Public Schools of North Carolina

State Board of Education | Department of Public Instruction

Crosswalk for the 2020 North Carolina K12 Computer Science Standards aligned with *SCRATCH: Creative Computing*.

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

This document is a general alignment of the 2020 NC K12 Computer Science Standards which are based on the 2017 Computer Science Teachers Association Computer Science Standards to a common national curriculum.

Kindergarten through Eighth Grade
Mapped to *SCRATCH: Creative Computing*

NC Standard	SCRATCH: Creative Computing						
	0	1	2	3	4	5	6
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.							
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							
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							
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 aspect 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							