.S.01 Represent data using multiple encoding schemes (e.g., ASCII, binary).	L1.DA.S.01 Translate between different bit representations of real-world phenomena, such as characters, numbers, and images.	
			L1.DA.S.02 Evaluate the trade-offs in how data elements are organized and where data is stored.	

Data Analysis: Collection, Visualization, & Transformation

K-2	3-5	6-8	9-12 (Level 1)	9-12 (Level 2)
2.DA.CVT.01 With guidance, collect data and independently present the same data in various visual formats.	5.DA.CVT.01 Organize and present collected data to highlight relationships and support a claim.	8.DA.CVT.01 Using computational tools, transform collected data to make it more useful and reliable.	L1.DA.CVT.01 Create interactive data representations using software tools to help others better understand real-world phenomena (e.g., paper surveys and online data sets).	L2.DA.CVT.01 Use data analysis tools and techniques to identify patterns in data representing complex systems.
				L2.DA.CVT.02 Select data collection tools and techniques, and use them to generate data sets that support a claim or communicate information.

Data Analysis: Inference & Models

K-2	3-5	6-8	9-12 (Level 1)	9-12 (Level 2)
2.DA.IM.01 With guidance, interpret data and present it in a chart or graph (visualization) in order to make a prediction, with or without a computing device.	5.DA.IM.01 Use data to highlight or propose relationships, predict outcomes, or communicate an idea.	8.DA.IM.01 Refine computational models based on generated data.	L1.DA.IM.01 Create computational models that represent the relationships among different elements of data collected from a phenomenon or process.	L2.DA.IM.01 Formulate, refine, and test scientific hypotheses using models and simulations.

Algorithms & Programming: Algorithms

K-2	3-5	6-8	9-12 (Level 1)	9-12 (Level 2)
2.AP.A.01 With guidance, identify and model daily processes by creating and following algorithms (sets of step-by- step instructions) to complete tasks (e.g., verbally, kinesthetically, with robot devices, or a programming language).	5.AP.A.01 Using grade appropriate content and complexity, compare and refine multiple algorithms for the same task and determine which is the most appropriate.	8.AP.A.01 Create flowcharts and pseudocode to design algorithms to solve complex problems.	L1.AP.A.01 Create a prototype that uses algorithms (e. g., searching, sorting, finding shortest distance) to provide a possible solution for a real-world problem relevant to the student.	L2.AP.A.01 Critically examine and trace classic algorithms. Use and adapt classic algorithms to solve computational problems (e.g., selection sort, insertion sort, binary search, linear search).
			L1.AP.A.02 Describe how artificial intelligence algorithms drive many software and physical systems.	L2.AP.A.02 Develop an artificial intelligence algorithm to play a game against a human opponent or solve a real-world problem.
				L2.AP.A.03 Evaluate algorithms (e.g., sorting, searching) in terms of their efficiency, correctness, and clarity.

Algorithms & Programming: Variables

K-2	3-5	6-8	9-12 (Level 1)	9-12 (Level 2)
<p>2.AP.V.01 Model the way programs store and manipulate data by using numbers or other symbols to represent information (e.g. thumbs up/down as representations of yes/no, arrows when writing algorithms to represent direction, or encode and decode words using numbers, pictographs, or other symbols to represent letters or words).</p>	<p>5.AP.V.01 Using grade appropriate content and complexity, create programs that use variables to store and modify data.</p>	<p>8.AP.V.01 Using grade appropriate content and complexity, create clearly named variables that represent different data types and perform operations on their values.</p>	<p>L1.AP.V.01 Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.</p>	<p>L2.AP.V.01 Compare and contrast simple data structures and their uses (e.g., lists, stacks, queues).</p>

Algorithms & Programming: Control

K-2	3-5	6-8	9-12 (Level 1)	9-12 (Level 2)
<p>2.AP.C.01 With guidance, independently and collaboratively create programs to accomplish tasks using a programming language, robot device, or unplugged activity that includes sequencing, conditionals, and repetition.</p>	<p>5.AP.C.01 Using grade appropriate content and complexity, create programs that include sequences, events, loops, and conditionals, both individually and collaboratively.</p>	<p>8.AP.C.01 Using grade appropriate content and complexity, design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.</p>	<p>L1.AP.C.01 Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.</p>	
			<p>L1.AP.C.02 Trace the execution of loops and conditional statements, illustrating output and changes in values of named variables.</p>	<p>L2.AP.C.01 Trace the execution of recursion, illustrating output and changes in values of named variables.</p>
			<p>L1.AP.C.03 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.</p>	

Algorithms & Programming: Modularity

K-2	3-5	6-8	9-12 (Level 1)	9-12 (Level 2)
<p>2.AP.M.01 Using grade appropriate content and complexity, decompose (breakdown) the steps needed to solve a problem into a precise sequence of instructions (e.g., develop a set of instructions on how to play your favorite game).</p>	<p>5.AP.M.01 Using grade appropriate content and complexity, decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p>	<p>8.AP.M.01 Using grade appropriate content and complexity, decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.</p>	<p>L1.AP.M.01 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.</p>	<p>L2.AP.M.01 Construct solutions to problems using student-created components, such as procedures, modules, and/or objects.</p>
	<p>5.AP.M.02 Using grade appropriate content and complexity, modify, remix, or incorporate portions of an existing program into one's own work to develop something new or add more advanced features.</p>	<p>8.AP.M.02 Using grade appropriate content and complexity, create procedures with parameters to organize code and make it easier to reuse.</p>	<p>L1.AP.M.02 Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.</p>	<p>L2.AP.M.02 Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.</p>
				<p>L2.AP.M.03 Demonstrate code reuse by creating programming solutions using libraries and APIs.</p>

Algorithms & Programming: Program Development

K-2	3-5	6-8	9-12 (Level 1)	9-12 (Level 2)
2.AP.PD.01 Develop plans that describe a program's sequence of events, goals, and expected outcomes.	5.AP.PD.01 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.	8.AP.PD.01 Using grade appropriate content and complexity, seek and incorporate feedback from team members and users to refine a solution to a problem.	L1.AP.PD.01 Plan and develop programs by analyzing a problem and/or process, developing and documenting a solution, testing outcomes, and adapting the program for a variety of users.	L2.AP.PD.01 Plan and develop programs that will provide solutions to a variety of users using a software life cycle process.
2.AP.PD.02 Give credit to ideas, creations, and solutions of others while writing and developing programs.	5.AP.PD.02 Using grade appropriate content and complexity, observe intellectual property rights and give appropriate credit when creating or remixing programs.	8.AP.PD.02 Incorporate existing code, media, and libraries into original programs of increasing complexity and give attribution.	L1.AP.PD.02 Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries.	L2.AP.PD.02 Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (e.g., code documentation) in a group software project.
2.AP.PD.03 Independently and collaboratively debug (identify and fix errors) programs using a programming language.	5.AP.PD.03 Using grade appropriate content and complexity, test and debug (i.e., identify and fix errors) a program or algorithm to ensure it runs as intended.	8.AP.PD.03 Systematically test and refine programs using a range of test cases.	L1.AP.PD.03 Use debugging tools to identify and fix errors in a program.	
			L1.AP.PD.04 Design and develop computational artifacts, working in team roles, using collaborative tools.	L2.AP.PD.03 Develop programs for multiple computing platforms.

<p>2.AP.PD.04 Use correct terminology (debug, program input/output, code) to explain the development of a program or an algorithm (e.g., in an unplugged activity, hands on manipulatives, or a programming language).</p>	<p>5.AP.PD.04 Using grade appropriate content and complexity, describe choices made during program development using code comments, presentations, and demonstrations.</p>	<p>8.AP.PD.04 Using grade appropriate content and complexity, document programs in order to make them easier to follow, test, and debug.</p>	<p>L1.AP.PD.05 Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.</p>	<p>L2.AP.PD.04 Evaluate key qualities of a program through a process such as a code review (e.g., qualities could include correctness, usability, readability, efficiency, portability, and scalability).</p>
	<p>5.AP.PD.05 Using grade appropriate content and complexity, with teacher guidance, perform varying roles when collaborating with peers during the design, implementation, and review stages of program development.</p>	<p>8.AP.PD.05 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.</p>	<p>L1.AP.PD.06 Evaluate and refine computational artifacts to make them more usable and accessible.</p>	<p>L2.AP.PD.05 Develop and use a series of test cases to verify that a program performs according to its design specifications.</p>
				<p>L2.AP.PD.06 Explain security issues that might lead to compromised computer programs.</p>
				<p>L2.AP.PD.07 Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality).</p>

				L2.AP.PD.08 Compare multiple programming languages and discuss how their features make them suitable for solving different types of problems.
--	--	--	--	---

Impacts of Computing: Culture

K-2	3-5	6-8	9-12 (Level 1)	9-12 (Level 2)
2.IC.C.01 Describe how people use different types of technologies in their daily work and personal lives.	5.IC.C.01 Give examples and explain how computing technologies have changed the world and express how those technologies influence and are influenced by cultural practices.	8.IC.C.01 Describe impacts associated with computing technologies that affect people's everyday activities and career options.	L1.IC.C.01 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.	L2.IC.C.01 Evaluate the beneficial and harmful effects that computational artifacts and innovations have on society.
	5.IC.C.02 Develop, test, and refine digital artifacts or devices to improve accessibility and usability for diverse end users.	8.IC.C.02 Describe issues of bias and accessibility in the design of technologies.	L1.IC.C.02 Test and refine computational artifacts to reduce bias and equity deficits.	L2.IC.C.02 Evaluate the impact of equity, access, and influence on the distribution of computing resources in a global society.
			L1.IC.C.03 Demonstrate how a given algorithm applies to problems across disciplines.	L2.IC.C.03 Predict how computational innovations that have revolutionized aspects of our culture might evolve.

Impacts of Computing: Social Interactions

K-2	3-5	6-8	9-12 (Level 1)	9-12 (Level 2)
	5.IC.SI.01 Seek diverse perspectives for the purpose of improving computational artifacts.	8.IC.SI.01 Using grade appropriate content and complexity, collaborate using tools to connect with peers when creating a computational artifact.	L1.IC.SI.01 Use tools and methods for collaboration.	
2.IC.SI.01 Practice grade-level appropriate behavior and responsibilities while participating in an online community. Identify and report inappropriate behavior.	5.IC.SI.02 Practice grade-level appropriate behavior and responsibilities while participating in an online community. Identify and report inappropriate behavior.	8.IC.SI.02 Practice grade-level appropriate behavior and responsibilities while participating in an online community. Identify and report inappropriate behavior.	L1.IC.SI.02 Practice grade-level appropriate behavior and responsibilities while participating in an online community. Identify and report inappropriate behavior.	L2.IC.SI.01 Practice grade-level appropriate behavior and responsibilities while participating in an online community. Identify and report inappropriate behavior.

Impacts of Computing: Safety, Law, & Ethics

K-2	3-5	6-8	9-12 (Level 1)	9-12 (Level 2)
	5.IC.SLE.01 Recognize and appropriately use public domain and creative commons media and discuss the social impact of violating intellectual property rights.	8.IC.SLE.01 Using grade appropriate content and complexity, describe tradeoffs between allowing information to be public and keeping information private and secure.	L1.IC.SLE.01 Explain the beneficial and harmful effects that intellectual property laws can have on innovation.	L2.IC.SLE.01 Debate laws and regulations that impact the development and use of software and technology.
			L1.IC.SLE.02 Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users.	
			L1.IC.SLE.03 Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.	
		8.IC.SLE.02 Using grade level appropriate content and complexity, discuss the legal, social, and ethical impacts associated with software development and use, including both positive and malicious intent.	L1.IC.SLE.04 Using grade level appropriate content and complexity, discuss the legal, social, and ethical impacts associated with software development and use, including both positive and malicious intent.	L2.IC.SLE.02 Using grade level appropriate content and complexity, discuss the legal, social, and ethical impacts associated with software development and use, including both positive and malicious intent.