North Carolina Computer Science, IT, and Technology Education

Grades 6-8

Computing Systems

Devices

1 Understand the design of computing devices based on an analysis of how users interact with the devices. 68-CS-01

Hardware & Software

2 Design projects that combine hardware and software components to collect and exchange data. 68-CS-02

Troubleshooting

3 Systematically identify and fix problems with computing devices and components. 68-CS-03

Networks & The Internet

Network Communication & Organization

1 Analyze different ways that data is transferred across a network and the role of protocols in transmitting data. 68-NI-01

Cybersecurity

2 Explain how physical and digital security measures protect electronic information. 68-NI-02

3 Explain permission and authorizations to access resources to computer systems online. 68-NI-03

4 Apply multiple methods of encryption to model the secure transmission of information. 68-NI-04

Data & Analysis

Storage

1 Represent data using multiple encoding schemes. 68-DA-01

Collection, Visualization, & Transformation

2 Collect data using computational tools. 68-DA-02

3 Transform the collected data to make it more useful and reliable. 68-DA-03

Inference & Models

4 Refine computational models based on the data they have generated and/or data collected. 68-DA-04

Algorithms & Programming

Algorithms

1 Implement flowcharts and/or pseudocode to address complex problems as algorithms. 68-AP-01

Variables

2 Create clearly named variables that represent different data types. 68-AP-02

Control

3 Design and iteratively develop programs that combine control structures including nested loops and compound conditionals. 68-AP-03

4 Construct programs that include events. 68-AP-04

Modularity

5 Organize problems and subproblems into parts. 68-AP-05

6 Explain the design, implementation, and review of programs. 68-AP-06

7 Create procedures with parameters to organize code and make it easier to reuse groups of instructions. 68-AP-07

Program Development

8 Assess feedback from team members and users to refine a solution that meets user needs. 68-AP-08

9 Incorporate existing code and media into original programs and give attribution. 68-AP-09

10 Systematically test and refine programs using a range of test cases. 68-AP-10

11 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts. 68-AP-11

12 Document programs in order to make them easier to follow, test, and debug. 68-AP-12

Impacts of Computing

Culture

1 Compare tradeoffs associated with computing technologies that affect everyday activities and career options. 68-IC-01

2 Describe how equity, access, and influence impact the distribution of computing resources in a global society. 68-IC-02

3 Discuss issues of bias and accessibility in the design of existing technologies 68-IC-03

4 Collaborate, model, and promote effective research strategies for assessing and evaluating innovative resources. 68-IC-04

Social Interactions

5 Collaborate with many contributors to create a computational artifact. 68-IC-05

68-IC-06. Utilize tools and methods for collaboration on a project to increase connectivity of peers. 68-IC-06

7 Examine the benefits and drawbacks of a digital footprint and online identity. 68-IC-07

8 Understand how online interactions make an impact on the social, emotional, and physical aspect of others. 68-IC-08

Safety, Law, & Ethics

9 Compare tradeoffs between allowing information to be public and keeping information private and secure. 68-IC-09

10 Explore how laws and regulations impact the development and use of software. 68-IC-10