Skill Struck ®

Skill Struck's alignment to

California's K-12 Computer Science Standards

Legend

✅ = Standard aligned

🔷 = Not currently aligned

Standard	Status
K-2.CS.1 Select and operate computing devices that perform a variety of tasks accurately and quickly based on user needs and preferences.	✅
K-2.CS.2 Explain the functions of common hardware and software components of computing systems.	✅
K-2.CS.3 Describe basic hardware and software problems using accurate terminology.	✅
K-2.NI.4 Model and describe how people connect to other people, places, information and ideas through a network.	✅
K-2.NI.5	✅

Explain why people use passwords	
K-2.NI.6 Create patterns to communicate a message.	🔷
K-2.DA.7 Store, copy, search, retrieve, modify, and delete information using a computing device, and define the information stored as data.	✅
K-2.DA.8 Collect and present data in various visual formats.	✅
K-2.DA.9 Identify and describe patterns in data visualizations, such as charts or graphs, to make predictions.	✅
K-2.AP.10 Model daily processes by creating and following algorithms to complete tasks.	✅
K-2.AP.11 Model the way programs store data.	✅
K-2.AP.12 Create programs with sequences of commands and simple loops, to express ideas or address a problem.	✅
K-2.AP.13 Decompose the steps needed to solve a problem into a sequence of instructions.	✅
K-2.AP.14 Develop plans that describe a program's sequence of events, goals, and expected outcomes.	✅
K-2.AP.15 Give attribution when using the ideas and creations of others while	✅

developing programs.	
K-2.AP.16 Debug errors in an algorithm or program that includes sequences and simple loops.	✅
K-2.AP.17 Describe the steps taken and choices made during the iterative process of program development.	✅
K-2.IC.18 Compare how people lived and worked before and after the adoption of new computing technologies.	✅
K-2.IC.19 Work respectfully and responsibly with others when communicating electronically.	✅
K-2.IC.20 Describe approaches and rationales for keeping login information private, and for logging off of devices appropriately.	✅
3-5.CS.1 Describe how computing devices connect to other components to form a system.	✅
3-5.CS.2 Demonstrate how computer hardware and software work together as a system to accomplish tasks.	✅
3-5.CS.3 Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies.	✅
3-5.NI.4 Model how information is broken down into smaller pieces, transmitted as	✅

packets through multiple devices over networks and the Internet, and reassembled at the destination.	
3-5.NI.5 Describe physical and digital security measures for protecting personal information.	✅
3-5.NI.6 Create patterns to protect information from unauthorized access.	🔷
3-5.DA.7 Explain that the amount of space required to store data differs based on the type of data and/or level of detail.	✅
3-5.DA.8 Organize and present collected data visually to highlight relationships and support a claim.	✅
3-5.DA.9 Use data to highlight and/or propose relationships, predict outcomes, or communicate ideas.	✅
3-5.AP.10 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.	✅
3-5.AP.11 Create programs that use variables to store and modify data.	✅
3-5.AP.12 Create programs that include events, loops, and conditionals.	✅
3-5.AP.13 Decompose problems into smaller, manageable tasks which may themselves be decomposed.	✅
3-5.AP.14	✅

Create programs by incorporating smaller portions of existing programs, to develop something new or add more advanced features.	
3-5.AP.15 Use an iterative process to plan and develop a program by considering the perspectives and preferences of others.	✅
3-5.AP.16 Observe intellectual property rights and give appropriate attribution when creating, remixing, or combining programs.	✅
3-5.AP.17 Test and debug a program or algorithm to ensure it accomplishes the intended task.	✅
3-5.AP.18 Perform different roles when collaborating with peers during the design, implementation, and review stages of program development.	✅
3-5.AP.19 Describe choices made during program development using code comments, presentations, and demonstrations.	✅
3-5.IC.20 Discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices.	✅
3-5.IC.21 Propose ways to improve the accessibility and usability of technology products for the diverse needs and wants of users.	✅
3-5.IC.22 Seek and explain the impact of diverse perspectives for the purpose of improving computational artifacts.	✅

3-5.IC.23 Describe reasons creators might limit the use of their work.	✅
6-8.CS.1 Design modifications to computing devices in order to improve the ways users interact with the devices.	✅
6-8.CS.2 Design a project that combines hardware and software components to collect and exchange data.	✅
6-8.CS.3 Systematically apply troubleshooting strategies to identify and resolve hardware and software problems in computing systems.	✅
6-8.NI.4 Model the role of protocols in transmitting data across networks and the Internet.	✅
6-8.NI.5 Explain potential security threats and security measures to mitigate threats.	✅
6-8.NI.6 Apply multiple methods of information protection to model the secure transmission of information.	✅
6-8.DA.7 Represent data in multiple ways.	✅
6-8.DA.8 Collect data using computational tools and transform the data to make it more useful.	✅
6-8.DA.9 Test and analyze the effects of changing variables while using	✅

computational models.	
6-8.AP.10 Use flowcharts and/or pseudocode to design and illustrate algorithms that solve complex problems.	✅
6-8.AP.11 Create clearly named variables that store data, and perform operations on their contents.	✅
6-8.AP.12 Design and iteratively develop programs that combine control structures and use compound conditions.	✅
6-8.AP.13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	✅
6-8.AP.14 Create procedures with parameters to organize code and make it easier to reuse.	✅
6-8.AP.15 Seek and incorporate feedback from team members and users to refine a solution that meets user needs.	✅
6-8.AP.16 Incorporate existing code, media, and libraries into original programs, and give attribution.	✅
6-8.AP.17 Systematically test and refine programs using a range of test cases.	✅
6-8.AP.18 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.	✅

6-8.AP.19 Document programs in order to make them easier to use, read, test, and debug.	✅
6-8.IC.20 Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.	✅
6-8.IC.21 Discuss issues of bias and accessibility in the design of existing technologies.	✅
6-8.IC.22 Collaborate with many contributors when creating a computational artifact.	✅
6-8.IC.23 Compare tradeoffs associated with licenses for computational artifacts to balance the protection of the creators' rights and the ability for others to use and modify the artifacts.	✅
6-8.IC.24 Compare tradeoffs between allowing information to be public and keeping information private and secure.	✅
9-12.CS.1 Describe ways in which abstractions hide the underlying implementation details of computing systems to simplify user experiences.	✅
9-12.CS.2 Compare levels of abstraction and interactions between application software, system software, and hardware.	✅
9-12.CS.3 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.	✅

9-12.NI.4 Describe issues that impact network functionality.	✅
9-12.NI.5 Describe the design characteristics of the Internet.	✅
9-12.NI.6 Compare and contrast security measures to address various security threats.	✅
9-12.NI.7 Compare and contrast cryptographic techniques to model the secure transmission of information.	✅
9-12.DA.8 Translate between different representations of data abstractions of real-world phenomena, such as characters, numbers, and images.	✅
9-12.DA.9 Describe tradeoffs associated with how data elements are organized and stored.	✅
9-12.DA.10 Create data visualizations to help others better understand real-world phenomena.	✅
9-12.DA.11 Refine computational models to better represent the relationships among different elements of data collected from a phenomenon or process.	✅
9-12.AP.12 Design algorithms to solve computational problems using a combination of original and existing algorithms.	✅
9-12.AP.13 Create more generalized computational solutions using collections	✅

instead of repeatedly using simple variables.	
9-12.AP.14 Justify the selection of specific control structures by identifying tradeoffs associated with implementation, readability, and performance.	✅
9-12.AP.15 Iteratively design and develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.	✅
9-12.AP.16 Decompose problems into smaller subproblems through systematic analysis, using constructs such as procedures, modules, and/or classes.	✅
9-12.AP.17 Create computational artifacts using modular design.	✅
9-12.AP.18 Systematically design programs for broad audiences by incorporating feedback from users.	✅
9-12.AP.19 Explain the limitations of licenses that restrict use of computational artifacts when using resources such as libraries.	✅
9-12.AP.20 Iteratively evaluate and refine a computational artifact to enhance its performance, reliability, usability, and accessibility.	✅
9-12.AP.21 Design and develop computational artifacts working in team roles using collaborative tools.	✅
9-12.AP.22 Document decisions made during the design process using text, graphics,	✅

presentations, and/or demonstrations in the development of complex programs.	
9-12.IC.23 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.	✅
9-12.IC.24 Identify impacts of bias and equity deficit on design and implementation of computational artifacts and apply appropriate processes for evaluating issues of bias.	✅
9-12.IC.25 Demonstrate ways a given algorithm applies to problems across disciplines.	✅
9-12.IC.26 Study, discuss, and think critically about the potential impacts and implications of emerging technologies on larger social, economic, and political structures, with evidence from credible sources.	✅
9-12.IC.27 Use collaboration tools and methods to increase connectivity with people of different cultures and careers.	✅
9-12.IC.28 Explain the beneficial and harmful effects that intellectual property laws can have on innovation.	✅
9-12.IC.29 Explain the privacy concerns related to the collection and generation of data through automated processes.	✅
9-12.IC.30 Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.	✅

9-12S.CS.1 Illustrate ways computing systems implement logic through hardware components.	✅
9-12S.CS.2 Categorize and describe the different functions of operating system software.	✅
9-12S.NI.3 Examine the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.	✅
9-12S.NI.4 Explain how the characteristics of the Internet influence the systems developed on it.	✅
9-12S.NI.5 Develop solutions to security threats.	✅
9-12S.NI.6 Analyze cryptographic techniques to model the secure transmission of information.	✅
9-12S.DA.7 Select and use data collection tools and techniques to generate data sets.	✅
9-12S.DA.8 Use data analysis tools and techniques to identify patterns in data representing complex systems.	✅
9-12S.DA.9 Evaluate the ability of models and simulations to test and support the refinement of hypotheses.	✅
9-12S.AP.10 Describe how artificial intelligence drives many software and physical	✅

systems.	
9-12S.AP.11 Implement an algorithm that uses artificial intelligence to overcome a simple challenge.	✅
9-12S.AP.12 Implement searching and sorting algorithms to solve computational problems.	✅
9-12S.AP.13 Evaluate algorithms in terms of their efficiency.	✅
9-12S.AP.14 Compare and contrast fundamental data structures and their uses.	✅
9-12S.AP.15 Demonstrate the flow of execution of a recursive algorithm.	✅
9-12S.AP.16 Analyze a large-scale computational problem and identify generalizable patterns or problem components that can be applied to a solution.	✅
9-12S.AP.17 Construct solutions to problems using student-created components, such as procedures, modules, and/or objects.	✅
9-12S.AP.18 Demonstrate code reuse by creating programming solutions using libraries and APIs.	✅
9-12S.AP.19 Plan and develop programs for broad audiences using a specific software life cycle process.	✅
9-12S.AP.20 Develop programs for multiple computing platforms.	✅

9-12S.AP.21 Identify and fix security issues that might compromise computer programs.	✅
9-12S.AP.22 Develop and use a series of test cases to verify that a program performs according to its design specifications.	✅
9-12S.AP.23 Modify an existing program to add additional functionality and discuss intended and unintended implications.	✅
9-12S.AP.24 Evaluate key qualities of a program through a process such as a code review.	✅
9-12S.AP.25 Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (e.g., code documentation) while developing software within a group.	✅
9-12S.AP.26 Compare multiple programming languages, and discuss how their features make them suitable for solving different types of problems.	✅
9-12S.IC.27 Evaluate computational artifacts with regard to improving their beneficial effects and reducing harmful effects on society.	✅
9-12S.IC.28 Evaluate how computational innovations that have revolutionized aspects of our culture might evolve.	✅
9-12S.IC.29 Evaluate the impact of equity, access, and influence on the distribution of computing resources in a global society.	✅

9-12S.IC.30 Debate laws and regulations that impact the development and use of software.	✅