NC K-12 Computer Science Standards

Concept	Subconcept	Grades K-2 (Ages 5-7) *By the end of Grade 2, students will be able to...*	Grades 3-5 (Ages 8-11) *By the end of Grade 5, students will be able to...*	Grades 6-8 (Ages 11-14) *By the end of Grade 8, students will be able to...*	Introduction to CS *By completing this course, students will be able to...*	High School – CS Level 1 *By completing a course, students will be able to...*
Computing Systems	Devices	**K2-CS-01** Choose appropriate devices to perform a variety of classroom tasks.	**35-CS-01** Evaluate the features available on digital devices to perform a variety of classroom tasks.	**68-CS-01** Understand the design of computing devices based on an analysis of how users interact with the devices.	**ICS-CS-01** Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.	**HS-CS-01** Describe the use of artificial intelligence within computing systems.
	Hardware & Software	**K2-CS-02** Describe the function of common physical components of computing systems (hardware) with appropriate terminology.	**35-CS-02** Model how computer hardware and software work together as a system to accomplish tasks.	**68-CS-02** Design projects that combine hardware and software components to collect and exchange data.	**ICS-CS-02** Compare levels of abstraction and interactions between application software, system software, and hardware layers.	**HS-CS-02** Explain how computing devices manage and allocate shared resources.
		K2-CS-03 Operate appropriate software to perform a variety of tasks.			**ICS-CS-03** Explain the roles of operating systems including memory management, data storage/retrieval, process management, and access control.	**HS-CS-03** Illustrate the ways computing systems implement logic, input, and output through hardware components.
	Troubleshooting	**K2-CS-04** Describe basic hardware and software problems with accurate terminology.	**35-CS-03** Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies.	**68-CS-03** Systematically identify and fix problems with computing devices and components.	**ICS-CS-04** Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.	**HS-CS-04** Utilize guidelines that convey systematic troubleshooting strategies that debug computer systems.
Networks & The Internet	Network Communication & Organization	**K2-NI-01** Illustrate how information is broken down into smaller pieces and can be reassembled.	**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.	**68-NI-01** Anaylze different ways that data is transferred across a network and the role of protocols in transmitting data.	**ICS-NI-01** Evaluate the relationship between routers, switches, servers, and topology with regard to networks.	**HS-NI-01** Identify issues of network functionality in computational artifact design.
						HS-NI-02 Analyze issues of network functionality in computational artifact design.
	Cybersecurity	**K2-NI-02** Apply knowledge of what passwords are and why we use strong passwords to protect devices and information from unauthorized access.		**68-NI-02** Explain how physical and digital security measures protect electronic information.	**ICS-NI-02** Identify examples to illustrate how sensitive data can be affected by malware and other attacks.	**HS-NI-03** Identify issues of unauthorized access and cybersecurity in computational artifact design.
		K2-NI-03 Discover your digital footprint and how personal information can be protected.	**35-NI-02 E**xplain your digital footprint and how personal information can be protected.	**68-NI-03** Explain permission and authorizations to access resources to computer systems online.		**HS-NI-04** Analyze issues of unauthorized access and cybersecurity in computational artifact design.
				68-NI-04 Apply multiple methods of encryption to model the secure transmission of information.	**ICS-NI-03** Recommend cybersecurity measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts.	**HS-NI-05** Explain tradeoffs when selecting and implementing cybersecurity recommendations for 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.	

Data & Analysis	**Storage**	**K2-DA-01** Store, copy, search, retrieve, modify, and delete information using a computing device.	**35-DA-01** Identify the type of data encoded in a file based on file extension.	**68-DA-01** Represent data using multiple encoding schemes.	**ICS-DA-01** Compare different binary representations of data, including text, sound, images, and numbers.	
		K2-DA-02 Define information stored on a computing device as data.	**35-DA-02** Illustrate the process of file management and version control.		**ICS-DA-02** Evaluate the tradeoffs in how data elements are organized and where data is stored.	
	Collection, Visualization, & Transformation					**HS-DA-01** Identify patterns in data representing complex systems with select data analysis tools and techniques.
		K2-DA-03 Collect and present the same data in various visual formats.	**35-DA-03** Organize and present collected data visually to highlight relationships and support a claim.	**68-DA-02** Collect data using computational tools.	**ICS-DA-03** Create interactive data visualizations using software tools to help others better understand real-world phenomena.	**HS-DA-02** Select appropriate data collection tools and techniques.
				68-DA-03 Transform the collected data to make it more useful and reliable.		**HS-DA-03** Compile data sets that support a claim or communicate information.
	Inference & Models	**K2-DA-04** Make predictions with patterns in data visualizations.	**35-DA-04** Communicate using data to highlight or predict outcomes.	**68-DA-04** Refine computational models based on the data they have generated and/or data collected.	**ICS-DA-04** Create computational models that represent the relationships among different elements of data collected.	**HS-DA-04** Identify the abilty of models and simulations to test hypotheses.
						HS-DA-05 Formulate hypotheses with select models and simulations.
	Algorithms	**K2-AP-01** Model daily processes with algorithms to complete tasks.	**35-AP-01** Create multiple algorithms for the same task to determine which is the most accurate and efficient.	**68-AP-01** Implement flowcharts and/or pseudocode to address complex problems as algorithms.	**ICS-AP-01** Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.	**HS-AP-01** Identify artificial intelligence algorithms.
					ICS-AP-02 Explain the use of artificial intelligence within computing systems.	**HS-AP-02** Solve computational problems with classic algorithms.
						HS-AP-03 Evaluate algorithms in terms of their efficiency, correctness, and clarity.
	Variables	**K2-AP-02** Demonstrate how programs store and manipulate data by using numbers or other symbols to represent information.	**35-AP-02** Create programs that use variables to store and modify data.	**68-AP-02** Create clearly named variables that represent different data types.	**ICS-AP-03** Utilize lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.	**HS-AP-04** Select an appropriate data structure for information of a given problem.
	Control	**K2-AP-03** Develop programs with sequences and simple loops to express ideas or address a problem.	**35-AP-03** Construct programs that include sequences.	**68-AP-03** Design and iteratively develop programs that combine control structures including nested loops and compound conditionals.	**ICS-AP-04** Justify the selection of specific control structures, considering implementation, readability, and program performance.	
			35-AP-04 Construct programs using simple loops.	**68-AP-04** Construct programs that include events.	**ICS-AP-05** Iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.	**HS-AP-05** Illustrate the flow of execution of a recursive algorithm.
			35-AP-05 Construct programs that implement conditionals.			

Algorithms & Programming	**Modularity**	**K2-AP-04** Decompose the steps needed to solve a problem into a precise sequence of instructions.	**35-AP-06** Decompose problems into smaller, manageable, subproblems to facilitate the program development process.	**68-AP-05** Organize problems and subproblems into parts.	**ICS-AP-06** Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.	**HS-AP-06** Identify a large-scale computational problem.
						HS-AP-07 Analyze general patterns applicable to a solution.
				68-AP-06 Explain the design, implementation, and review of programs.		
			35-AP-07 Modify, remix, or incorporate portions of an existing program into one's own work.	**68-AP-07** Create procedures with parameters to organize code and make it easier to reuse groups of instructions.	**ICS-AP-07** Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.	**HS-AP-08** Create computational artifacts with pre-existing procedures, external components, libraries and APIs.
	Program Development	**K2-AP-05** Develop plans that describe a program's sequence of events, goals, and expected outcomes.	**35-AP-08** Apply an iterative process to the development of a program by including diverse perspectives and considering user preferences.	**68-AP-08** Assess feedback from team members and users to refine a solution that meets user needs.	**ICS-AP-08 S**ystematically design programs for broad audiences.	**HS-AP-09** Create a computational artifact through an industry-standard process.
					ICS-AP-09 Refine programs by incorporating feedback from users.	
		K2-AP-06 Give attribution when using the ideas and creations of others while developing programs.	**35-AP-09** Give appropriate attribution when creating or remixing programs while respecting intellectual property rights.	**68-AP-09** Incorporate existing code and media into original programs and give attribution.	**ICS-AP-10** Evaluate licenses that limit or restrict use of computational artifacts when using resources such as software libraries.	
		K2-AP-07 Identify and debug errors in an algorithm or program that includes sequences and simple loops.	**35-AP-10** Identify and debug erros in an algorithm or program to ensure it runs as intended.	**68-AP-10** Systematically test and refine programs using a range of test cases.	**ICS-AP-11** Evaluate computational artifacts for usability.	**HS-AP-10** Justify that a computational artifact meets design specifications with systematic testing and debugging methods.
					ICS-AP-12 Modify computational artifacts to increase usability and accessibility.	
			35-AP-11 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.	**68-AP-11** Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.	**ICS-AP-13** Develop computational artifacts working in team roles using collaborative tools.	**HS-AP-11** Construct a computational artifact as a team through industry appropriate collaborative tools and processes.
		K2-AP-08 Using correct terminology, describe steps taken and choices made during the iterative process of program development.	**35-AP-12** Describe choices made during program development using code comments, presentations, and demonstrations.	**68-AP-12** Document programs in order to make them easier to follow, test, and debug.	**ICS-AP-14** Explain design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.	**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 intended and unintended implications of a modified computational artifact.
						HS-AP-15 Develop computational artifacts for multiple platforms.

Impacts of Computing	Culture	**K2-IC-01** Compare how people live and work before and after the implementation or adoption of new computing technology.	**35-IC-01** Compare computing technologies that have changed the world and how they both influence and are influenced by cultural practices.	**68-IC-01** Compare tradeoffs associated with computing technologies that affect everyday activities and career options.	**ICS-IC-01** Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.	**HS-IC-01** Evaluate computational artifacts for their effects on society.
						HS-IC-02 Make computational artifact recommendations for maximized benefical and minimal harmful effects on society.
		K2-IC-02 Select software that meets the diverse needs and preferences for the technology individuals use in the classroom.	**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.		**ICS-IC-02** Elaborate how computational innovations have and may continue to impact society.	**HS-IC-03** Predict how computational innovations that revolutionized aspects of our culture might evolve.
				68-IC-02 Describe how equity, access, and influence impact the distribution of computing resources in a global society.	**ICS-IC-03** Evaluate how equity, access, and influence impact distribution of computing resources in a global society.	**HS-IC-04** Evaluate how equity, access, and influence impact distribution of computing resources in a global society.
				68-IC-03 Discuss issues of bias and accessibility in the design of existing technologies.	**ICS-IC-04** Test computational artifacts to reduce bias and equity deficits.	**HS-IC-05** Create computational artifacts to ensure accessibility and reduce computational bias.
				68-IC-04 Collaborate, model, and promote effective research strategies for assessing and evaluating innovative resources.	**ICS-IC-05** Demonstrate ways a given algorithm applies to problems across disciplines.	
	Social Interactions	**K2-IC-03** Work respectfully and responsibly with others online.	**35-IC-03** Seek diverse perspectives with collaboration for the purpose of improving computational artifacts.	**68-IC-05** Collaborate with many contributors to create a computational artifact.	**ICS-IC-06** Utilize tools and methods for collaboration on a project to increase connectivity of peers.	**HS-IC-06** Utilize tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields.
				68-IC-06 Utilize tools and methods for collaboration on a project to increase connectivity of peers.		
			35-IC-04 Exhibit positive digital citizenship and social responsibility in online interactions.	**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 aspect of others.		
	Safety, Law, & Ethics	**K2-IC-04** Model responsible login and logoff procedures on all devices.	**35-IC-05** Utilize public domain or creative commons media, and refrain from copying or using material created by others without permission.	**68-IC-09** Compare tradeoffs between allowing information to be public and keeping information private and secure.	**ICS-IC-07** Explain the beneficial and harmful effects that intellectual property laws can have on innovation.	
				68-IC-10 Explore how laws and regulations impact the development and use of software.	**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.	