Public Schools of North Carolina
State Board of Education | Department of Public Instruction

2020 North Carolina K12 Computer Science Standards with Descriptions.

This document is designed to help North Carolina educators teach the NC Standard Course of Study for Computer Science.

This document provides more detailed descriptions of each standard in the 2020 NC K12 Computer Science Standards which are based on the 2017 Computer Science Teachers Association Computer Science Standards.

High School Level 1 CS

HS-CS-01	**Describe the use of artificial intelligence within computing systems.**	**Computing Systems** Devices
	AI is present in nearly all computing systems. Students should be able to identify and describe how AI is playing a role in various industries and applications. Examples include digital ad delivery, self-driving cars, and credit card fraud detection.	
HS-CS-02	**Explain how computing devices manage and allocate shared resources.**	**Computing Systems** Hardware & Software
	Computing systems are often very complicated interconnected systems with numerous shared resources. Students need to understand how computing systems manage these resources and avoid conflicts.	
HS-CS-03	**Illustrate the ways computing systems implement logic, input, and output through hardware components.**	**Computing Systems** Troubleshooting
	While much of what students perceive as computer science revolves around the input and output of computing systems which on the surface appears to be purely software, students should understand how every piece of software relies on the physical hardware that translates the code into a computing solution. Students should explore the hardware implementations of such components as logic gates and IO pins.	
HS-CS-04	**Utilize guidelines that convey systematic troubleshooting strategies that debug computer systems.**	**Computing Systems** Troubleshooting
	Troubleshooting complex problems involves the use of multiple sources when researching, evaluating, and implementing potential solutions. Troubleshooting also relies on experience, such as when people recognize that a problem is similar to one they have seen before or adapt solutions that have worked in the past. Examples of complex troubleshooting strategies include resolving connectivity problems, adjusting system configurations and settings, ensuring hardware and software compatibility, and transferring data from one device to another. Students could create a flow chart, a job aid for a help desk employee, or an expert system.	
HS-NI-01	**Identify issues of network functionality in computational artifact design.**	**Networks & the Internet** Network Communication & Organization
	As more computational artifacts rely on online resources or components for functionality, students should be able to identify issues that may arise from connected applications (bandwidth, latency, security). There are online simulators that can help students explore these issues.	

HS-NI-02	**Analyze issues of network functionality in computational artifact design.**	**Networks & the Internet** Network Communication & Organization
	As students work with network dependent artifacts, they should be able to test and analyze the impacts of networking issues on artifacts. Students can test artifacts on various devices and utilize different networks.	
HS-NI-03	**Identify issues of unauthorized access and cybersecurity in computational artifact design.**	**Networks & the Internet** Cybersecurity
	Among the most challenging aspects of network connected artifacts is the inherent threat of unauthorized access and cybersecurity issues. Students should be able to identify the risks posed by these threats.	
HS-NI-04	**Analyze issues of unauthorized access and cybersecurity in computational artifact design.**	**Networks & the Internet** Cybersecurity
	Students should be able to recognize particular security threats and compare ways developers might protect against them. In particular, students should explore examples of mitigation strategies such as encryption and authentication strategies, secure coding, and safeguarding keys.	
HS-NI-05	**Explain tradeoffs when selecting and implementing cybersecurity recommendations for various scenarios based on factors such as efficiency,**	**Networks & the Internet** Cybersecurity
	Network security depends on a combination of hardware, software, and practices that control access to data and systems. The needs of users and the sensitivity of data determine the level of security implemented. Every security measure involves tradeoffs between the accessibility and security of the system. Students should be able to describe, justify, and document choices they make using terminology appropriate for the intended audience and purpose. Students could debate issues from the perspective of diverse audiences, including individuals, corporations, privacy advocates, security experts, and government.	
HS-DA-01	**Identify patterns in data representing complex systems with select data analysis tools and techniques.**	**Data & Analysis** Collection Visualization & Transformation
	Students should be able to identify patterns in data representing complex systems, such as identify trends in a dataset representing social media interactions, movie reviews, or shopping patterns.	

HS-DA-02	**Select appropriate data collection tools and techniques.**	**Data & Analysis** Collection Visualization & Transformation
	The use of data in computing artifacts is one of the dominant forces shaping technology today. Students should be familiar with data collection tools (surveys, scientific probes, online repositories) and techniques (harvesting, crowd-sourcing, simulations). Students should be able to select the appropriate tools and techniques for a given task.	
HS-DA-03	**Compile data sets that support a claim or communicate information.**	**Data & Analysis** Collection Visualization & Transformation
	Communicating with data is a critical component for many technology innovations. Whether it is rank ordering possible purchases on an ecommerce site or displaying the progress in a video game, data is often used to communicate information to the end user. Students should be able to articulate a story or impact through data and visualizations.	
HS-DA-04	**Identify the ability of models and simulations to test hypotheses.**	**Data & Analysis** Inference & Models
	Especially in scientific research, computers are playing an increasing role in testing hypotheses. Because computers have the ability to run millions of simulations in the time a lab experiment might conduct a handful, modeling and simulations are critical to modern science. Students should explore the application of modeling to solve society's challenges, recent examples include medicine, industrial design, and environmental science.	
HS-DA-05	**Formulate hypotheses with select models and simulations.**	**Data & Analysis** Inference & Models
	Students should experience models and simulations in hands-on experiences. They should use existing models and simulations to formulate hypotheses. These can be online interactives (Phet) or could be specific applications written for the class to test.	
HS-AP-01	**Identify artificial intelligence algorithms.**	**Algorithms & Programming** Algorithms
	As algorithms are a foundational component of computer science, students should recognize the role of algorithms in machine learning. Among the most common algorithms are regression, decision trees, and search. By exploring these algorithms, students will begin to develop a foundation for implementing AI.	

HS-AP-02	**Solve computational problems with classic algorithms.**	**Algorithms & Programming** Algorithms
	Algorithms are a foundational component of computer science. There are numerous foundational algorithms that serve as the basis for many applications. Students should be familiar with and should implement many of the classic algorithms, especially those for sorting (bubble, selection) and search (binary, linear) .	
HS-AP-03	**Evaluate algorithms in terms of their efficiency, correctness, and clarity.**	**Algorithms & Programming** Algorithms
	When selecting appropriate algorithms, students must make decisions based on the resources available and the constraints of the problem. Students should evaluate various algorithms for efficiency, correctness, and clarity when implementing. Search and sort algorithms are ideal for such explorations.	
HS-AP-04	**Select an appropriate data structure for information of a given problem.**	**Algorithms & Programming** Variables
	Managing data within a computational artifact is critical. Students should be able to implement data storage using various methods. In addition to understanding primitive data types (int, float, char, string), students should be able to implement data structures such as lists, arrays, stacks, and queues.	
HS-AP-05	**Illustrate the flow of execution of a recursive algorithm.**	**Algorithms & Programming** Control
	While recursive algorithms are relatively simplistic in design, they can be quite challenging to implement and debug. Students should be able to trace the implementation of recursive algorithms. Recursive sorting and searching are good algorithms to explore.	
HS-AP-06	**Identify a large-scale computational problem.**	**Algorithms & Programming** Modularity
	Students should be able to identify real-world problems that span several domains which represent opportunities for large-scale computational solutions. Students might explore the Grand Challenges organization for ideas.	
HS-AP-07	**Analyze general patterns applicable to a solution.**	**Algorithms & Programming** Modularity

	As students encounter complex, real-world problems that span multiple disciplines or social systems, they should decompose complex problems into manageable subproblems that could potentially be solved with programs or procedures that already exist. For example, students could create an app to solve a community problem by connecting to an online database through an application programming interface (API).	
HS-AP-08	**Create computational artifacts with pre-existing procedures, external components, libraries and APIs.**	**Algorithms & Programming** Modularity
	Computational artifacts can be created by combining and modifying existing artifacts or by developing new artifacts. Examples of computational artifacts include programs, simulations, visualizations, digital animations, robotic systems, and apps. Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose. Modules allow for better management of complex tasks. The focus at this level is understanding a program as a system with relationships between modules. The choice of implementation, such as programming language or paradigm, may vary. Students could incorporate computer vision libraries to increase the capabilities of a robot or leverage open-source JavaScript libraries to expand the functionality of a web application.	
HS-AP-09	**Create a computational artifact through an industry-standard process.**	**Algorithms & Programming** Program Development
	As much as computer science involves knowledge and skills withinin program development, the real application of computer science also requires an understanding of the key processes used in the workplace. Students should begin to develop comfort with these processes, including agile, spiral, or waterfall.	
HS-AP-10	**Justify that a computational artifact meets design specifications with systematic testing and debugging methods.**	**Algorithms & Programming** Program Development
	Students should develop and use a series of test cases to verify that a program performs according to its design specifications.	
HS-AP-11	**Construct a computational artifact as a team through industry appropriate collaborative tools and processes.**	**Algorithms & Programming** Program Development
	When developing software, development teams must use tools designed to facilitate collaboration and to manage version control. Students should work in teams to experience these challenges and to understand industry solutions, such as GITHUB.	
HS-AP-12	**Compose standard documentation for computational artifacts to make it easier to follow, test, and debug.**	**Algorithms & Programming** Program Development

	As part of best practices, especially when working on development teams, students should implement appropriate documentation strategies. These strategies make it easier for teammates to better follow the logic and debug any issues.	
HS-AP-13	**Modify an existing computational artifact for additional functionality.**	**Algorithms & Programming** Program Development
	While students should be able to develop computational artifacts of their own design, it is also critical that students be able to work on existing artifacts. Students should be comfortable with taking an existing artifact and making enhancements or improvements.	
HS-AP-14	**Discuss intended and unintended implications of a modified computational artifact.**	**Algorithms & Programming** Program Development
	As students modify an existing application, they must be aware that all changes have implications, some intended and some not. Students should be prepared to justify these tradeoffs and to document the implications. For instance, changes made to a method or function signature could break invocations of that method elsewhere in a system.	
HS-AP-15	**Develop computational artifacts for multiple platforms.**	**Algorithms & Programming** Program Development
	As students work to create computational artifacts, they should be aware that end users may experience the artifact on various devices. As such, students should design and develop their solution with various experiences and operating systems in mind. Students might develop multi-platform solutions that work across computer desktop, web, and mobile.	
HS-IC-01	**Evaluate computational artifacts for their effects on society.**	**Impacts of Computing** Culture
	As much as humans are shaping the technology innovations in society, it is also the case that technology is shaping society. Students should be able to recognize how recent technological innovations have impacted our world. Students may consider the Internet broadly, but should also examine specific applications on the web, including online e-commerce, social networking, and collaboration tools.	
HS-IC-02	**Make computational artifact recommendations for maximized beneficial and minimal harmful effects on society.**	**Impacts of Computing** Culture
	Everyone who leverages the power of computing to create innovative products bears the responsibility for minimizing the harmful impacts that their work has on society. Students should use existing innovations to provide meaningful feedback on how those artifacts might maximize their beneficial effects and minimize harmful effects on society.	

HS-IC-03	**Predict how computational innovations that revolutionized aspects of our culture might evolve.**	**Impacts of Computing** Culture
	Technology is not static. Our use of digital tools will only increase with time. Students might consider how these technologies might evolve especially in the areas of education, healthcare, art/entertainment, and energy.	
HS-IC-04	**Evaluate how equity, access, and influence impact distribution of computing resources in a global society.**	**Impacts of Computing** Culture
	The distribution of computing resources is not uniform across all parts of our global society. Students should evaluate how equity and access are impacted by factors such as wealth, geographic location, governmental programs. For example, students might explore the effectiveness of the US E-rate program for increasing access to technology in schools.	
HS-IC-05	**Create computational artifacts to ensure accessibility and reduce computational bias.**	**Impacts of Computing** Culture
	Biases could include incorrect assumptions developers have made about their user base. Equity deficits include minimal exposure to computing, access to education, and training opportunities. Students should begin to identify potential bias during the design process to maximize accessibility in product design and become aware of professionally accepted accessibility standards to evaluate computational artifacts for accessibility.	
HS-IC-06	**Utilize tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields.**	**Impacts of Computing** Social Interactions
	Many aspects of society, especially careers, have been affected by the degree of communication afforded by computing. The increased connectivity between people in different cultures and in different career fields has changed the nature and content of many careers. Students should explore different collaborative tools and methods used to solicit input from team members, classmates, and others, such as participation in online forums or local communities. For example, students could compare ways different social media tools could help a team become more cohesive.	