

Skill Struck's alignment to

South Carolina's Computer Science Standards for High School

Legend

✓ = Standard aligned

♦ = Not currently aligned

🚧 = Partially aligned (parts of the standard we do align with are highlighted yellow)

Standard	Status
Computing Systems	
Standard 1: Use software applications to create an authentic product.	
HS1.CS.1.1 Analyze the impact that computing devices have in real-world settings (e.g., traffic lights, medical devices, facial recognition).	✓
HS1.CS.1.2 Compare and contrast the elements of a computing system by examining hardware elements for their intended use (e.g., input-output (I/O) devices, random access memory (RAM), read only memory (ROM), storage devices, motherboards, and processors including the arithmetic logic unit (ALU), control unit, registers, cache memory, example implementations of some of these components using logic gates) (Virginia, 2017).	✓
HS1.CS.1.3 Compare and contrast the strengths and weaknesses of various popular operating systems, (e.g., Windows, Windows Server, OS-X, Android, iOS, UNIX, Linux).	✓

<p>HS2.CS.1.1 Investigate how a problem is systematically solved through the selection and integration of hardware and software components.</p>	
<p>HS2.CS.1.2 Analyze how various hardware and software layers provide simplifying abstractions (e.g., a redundant array of independent disks (RAID) controller hiding details of data storage on multiple disks, an operating system hiding details of virtual memory, the presentation layer of a network hiding details about encryption).</p>	
<p>HS2.CS.1.3 Analyze the roles of operating system software components (e.g., memory management, data storage and retrieval, process management, access control) in a specific solution (CSTA, 2017).</p>	
<p>HS3.CS.1.1 Recommend modifications for existing computing devices and software to improve functionality for end users.</p>	
<p>HS3.CS.1.2 Justify hardware and software selections for specific applications by evaluating the components (e.g., databases, sensors, application programming interfaces (APIs)) of various computing devices (e.g., desktops, laptops, tablets, smartphones, specialized devices like global positioning systems (GPSs)).</p>	
<p>HS3.CS.1.3 Investigate how the components of an operating system provide simplifying abstractions for a user or developer (e.g., the basic input/output system (BIOS), user interface, memory management, process management, file systems, devices, drivers).</p>	
<p>HS4.CS.1.1 Develop a solution to a given problem using appropriate hardware and</p>	

software (e.g., sensor devices, Wi-Fi capabilities, specialized displays, runtime modules, operating systems, application programming interfaces (APIs)).	
HS4.CS.1.2 Cite evidence of how selecting appropriate hardware and software components enhances user interfaces to provide better solutions for real-world problems	✓
HS4.CS.1.3 Defend the choice of an appropriate operating system based on the requirements of a given computer system or project.	◆
Standard 2: Troubleshoot common hardware and software problems.	
HS1.CS.2.1 Interpret various types of error messages from various sources (e.g., operating systems, applications, application programming interfaces (APIs)) to assist in solving common computer malfunctions.	✓
HS2.CS.2.1 Research credible sources of information that can be used for complex troubleshooting strategies (e.g., modifying system settings, correcting connectivity problems).	✓
HS3.CS.2.1 Solve common computer malfunctions or describe the problem accurately, using technical vocabulary, so that others can solve it.	✓
HS4.CS.2.1 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors (CSTA, 2017).	✓
Networks and the Internet	
Standard 1: Evaluate data transmission across networks, including the Internet.	

HS1.NI.1.1 Describe how hardware, software, and protocols work together for transmitting data across networks.	
HS1.NI.1.2 Model the relationship between routers, switches, servers, topology, and addressing (CSTA, 2017).	
HS2.NI.1.1 Compare and contrast common network protocols (e.g., Internet Protocol (IP), File Transfer Protocol (FTP), Transmission Control Protocol (TCP), Domain Name System (DNS), Hypertext Transfer Protocol (HTTP), Hypertext Transfer Protocol with Secure Sockets Layer (HTTPS), Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP), Internet Message Access Protocol (IMAP), Telnet, Secure Shell Protocol (SSH)) (Arkansas, 2016).	
HS2.NI.1.2 Analyze the tradeoffs and hierarchical structures of various common network topologies (Arkansas, 2016).	
HS3.NI.1.1 Explain each layer of the Open Systems Interconnect (OSI) Model (Arkansas, 2016).	
HS3.NI.1.2 Design a practical, efficient, and secure network solution (e.g., small office network) (Arkansas, 2016).	
HS4.NI.1.1 Trace data through the Open Systems Interconnect (OSI) Model.	
HS4.NI.1.2 Evaluate the scalability and reliability of a network (e.g., bandwidth, load, delay, topology) (CSTA, 2017).	
Standard 2: Evaluate cybersecurity threats and appropriate security	

measures across networks.	
HS1.NI.2.1 Reflect on case studies or current events in which governments or organizations experienced data leaks or data loss as a result of cybersecurity attacks (CSTA, 2017).	◆
HS1.NI.2.2 Identify best practices of software development that improve computer security and protect devices and information from unauthorized access (e.g., encryption, authentication strategies, secure coding, safeguarding keys) (CSTA, 2017).	✓
HS2.NI.2.1 Evaluate how sensitive data can be affected by malware and other attacks (e.g., denial-of-service attacks, ransomware, viruses, worms, spyware, phishing) (CSTA, 2017).	✓
HS2.NI.2.2 Compare and contrast ways software developers protect devices and information from unauthorized access (e.g., encryption, authentication strategies, secure coding, safeguarding keys) (CSTA, 2017).	✓
HS3.NI.2.1 Research security measures (i.e., hardware, software, and practices that control access to data and systems) to combat a variety of cybersecurity vulnerabilities (CSTA, 2017).	✓
HS3.NI.2.2 Evaluate various security measures, considering tradeoffs between the usability and security of a computing system. (e.g., a web filter that prevents access to many educational sites but keeps a campus' network safe) (CSTA, 2017).	✓
HS4.NI.2.1 Recommend security measures (i.e., hardware, software, and practices	✓

that control access to data and systems) to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts (CSTA, 2017).	
HS4.NI.2.2 Select and justify cybersecurity recommendations (i.e., hardware, software, and practices that control access to data and systems) appropriate for an intended audience and purpose (CSTA, 2017).	✓
Data and Analysis	
Standard 1: Evaluate various data collection methods, data storage tools, data analysis tools, data representation tools, and bit representation.	
HS1.DA.1.1 Describe the various data collection methods, data analysis tools, and data representation tools.	✓
HS1.DA.1.2 Describe the various data storage tools and data organization methods.	✓
HS1.DA.1.3 Distinguish between various methods of data representation (i.e., analog, digital, binary).	✓
HS2.DA.1.1 Compare and contrast the various data collection methods, data analysis tools, and data representation tools.	✓
HS2.DA.1.2 Compare and contrast the various data storage tools and data organization methods.	✓
HS2.DA.1.3 Translate between various methods of data representation (i.e., analog, digital, ASCII, binary).	✓

<p>HS3.DA.1.1 Explain how different collection methods and tools influence the amount and quality of the data that is observed and recorded.</p>	
<p>HS3.DA.1.2 Justify choices on how data elements are organized and where data is stored considering cost, speed, reliability, accessibility, privacy, and integrity (e.g., local storage, portable storage, cloud storage).</p>	
<p>HS3.DA.1.3 Continuation of this indicator is not specifically included or excluded.</p>	
<p>HS4.DA.1.1 Justify the choice of a data collection method, data analysis tool, and data representation tool over alternate options.</p>	
<p>HS4.DA.1.2 Evaluate the data storage needs of a computing solution (e.g., file compression).</p>	
<p>HS4.DA.1.3 Continuation of this indicator is not specifically included or excluded.</p>	
<p>Standard 2: Construct a computational model using large data sets of real-world phenomenon.</p>	
<p>HS1.DA.2.1 Describe the properties of a data set that could be used to explore a realworld phenomenon or support a claim.</p>	
<p>HS1.DA.2.2 Identify various types of computational models and their uses for data composed of multiple data elements that relate to one another (e.g., population data may contain information about age, gender, height) (K – 12 Framework, 2016).</p>	

HS2.DA.2.1 Compare and contrast data sets that could be used to explore a real-world phenomenon or support a claim.	<input checked="" type="checkbox"/>
HS2.DA.2.2 Compare and contrast various types of computational models and their uses for data composed of multiple data elements that relate to one another (e.g., population data may contain information about age, gender, height) (K – 12 Framework, 2016).	<input checked="" type="checkbox"/>
HS3.DA.2.1 Create data sets that could be used to explore a realworld phenomenon or support a claim.	<input checked="" type="checkbox"/>
HS3.DA.2.2 Evaluate the limitations of a computational model and the accuracy of inferences.	<input checked="" type="checkbox"/>
HS4.DA.2.1 Evaluate the use of large data sets to explore a realworld phenomenon or support a claim.	<input checked="" type="checkbox"/>
HS4.DA.2.2 Create a computational model using large data sets, make inferences, and address the limitations of the model.	<input checked="" type="checkbox"/>
Standard 3: Create various ways to visually represent data	
HS1.DA.3.1 Identify a data set that could be used to solve a real-world problem.	<input checked="" type="checkbox"/>
Impacts of Computing	
HS1.DA.3.2 Organize collected data to communicate the solution to a real-world phenomenon and support a claim.	<input checked="" type="checkbox"/>

HS2.DA.3.1 Evaluate how the same data set can be visualized and reconstructed to support multiple sides of an issue.	♦
HS2.DA.3.2 Compare and contrast data visualizations for exploring real-world phenomenon or supporting a claim.	✓
HS3.DA.3.1 Construct a data visualization to solve a real-world problem using software tools or programming (e.g., generated scatter, bar, and line charts).	✓
HS3.DA.3.2 Evaluate possible computational models for data visualizations that aid in solving a variety of problems.	✓
HS4.DA.3.1 Analyze patterns in a data visualization then select a collection tool to validate a claim or share information with a group of people.	✓
HS4.DA.3.2 Create a computational model for data visualization.	✓
Algorithms and Programming	
Standard 1: Design algorithms that can be adapted to express an idea or solve a problem.	
HS1.AP.1.1 Create flowcharts and/or pseudocode to express a problem or idea as an algorithm.	✓
HS2.AP.1.1 Create algorithms to solve computational problems that have an application in the real world (e.g., local community, church, civic	✓

organization, school, home life).	
HS3.AP.1.1 Adapt predefined algorithms to solve computational problems.	✓
HS4.AP.1.1 Evaluate algorithms in terms of efficiency, correctness, and clarity (CSTA, 2017).	✓
Standard 2: Build a combination of control structures that supports complex execution, readability, and program performance.	
HS1.AP.2.1 Trace the flow of execution of a program that uses a combination of control structures (e.g., conditionals, loops, event handlers, recursion).	✓
HS1.AP.2.2 Trace the flow of execution of a program that uses a variety of programming constructs (e.g., procedures, modules, objects).	✓
HS2.AP.2.1 Design and iteratively develop programs that combine control structures (e.g., conditionals, loops, event handlers, recursion).	✓
HS2.AP.2.2 Design a solution through systematic analysis using programming constructs (e.g., procedures, modules, objects).	✓
HS3.AP.2.1 Justify the selection of specific control structures explaining the benefits and drawbacks of choices made (e.g., tradeoffs involving implementation, readability, and program performance).	✓
HS3.AP.2.2 Justify the selection of specific programming constructs, explaining the benefits and drawbacks of choices made on the program's execution.	✓

HS4.AP.2.1 Continuation of this indicator is not specifically included or excluded.	
HS4.AP.2.2 Continuation of this indicator is not specifically included or excluded.	
Standard 3: Divide a task into sets of functional units that can be reused to compose a complex solution.	
HS1.AP.3.1 Decompose tasks into smaller, reusable parts to facilitate the design, implementation, and review of programs.	<input checked="" type="checkbox"/>
HS2.AP.3.1 Develop code to solve the smaller parts of a decomposed task that can be reused to solve similar problems (e.g., procedures, functions, objects).	<input checked="" type="checkbox"/>
HS3.AP.3.1 Build a complex solution to a problem that incorporates reusable code (e.g., student-created, application programming interfaces (APIs), libraries).	<input checked="" type="checkbox"/>
HS4.AP.3.1 Justify the selection of modular parts in the creation of a complex solution.	<input checked="" type="checkbox"/>
Standard 4: Plan, build, test, refine, and document programs using text-based coding languages to solve problems with varying degrees of difficulty.	
HS1.AP.4.1 Plan and develop programs for a variety of audiences using a process that incorporates development, feedback, and revision.	<input checked="" type="checkbox"/>
HS1.AP.4.2 Seek and incorporate feedback to refine a solution (e.g., users, team members, code review, teachers).	<input checked="" type="checkbox"/>

<p>HS1.AP.4.3 Recognize the variety of documentation methods available while developing a program (e.g., inline comments, procedure header, purposeful naming).</p>	♦
<p>HS1.AP.4.4 Examine licenses (i.e., permissions) that limit or restrict use of resources (e.g., freeware, shareware, open source, creative commons).</p>	✓
<p>HS2.AP.4.1 Plan and develop a program that addresses potential security issues.</p>	✓
<p>HS2.AP.4.2 Systematically test programs using a range of test cases to meet design specifications (e.g., specific outcomes, functionality, user interface, error handling) (CSTA, 2017).</p>	✓
<p>HS2.AP.4.3 Document programs in order to make them easier to follow, test, and debug.</p>	✓
<p>HS2.AP.4.4 Discuss the implications of using licensed resources in a developed solution.</p>	✓
<p>HS3.AP.4.1 Plan and develop a program that is accessible across multiple computing platforms (e.g., iOS, Unix, Windows, web-based).</p>	✓
<p>HS3.AP.4.2 Evaluate and refine programs to make them more usable, functional, and accessible.</p>	✓
Impacts of Computing	
<p>HS3.AP.4.3</p>	✓

Document programs that use non-user-created resources (e.g., code, media, libraries) giving attribution to the original creator.	
HS3.AP.4.4 Develop a systematic solution that incorporates licensed resources.	♦
HS4.AP.4.1 Evaluate a program through a review process (e.g., code review, beta testing, pilot group).	✓
HS4.AP.4.2 Implement version control to track program refinements.	♦
HS4.AP.4.3 Justify design decisions by documenting the design process of complex programs (e.g., developer journal, digital portfolio, presentation).	✓
HS4.AP.4.4 Research the process for licensing student-created resources.	✓
Standard 5: Choose data types and data structures based on functionality, storage, and performance tradeoffs.	
HS1.AP.5.1 Justify and use appropriate data types (i.e., primitive and non-primitive) in simple programs.	✓
HS2.AP.5.1 Determine when data structures (e.g., lists, arrays, tuples, stacks, queues, structures) are more appropriate than simple data types and incorporate them in programs.	✓
HS3.AP.5.1 Determine when external data structures (e.g., databases, flat files) are appropriate and incorporate them in programs.	♦

HS4.AP.5.1 Justify how data structures and abstraction are used to manage program complexity.	
Impact of Computing	
Standard 1: Evaluate the impact of computing from a global perspective.	
HS1.IC.1.1 Research computing solutions to problems in different countries, considering the personal, ethical, social, economic, and cultural impact (e.g., the use of drones to deliver blood and medical supplies in countries in Africa, the use of Uber in India to address traffic congestion).	
HS1.IC.1.2 Research traditional and non-traditional computer science careers.	
HS2.IC.1.1 Compare and contrast the efficiency, feasibility, and ethical impacts of deploying the same computing solution in various countries.	
HS2.IC.1.2 Identify a computer science career in a nontraditional computer science industry for each of the five computing disciplines (i.e., computer science, software engineering, information technology, information systems, computer engineering).	
HS3.IC.1.1 Hypothesize the future impacts of computing across the world (e.g., personal, health, workforce, economy, education, culture, environment).	
HS3.IC.1.2	

Research how computing is used in non-traditional computer science careers (e.g., sensors on soldiers' or firefighters' uniforms, robots to detect and diffuse explosive devices).	
HS4.IC.1.1 <i>Continuation of this indicator is not specifically included or excluded.</i>	
HS4.IC.1.2 Continuation of this indicator is not specifically included or excluded.	
Standard 2: Evaluate the evolving legal and ethical tradeoffs that shape computing practices.	
HS1.IC.2.1 Select the most appropriate means of communication for given situations (e.g., personal versus professional communication, communication with teachers and employers).	✓
HS1.IC.2.2 Discuss issues related to personal security (e.g., personal, financial, professional).	✓
HS1.IC.2.3 Explain the implications of proper and improper use of social media (e.g., college admissions, employment, cyberbullying laws).	✓
HS2.IC.2.1 Discuss how social media and computing devices have positively and negatively impacted communication.	✓
HS2.IC.2.2 Define and visually display students' digital footprint.	✓

HS2.IC.2.3 Identify ethical and legal computing practices.	
HS3.IC.2.1 Justify proper and improper use of social media and computing devices (e.g., role-playing and example scenarios).	
HS3.IC.2.2 Analyze the relationship between students' digital footprint and personal security	
HS3.IC.2.3 Distinguish among ethical, unethical, legal, and illegal computing practices (e.g., fair-use, illegal music/video downloads, sharing copyrighted pictures/videos, black-hat hacking, white-hat hacking).	
HS4.IC.2.1 Create rules of etiquette for proper use of social media and computing devices.	
HS4.IC.2.2 Recommend methods to protect digital information in different situations (e.g., traveling to other countries, two-factor authentication, encryption).	
HS4.IC.2.3 Investigate how computer use and digital privacy are governed across the globe (e.g., government regulations for computer use in the United State, Canada, China, North Korea, and Russia).	
Standard 3: Understand the importance of access and equity in computing.	
HS1.IC.3.1	

Identify factors (e.g., net neutrality, government regulations, infrastructure, funding) that impact equitable access to computing resources for underrepresented groups (e.g., race, ethnicity, gender, geographic location, socioeconomic status).	
HS1.IC.3.2 Identify computer scientists from underrepresented groups and their specific contributions (e.g., African-American, Latino, women, disabled).	♦
HS1.IC.3.3 Identify the advantages and disadvantages of diverse perspectives and backgrounds when solving computational problems.	✓
HS2.IC.3.1 Research current efforts to provide equitable access to computing resources for underrepresented groups (e.g., race, ethnicity, gender, geographic location, socioeconomic status).	✓
HS2.IC.3.2 <i>Continuation of this indicator is not specifically included or excluded.</i>	
HS2.IC.3.3 Evaluate existing computing solutions according to inclusivity or non-inclusivity (e.g., sight and hearing impairment, ethnicity, age).	✓
HS3.IC.3.1 Evaluate the effectiveness of current efforts to provide equitable access to computing resources for underrepresented groups (e.g., race, ethnicity, gender, geographic location, socioeconomic status).	✓
HS3.IC.3.2	

<i>Continuation of this indicator is not specifically included or excluded.</i>	
HS3.IC.3.3 Recommend modifications to make a current computing solution more inclusive for all users.	✓
HS4.IC.3.1 Design a solution to improve equitable access to computing resources for underrepresented groups (e.g. classroom, school, neighborhood).	✓
HS4.IC.3.2 <i>Continuation of this indicator is not specifically included or excluded.</i>	
HS4.IC.3.3 <i>Continuation of this indicator is not specifically included or excluded.</i>	✓
Standard 4: Evaluate the history of computers and computing.	
HS1.IC.4.1 Discuss how Moore’s Law has impacted computing.	✓
HS1.IC.4.2 Define and provide examples of big data (e.g., information collected from social media or smartphone use).	✓
HS2.IC.4.1 Discuss the advantages and disadvantages of advancing and emerging technologies over time (e.g., the impacts of artificial intelligence, virtual reality, and biometrics on productivity, job loss, inventions, quality of life, and globalization).	✓

HS2.IC.4.2 Research how big data is used to solve computing problems.	<input checked="" type="checkbox"/>
HS3.IC.4.1 Hypothesize problems that the next generation of computing will solve.	<input checked="" type="checkbox"/>
HS3.IC.4.2 <i>Continuation of this indicator is not specifically included or excluded.</i>	
HS4.IC.4.1 <i>Continuation of this indicator is not specifically included or excluded.</i>	
HS4.IC.4.2 <i>Continuation of this indicator is not specifically included or excluded.</i>	