April 2023

Skill Struck's alignment to

Idaho Content Standards Computer Science

Legend

✅ = Standard aligned

🔹 = Not currently aligned

Standard	Status
Computing Systems (CS)	
K-2.CS.01 Locate and identify computing, input, and output devices in a variety of environments (e.g. desktop and laptop computers, tablets, mobile devices, monitors, keyboards, mouse, printers). (Grades K-2)	✅
K-2.CS.02 Demonstrate how to operate a variety of computing devices (e.g. turn on, navigate, open/close programs or apps). (Grades K-2)	✅
K-2.CS.03 Recognize that software is required to control all computing devices (e.g. programs, browsers, websites, apps). (Grades K-2)	✅
K-2.CS.04 Identify, using accurate terminology, simple hardware and software problems and apply strategies for solving these problems (e.g. rebooting the device, checking the power, access to the network, read error messages, discuss problems with peers and adults). (Grades K-5)	✅
3-5.CS.01	✅

Create code to model intelligent behavior in computing devices (e.g. CS unplugged activities, robot programming). (Grades 3-5)	
3-5.CS.02 Identify, using accurate terminology, simple hardware and software problems and apply strategies for solving these problems (e.g. rebooting the device, checking the power, access to the network, read error messages, discuss problems with peers and adults). (Grades K-5)	✅
6-8.CS.01 Exemplify how computational devices impact the quality of life (both positively and negatively) and enhance the ability of people to perform work, communicate, and interact with others. (Grades 6-8)	✅
6-8.CS.02 Compare and contrast the ways that humans and machines process instructions and sense the world. (Grades 6-8)	✅
6-8.CS.03 Differentiate features of everyday objects that contain computing components (i.e., computing systems that collect, store, analyze, and/or transmit data) (e.g. Kinect, GoPro, smartphone, car). (Grades 6-8)	✅
6-8.CS.04 Apply troubleshooting strategies for solving hardware and software problems (e.g. recognizing, describing, reproducing, isolating, fixing and retesting). (Grades 6-8)	✅
6-8.CS.05 Compare and contrast the capabilities of different hardware and software in computer systems (e.g. processors, display types, input devices, communication, and storage capabilities). (Grades 6-8)	✅
9-10.CS.01 Identify and evaluate what computing resources are required for a given purpose (e.g. system requirements needed to run a program, hardware, and software needed to run game X). (Grades 9-10)	✅
9-10.CS.02 Explore the unique features of embedded computers in areas such as mobile devices, sensors, and vehicles. (Grades 9-10)	✅

9-10.CS.03 Create or modify a program that uses different forms of input and output. (e.g. use voice input instead of text input, use text-to-speech for output) (Grades 9-10)	🔹
9-10.CS.04 Demonstrate the multiple levels of abstraction that support program execution including programming languages, translations, and low-level systems including the fetch-execute cycle (e.g. model, dance, create a play/presentation). (Grades 9-10)	✅
11-12.CS.01 Identify and describe hardware (e.g. physical layers, logic gates, chips, components). (Grades 11-12)	✅
11-12.CS.02 Create a model of how embedded systems sense, process, and actuate in a given environment (e.g. ocean, atmosphere, and highway) (Grades 11-12)	✅
Data and Analysis (DA)	
K-2.DA.01 Classify and sort information into useful order without using a computer (e.g. sorting objects by various attributes). (Grades K-2)	✅
K-2.DA.02 Demonstrate that computing devices save information as data that can be stored, searched, retrieved, modified, and deleted. (Grades K-2)	✅
K-2.DA.03 Explain that networks, like the Internet, link people using computers and other computing devices allowing them to communicate, access, and share information. (Grades K-2)	✅
3-5.DA.01 Use outcome data (results) from running a simulation to solve a problem or answer a question in a core subject area, either individually or collaboratively. (Grades 3-5)	✅
3-5.DA.02 Understand how computers encode and store data (e.g. simple mapping	✅

of binary number to decimal number, letter, or color). (Grades 3-5)	
3-5.DA.03 Gather, manipulate, and evaluate data to explore a real world problem that is of interest to the student. (Grades 3-5)	✅
6-8.DA.01 Describe the trade-off between quality and file size of stored data (e.g. music, video, text, images). (Grades 6-8)	✅
6-8.DA.02 Defend the selection of the data, collection, and analysis needed to answer a question. (Grades 6-8)	✅
6-8.DA.03 Understand that data collection is used to make recommendations to influence decisions as well as predict behavior. List the positive and negative impacts. (Grades 6-8)	✅
6-8.DA.04 Encode and decode information using encryption/decryption schemes. (e.g. Morse code, Unicode, binary, symbols, student-created codes, simple ciphers). (Grades 6-8)	✅
6-8.DA.05 Identify layers of abstraction in different contexts (e.g. video and animation are made of audio and video frames, which are made of pixels, which are made of color codes). (Grades 6-8)	✅
9-10.DA.01 Illustrate how various types of data are stored in a computer system (e.g. how sound and images are stored). (Grades 9-10)	✅
9-10.DA.02 Differentiate between information access and distribution rights (e.g. write, discuss). (Grades 9-10)	✅
9-10.DA.03 Compare and contrast the viewpoints on cybersecurity from the perspective of security experts, privacy advocates, the government (e.g. persuasive essay, presentation, or debate). (Grades 9-12)	✅

9-10.DA.04 Explain the principles of security by examining encryption, cryptography, and authentication techniques. (Grades 9-12)	✅
9-10.DA.05 Apply basic techniques for locating, collecting, and understanding the quality of small and large-scale data sets (e.g. public data sets). (Grades 9-10)	✅
9-10.DA.06 Convert between binary, decimal, octal, and hexadecimal representations of data. (Grades 9-10)	✅
9-10.DA.07 Analyze the representation and trade-offs among various forms of digital information (e.g. lossy versus lossless compression). (Grades 9-10)	✅
9-10.DA.08 Analyze data and identify patterns through modeling and simulation.(Grades 9-12)	✅
11-12.DA.01 Use data analysis to enhance understanding and gain knowledge of complex systems to show the transformation from data to information to knowledge (e.g. using existing data sets). (Grades 11-12)	✅
11-12.DA.02 Use various data collection techniques for different types of problems (e.g. mobile device GPS, user surveys, embedded system sensors, open data sets, social media data sets). (Grades 11-12)	✅
11-12.DA.03 Understand and explain security policies by comparing encryption and authentication strategies (e.g. trapdoor functions and man in the middle attacks). (Grades 11-12)	✅
11-12.DA.04 Discuss the variety of interpretations of binary sequences (e.g. instructions, numbers, text, sound, image). (Grades 11-12)	✅
11-12.DA.05	✅

Use models and simulations to help formulate, refine, and test scientific hypotheses. (Grades 11-12)	
11-12.DA.06 Analyze data and identify patterns through modeling and simulation.(Grades 9-12)	✅
Impacts of Computing (IC)	
K-2.IC.01 Practice responsible digital citizenship (legal and ethical behaviors) in the use of technology systems and software. (Grades K-5)	✅
K-2.IC.02 Understand that a wide range of jobs require knowledge or use of computer science. (Grades K-2)	✅
3-5.IC.01 Practice responsible digital citizenship (legal and ethical behaviors) in the use of technology systems and software. (Grades K-5)	✅
3-5.IC.02 Explore the connections between computer science and other fields. (Grades 3-5)	✅
3-5.IC.03 Generate examples of how the use of computing can affect society and how society can influence the use of computing. (Grades 3-5)	✅
3-5.IC.04 Explain ethical issues that relate to computers and networks (e.g. equity of access, security, privacy, copyright, digital citizenship, and intellectual property). (Grades 3-5)	✅
3-5.IC.05 Evaluate the positive and negative impacts of computing devices in daily life. (e.g., downloading videos and audio files, electronic appliances, wireless Internet, mobile computing devices, GPS systems, Internet of Things, wearable computing). Describe the pros and cons of these impacts.(Grades 3-5)	✅

6-8.IC.01 Explore security risks associated with using weak passwords, lack of encryption and/or insecure transactions. (Grades 6-8)	✅
6-8.IC.02 Explore how computer science fosters innovation and enhances other careers and disciplines. (Grades 6-8)	✅
6-8.IC.03 Describe ethical issues that relate to computers and networks (e.g. equity of access, security, privacy, ownership and information sharing, copyright, licensing). (Grades 6-8)	✅
6-8.IC.04 Explore how the Internet impacts global communication and collaboration. (Grades 6-8)	✅
6-8.IC.05 Design, develop, and present computational artifacts that have a positive social impact (e.g. web pages, mobile applications, animations). (Grades 6-8)	✅
6-8.IC.06 Redesign user interfaces to be more inclusive, accessible, and minimizing the impact of the designer's inherent bias. (e.g. web pages, mobile applications, animations). (Grades 6-8)	✅
6-8.IC.07 Understand and explain the elements of federal, state, and local regulations that relate to digital citizenship (e.g. COPPA, CIPA, state laws, district policies). (Grades 6-8)	🔹
6-8.IC.08 Summarize current events and changes resulting from computing and their effects on education, the workplace, and society. (Grades 6-8)	🔹
6-8.IC.09 Predict positive and negative social impacts of existing or student created content and computational artifacts (e.g. economic, entertainment, education, or political). (Grades 6-8)	✅

9-10.IC.01 Explain the social and economic implications associated with unethical computing practices (e.g. software piracy, intrusion, malware, current corporate fraud examples). (Grades 9-10)	✅
9-10.IC.02 Discuss trade-offs such as privacy, safety, and convenience associated with the collection and large scale analysis of information about individuals (e.g. social media, online shopping, how grocery/dept stores collect and use personal data). (Grades 9-10)	✅
9-10.IC.03 Understand and explain the impact of artificial intelligence and robotics. (Grades 9-10)	✅
9-10.IC.04 Describe how computer science shares features with creating and designing an artifact such as in music and art. (Grades 9-12)	✅
9-10.IC.05 Demonstrate how computing enhances traditional forms and enables new forms of experience, expression, communication, and collaboration (e.g. virtual reality). (Grades 9-10)	✅
9-10.IC.06 Explain the impact of the digital divide on access to critical information (e.g. education, healthcare, medical records, access to training). (Grades 9-10)	✅
9-10.IC.0 7 Compare the positive and negative impacts of computing on behavior and culture. (Grades 9-10)	✅
9-10.IC.08 Evaluate a computational artifact for its effectiveness for universal access (e.g. compare sample code with accessibility standards, building in access from initial design). (Grades 9-10)	✅
9-10.IC.09 Practice responsible digital citizenship (legal and ethical behaviors) in the use of technology systems and software. (Grades 9-10)	✅

9-10.IC.10 Explain how computer science fosters innovation and enhances other careers and disciplines. (Grades 6-8)	✅
9-10.IC.11 Explain the impacts of computing on business, manufacturing, commerce, and society. (Grades 9-12)	✅
11-12.IC.01 Understand the ecosystem of open source software development and its impact on global collaboration through an open-source software project (e.g. https://codein.withgoogle.com). (Grades 11-12)	🔹
11-12.IC.02 Debate laws and regulations that impact the development and use of software. (e.g. compare and contrast licensing versus certification, professional societies, professional code of ethics). (Grades 11-12)	✅
11-12.IC.03 Research, analyze, and present how computational thinking has revolutionized an aspect of our culture (e.g. agriculture, communication, work, healthcare, music, art). (Grades 11-12)	✅
11-12.IC.04 Analyze the role and impact of government regulation on privacy and security. (Grades 11-12)	✅
11-12.IC.05 Debate how the issues of equity, access, and power relate to the distribution of computing resources in a global society. (Grades 11-12)	✅
11-12.IC.06 Identify and evaluate the beneficial and harmful effects of computing innovations. (Grades 11-12)	✅
11-12.IC.07 Practice responsible digital citizenship (legal and ethical behaviors) in the use of technology systems and software. (Grades 11-12)	✅
11-12.IC.08 Describe how computer science shares features with creating and	✅

designing an artifact such as in music and art. (Grades 9-12)	
11-12.IC.09 Explain the impacts of computing on business, manufacturing, commerce, and society. (Grades 9-12)	✅
11-12.IC.10 Summarize how computer automation and control is transforming society and the global economy (e.g. financial markets, transactions, predictions). (Grades 11-12)	✅
Networks and the Internet (NI)	
3-5.NI.01 Demonstrate how a device on a network sends and receives information. (Grades 3-5)	✅
6-8.NI.01 Simulate the flow of information as packets on the Internet and networks (e.g. model using strings and paper, note passing). (Grades 6-8)	✅
6-8.NI.02 Compare and contrast the trade-offs between physical (wired), wireless, and mobile networks (e.g. speed, security, and cost). (Grades 6-8)	✅
9-10.NI.01 Describe the underlying process of Internet-based services. (e.g. illustrate how information flows in a global network, servers and clients, cloud services, secure versus insecure communication). (Grades 9-10)	✅
9-10.NI.02 Illustrate the basic components of computer networks, protocols and routing (e.g. team based activities which may include drawing a diagram of a network including routers, switches, local networks, and end user computing devices, creating models with string and paper, see CS unplugged activities). (Grades 9-10)	✅
11-12.NI.01 Simulate and discuss the issues that impact network functionality (e.g. use ns3 or other free network simulators). (Grades 11-12)	✅

11-12.NI.02 Examine how encryption is essential to ensuring privacy and security over the internet. (Grades 11-12)	✅
Algorithms and Programming (AP)	
K-2.AP.01 Construct and test problem solutions using a block-based visual programming language, both independently and collaboratively (e.g. pair programming). (Grades K-5)	✅
K-2.AP.02 Create a design document to illustrate thoughts, ideas, and stories in a sequential manner (e.g., storyboard, mind map). (Grades K-2)	✅
K-2.AP.03 Construct an algorithm to accomplish a task, both independently and collaboratively. (Grades K-5)	✅
K-2.AP.04 Follow the sequencing in an algorithm. (Grades K-2)	✅
3-5.AP.01 Identify and understand ways that teamwork and collaboration can support problem solving and the software design cycle. (Grades 3-5)	✅
3-5.AP.02 Construct and test problem solutions using a block-based visual programming language, both independently and collaboratively (e.g. pair programming). (Grades K-5)	✅
3-5.AP.03 Generate a list of sub-problems to consider while addressing a larger problem. (Grades 3-5)	✅
3-5.AP.04 Understand that computer program design is an iterative process that includes the following steps: define the problem, generate ideas, build a program, test the program, improve the program. (Grades 3-5)	✅
3-5.AP.05	✅

Understand, explain and debug the sequencing in an algorithm. (Grades 3-5)	
3-5.AP.06 Construct and test problem solutions using a block-based visual programming language, both independently and collaboratively (e.g. pair programming). (Grades K-5)	✅
3-5.AP.07 Construct an algorithm to accomplish a task, both independently and collaboratively. (Grades K-5)	✅
6-8.AP.01 Solicit, evaluate, and integrate peer feedback as appropriate to develop or refine a product. (Grades 6-8)	✅
6-8.AP.02 Compare different algorithms that may be used to solve the same problem by time and space efficiency. (Grades 6-8)	✅
6-8.AP.03 Interpret, modify, and analyze content-specific models used to run simulations (e.g. ecosystems, epidemics, spread of ideas) . (Grades 6-8)	✅
6-8.AP.04 Apply an iterative design process (define the problem, generate ideas, build, test, and improve solutions) in problem solving, both individually and collaboratively. (Grades 6-8)	✅
6-8.AP.05 Create, analyze, and modify control structures to create programming solutions. (Grades 6-8)	✅
6-8.AP.06 Predict the outcome of an algorithm and then step through it to verify your predictions. (Grades 6-8)	✅
6-8.AP.07 Decompose a problem into sub-problems and demonstrate how the parts can be synthesized to create a solution. (Grades 6-8)	✅

6-8.AP.08 Evaluate the correctness of a program by collecting and analyzing data generated from multiple runs of the program. (Grades 6-8)	✅
6-8.AP.09 Use debugging and testing to improve program quality. (Grades 6-8)	✅
9-10.AP.01 Design and develop a software artifact by leading, initiating, and participating in a team (e.g. pair programming, agile software development). (Grades 9-12)	✅
9-10.AP.02 Demonstrate how diverse collaboration, both inside and outside of a team, impacts the design and development of software products (e.g. students show their own artifacts and demonstrate and reflect how diverse collaboration made a product better). (Grades 9-12)	✅
9-10.AP.03 Compare a variety of programming languages available to solve problems and develop systems. (Grades 9-10)	✅
9-10.AP.04 Explore security issues that might lead to compromised computer programs (e.g. ambiguous function calls, lack of error checking of the input, buffer overflow, SQL injection attacks, denial of service attacks). (Grades 9-12)	✅
9-10.AP.05 Classify and define the different types of software licenses in order to understand how to apply each one to a specific software example. (Grades 9-12)	✅
9-10.AP.06 Understand the notion of hierarchy and abstraction in high-level languages, translation, instruction sets, and logic circuits. (Grades 9-10)	✅
9-10.AP.07 Explore issues surrounding mobile computing by creating a mobile computing application (e.g. App Inventor). (Grades 9-10)	🔹

9-10.AP.08 Create software solutions by applying analysis, design, implementation and testing techniques. (Grades 9-10)	✅
9-10.AP.09 Demonstrate code reuse by creating programming solutions using APIs and libraries (e.g. using text to speech in App Inventor, using Twitter API). (Grades 9-10)	✅
9-10.AP.10 Illustrate the flow of execution and output of a given program (e.g. flow and control diagrams). (Grades 9-10)	✅
9-10.AP.11 Illustrate how mathematical and statistical functions, sets, and logic are used in computation. (Grades 9-10)	✅
9-10.AP.12 Design algorithms using sequence, selection, iteration and recursion. (Grades 9-10)	✅
9-10.AP.13 Explain, represent, and understand natural phenomena using modeling and simulation (Grade 9-10).	✅
9-10.AP.14 Describe the concept of parallel processing as a strategy to solve large problems. (Grades 9-10)	✅
9-10.AP.15 Compare and evaluate software development processes used to solve problems (e.g. waterfall, agile). (Grades 9-10)	✅
9-10.AP.16 Decompose a complex problem into simpler parts using predefined functions and parameters, classes, and methods. (Grades 9-10)	✅
9-10.AP.17 Demonstrate the value of abstraction to manage problem complexity. (Grades 9- 10)	✅

9-10.AP.18 Evaluate and improve program quality using various debugging and testing methods and examine the difference between verification and validation. (Grades 9-12)	✅
9-10.AP.19 Evaluate programs written by others for readability and usability. (Grades 9-10)	✅
11-12.AP.01 Analyze the notion of intelligent behavior through programs that learn and adapt, play games, do image recognition, perform text analysis, and control the behavior of robots. (Grades 11-12)	✅
11-12.AP.02 Create collaborative software projects using version control systems, Integrated Development Environments (IDEs), and collaborative tools. (Grades 11-12)	✅
11-12.AP.03 Demonstrate an understanding of the software life cycle process (e.g. by participating on a software project team). (Grades 11-12)	✅
11-12.AP.04 Modify an existing program to add additional functionality and discus the positive and negative implications (e.g., breaking other functionality). (Grades 11-12)	✅
11-12.AP.05 Explain the value of heuristic algorithms to approximate solutions for intractable problems. (Grades 11-12)	✅
11-12.AP.06 Decompose a computational problem through data abstraction and modularity. (Grades 9-12)	✅
11-12.AP.07 Critically examine algorithms and design an original algorithm (e.g. adapt, remix, improve). (Grades 11-12)	✅
11-12.AP.08	✅

Evaluate efficiency, correctness, and clarity of algorithms. (Grades 11-12)	
11-12.AP.09 Compare and contrast simple data structures and their uses (e.g. arrays, lists, stacks, queues, maps, trees, graphs). (Grades 11-12)	✅
11-12.AP.10 Decompose a problem by creating functions and classes. (Grades 11-12)	✅
11-12.AP.11 Use variable scope and encapsulation to design programs with cohesive and decoupled components. (Grades 9-12)	✅
11-12.AP.12 Classify problems as tractable, intractable, or computationally unsolvable. (Grades 11-12)	🔹
11-12.AP.13 Understand and explain the use of concurrency (e.g. separate processes into threads and divide data into parallel streams, have students self sort by height). (Grades 11-12)	✅
11-12.AP.14 Evaluate the qualities of a program such as correctness, usability, readability, efficiency, portability and scalability through a process such as a code review. (Grades 11-12)	✅