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 *Coding with Minecraft*.

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.

Grades Sixth through Eighth
Mapped to *Coding with Minecraft*

	Coding with Minecraft Program		
NC Standard	**Intro**	**Inter**	**Adv**
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			

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			