Skill Struck ®

Skill Struck's alignment to

North Carolina's K-12 Computer Science Standards

Legend

✅ = Standard aligned

🔷 = Not currently aligned

Standard	Status
K2-CS-01 Choose appropriate devices to perform a variety of classroom tasks.	✅
K2-CS-02 Describe the function of common physical components of computing systems (hardware) with appropriate terminology.	✅
K2-CS-03 Operate appropriate software to perform a variety of tasks.	✅
K2-CS-04 Describe basic hardware and software problems with accurate terminology	✅
K2-NI-01 Illustrate how information is broken down into smaller pieces and can be reassembled.	✅

K2-NI-02 Apply knowledge of what passwords are and why we use strong passwords to protect devices and information from unauthorized access.	✅
K2-NI-03 Discover your digital footprint and how personal information can be protected.	✅
K2-DA-01 Store, copy, search, retrieve, modify, and delete information using a computing device.	✅
K2-DA-02 Define information stored on a computing device as data.	✅
K2-DA-03 Collect and present the same data in various visual formats.	✅
K2-DA-04 Make predictions with patterns in data visualizations	✅
K2-AP-01 Model daily processes with algorithms to complete tasks.	✅
K2-AP-02 Demonstrate how programs store and manipulate data by using numbers or other symbols to represent information	✅
K2-AP-03 Develop programs with sequences and simple loops to express ideas or address a problem.	✅
K2-AP-04 Decompose the steps needed to solve a problem into a precise sequence of instructions.	✅
K2-AP-05	✅

Develop plans that describe a program's sequence of events, goals, and expected outcomes.	
K2-AP-06 Give attribution when using the ideas and creations of others while developing programs.	✅
K2-AP-07 Identify and debug errors in an algorithm or program that includes sequences and simple loops.	✅
K2-AP-08 Using correct terminology, describe steps taken and choices made during the iterative process of program development.	✅
K2-IC-01 Compare how people live and work before and after the implementation or adoption of new computing technology	✅
K2-IC-02 Select software that meets the diverse needs and preferences for the technology individuals use in the classroom.	✅
K2-IC-03 Work respectfully and responsibly with others online.	✅
K2-IC-04 Model responsible login and logoff procedures on all devices.	✅
35-CS-01 Evaluate the features available on digital devices to perform a variety of classroom tasks.	✅
35-CS-02 Model how computer hardware and software work together as a system to accomplish tasks.	✅

35-CS-03 Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies	✅
35-NI-01 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.	✅
35-NI-02 Explain your digital footprint and how personal information can be protected	✅
35-DA-01 Identify the type of data encoded in a file based on file extension.	✅
35-DA-02 Illustrate the process of file management and version control.	✅
35-DA-03 Organize and present collected data visually to highlight relationships and support a claim.	✅
35-DA-04 Communicate using data to highlight or predict outcomes.	✅
35-AP-01 Create multiple algorithms for the same task to determine which is the most accurate and efficient.	✅
35-AP-02 Create programs that use variables to store and modify data.	✅
35-AP-03 Construct programs that include sequences	✅
35-AP-04	✅

Construct programs using simple loops.	
35-AP-05 Construct programs that implement conditionals.	✅
35-AP-06 Decompose problems into smaller, manageable, subproblems to facilitate the program development process.	✅
35-AP-07 Modify, remix, or incorporate portions of an existing program into one's own work.	✅
35-AP-08 Apply an iterative process to the development of a program by including diverse perspectives and considering user preferences.	✅
35-AP-09 Give appropriate attribution when creating or remixing programs while respecting intellectual property rights	✅
35-AP-10 Identify and debug errors in an algorithm or program to ensure it runs as intended.	✅
35-AP-11 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.	✅
35-AP-12 Describe choices made during program development using code comments, presentations, and demonstrations.	✅
35-IC-01 Compare computing technologies that have changed the world and how	✅

they both influence and are influenced by cultural practices.	
35-IC-02 Explore the tools that can be used to improve accessibility and usability of technology products for the diverse needs and wants of users.	✅
35-IC-03 Seek diverse perspectives with collaboration for the purpose of improving computational artifacts.	✅
35-IC-04 Exhibit positive digital citizenship and social responsibility in online interactions.	✅
35-IC-05 Utilize public domain or creative commons media, and refrain from copying or using material created by others without permission.	✅
68-CS-01 Understand the design of computing devices based on an analysis of how users interact with the devices.	✅
68-CS-02 Design projects that combine hardware and software components to collect and exchange data.	✅
68-CS-03 Systematically identify and fix problems with computing devices and components.	✅
68-NI-01 Analyze different ways that data is transferred across a network and the role of protocols in transmitting data.	✅
68-NI-02 Explain how physical and digital security measures protect electronic	✅

information.	
68-NI-03 Explain permission and authorizations to access resources to computer systems online	✅
68-NI-04 Apply multiple methods of encryption to model the secure transmission of information.	✅
68-DA-01 Represent data using multiple encoding schemes.	✅
68-DA-02 Collect data using computational tools.	✅
68-DA-03 Transform the collected data to make it more useful and reliable.	✅
68-DA-04 Refine computational models based on the data they have generated and/or data collected.	✅
68-AP-01 Implement flowcharts and/or pseudocode to address complex problems as algorithms.	✅
68-AP-02 Create clearly named variables that represent different data types.	✅
68-AP-03 Design and iteratively develop programs that combine control structures including nested loops and compound conditionals.	✅
68-AP-04 Construct programs that include events.	✅

68-AP-05 Organize problems and subproblems into parts.	✅
68-AP-06 Explain the design, implementation, and review of programs.	✅
68-AP-07 Create procedures with parameters to organize code and make it easier to reuse groups of instructions.	✅
68-AP-08 Assess feedback from team members and users to refine a solution that meets user needs.	✅
68-AP-09 Incorporate existing code and media into original programs and give attribution.	✅
68-AP-10 Systematically test and refine programs using a range of test cases.	✅
68-AP-11 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.	✅
68-AP-12 Document programs in order to make them easier to follow, test, and debug.	✅
68-IC-01 Compare tradeoffs associated with computing technologies that affect everyday activities and career options.	✅
68-IC-02 Describe how equity, access, and influence impact the distribution of computing resources in a global society.	✅

68-IC-03 Discuss issues of bias and accessibility in the design of existing technologies.	✅
68-IC-04 Collaborate, model, and promote effective research strategies for assessing and evaluating innovative resources.	✅
68-IC-05 Collaborate with many contributors to create a computational artifact.	✅
68-IC-06 Utilize tools and methods for collaboration on a project to increase connectivity of peers.	✅
68-IC-07 Examine the benefits and drawbacks of a digital footprint and online identity.	✅
68-IC-08 Understand how online interactions make an impact on the social, emotional, and physical aspects of others	✅
68-IC-09 Compare tradeoffs between allowing information to be public and keeping information private and secure.	✅
68-IC-10 Explore how laws and regulations impact the development and use of software.	✅
ICS-CS-01 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.	✅

ICS-CS-02 Compare levels of abstraction and interactions between application software, system software, and hardware layers	✅
ICS-CS-03 Explain the roles of operating systems including memory management, data storage/retrieval, process management, and access control	✅
ICS-CS-04 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.	✅
ICS-NI-01 Evaluate the relationship between routers, switches, servers, and topology with regard to networks.	✅
ICS-NI-02 Identify examples to illustrate how sensitive data can be affected by malware and other attacks.	✅
ICS-NI-03 Recommend cybersecurity measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts.	✅
ICS-NI-04 Compare various security measures and consider tradeoffs between the usability and security of a computing system.	✅
ICS-DA-01 Compare different binary representations of data, including text, sound, images, and numbers.	✅
ICS-DA-02 Evaluate the tradeoffs in how data elements are organized and where data is stored.	✅

ICS-DA-03 Create interactive data visualizations using software tools to help others better understand real-world phenomena.	✅
ICS-DA-04 Create computational models that represent the relationships among different elements of data collected.	✅
ICS-AP-01 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests	✅
ICS-AP-02 Explain the use of artificial intelligence within computing systems.	✅
ICS-AP-03 Utilize lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.	✅
ICS-AP-04 Justify the selection of specific control structures, considering implementation, readability, and program performance.	✅
ICS-AP-05 Iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.	✅
ICS-AP-06 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects	✅
ICS-AP-07 Create artifacts by using procedures within a program, combinations of data and procedures, or	✅

independent but interrelated programs.	
ICS-AP-08 Systematically design programs for broad audiences.	✅
ICS-AP-09 Refine programs by incorporating feedback from users.	✅
ICS-AP-10 Evaluate licenses that limit or restrict use of computational artifacts when using resources such as software libraries.	✅
ICS-AP-11 Use data analysis tools and techniques to identify patterns in data representing complex systems.	✅
ICS-AP-12 Modify computational artifacts to increase usability and accessibility.	✅
ICS-AP-13 Develop computational artifacts working in team roles using collaborative tools.	✅
ICS-AP-14 Explain design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.	✅
ICS-IC-01 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.	✅
ICS-IC-02 Elaborate how computational innovations have and may continue to impact society.	✅
ICS-IC-03 Evaluate how equity, access, and influence impact distribution of	✅

computing resources in a global society.	
ICS-IC-04 Test computational artifacts to reduce bias and equity deficits.	✅
ICS-IC-05 Demonstrate ways a given algorithm applies to problems across disciplines.	✅
ICS-IC-06 Utilize tools and methods for collaboration on a project to increase connectivity of peers.	✅
ICS-IC-07 Explain the beneficial and harmful effects that intellectual property laws can have on innovation.	✅
ICS-IC-08 Explain privacy concerns related to the collection and generation of data through automated processes that may not be evident to users.	✅
ICS-IC-09 Evaluate the social and economic implications of privacy in the context of safety, law, and ethics	✅
ICS-IC-10 Explain how laws and regulations impact the development and use of software.	✅
HS-CS-01 Describe the use of artificial intelligence within computing systems.	✅
HS-CS-02 Explain how computing devices manage and allocate shared resources.	✅
HS-CS-03 Illustrate the ways computing systems implement logic, input, and output	✅

through hardware components.	
HS-CS-04 Utilize guidelines that convey systematic troubleshooting strategies that debug computer systems.	✅
HS-NI-01 1 Identify issues of network functionality in computational artifact design.	✅
HS-NI-02 Analyze issues of network functionality in computational artifact design.	✅
HS-NI-03 Identify issues of unauthorized access and cybersecurity in computational artifact design.	✅
HS-NI-04 Analyze issues of unauthorized access and cybersecurity in computational artifact design.	✅
HS-NI-05 Explain tradeoffs when selecting and implementing cybersecurity recommendations for various scenarios based on factors such as efficiency, feasibility, and ethical impacts.	✅
HS-DA-01 Identify patterns in data representing complex systems with select data analysis tools and techniques.	✅
HS-DA-02 Select appropriate data collection tools and techniques.	✅
HS-DA-03 Compile data sets that support a claim or communicate information.	✅
HS-DA-04 Identify the ability of models and simulations to test hypotheses.	✅

HS-DA-05 Formulate hypotheses with select models and simulations.	✅
HS-AP-01 Identify artificial intelligence algorithms.	✅
HS-AP-02 Solve computational problems with classic algorithms.	✅
HS-AP-03 Evaluate algorithms in terms of their efficiency, correctness, and clarity.	✅
HS-AP-04 Select an appropriate data structure for information of a given problem.	✅
HS-AP-05 Illustrate the flow of execution of a recursive algorithm.	✅
HS-AP-06 Identify a large-scale computational problem.	✅
HS-AP-07 Analyze general patterns applicable to a solution.	✅
HS-AP-08 Create computational artifacts with pre-existing procedures, external components, libraries and APIs.	✅
HS-AP-09 Create a computational artifact through an industry-standard process.	✅
HS-AP-10 Justify that a computational artifact meets design specifications with systematic testing and debugging methods.	✅
HS-AP-11 Construct a computational artifact as a team through industry	✅

appropriate collaborative tools and processes.	
HS-AP-12 Compose standard documentation for computational artifacts to make it easier to follow, test and debug.	✅
HS-AP-13 Modify an existing computational artifact for additional functionality.	✅
HS-AP-14 Discuss the intended and unintended implications of a modified computational artifact.	✅
HS-AP-15 Develop computational artifacts for multiple platforms	✅
HS-IC-01 Evaluate computational artifacts for their effects on society.	✅
HS-IC-02 Make computational artifact recommendations for maximized beneficial and minimal harmful effects on society.	✅
HS-IC-03 Predict how computational innovations that revolutionized aspects of our culture might evolve.	✅
HS-IC-04 Evaluate how equity, access, and influence impact the distribution of computing resources in a global society.	✅
HS-IC-05 Create computational artifacts to ensure accessibility and reduce computational bias.	✅
HS-IC-06 Utilize tools and methods for collaboration on a project to increase	✅

connectivity of people in different cultures and career fields.	