North Carolina Standards Alignment with CodeX Curriculum

	Unit 1	Unit 2	Unit 3
Computing Systems			
68-CS-01 Understand the design of computing devices based on ananalys is of how users interact with the devices.			
68-CS-02 Design projects that combine hardware and software components to collect and exchange data.	[1]		
68-CS-03 Systematically identify and fix problems with computing devices and components.	[2]		
Networks & the Internet			
68-NI-01 Anaylze different ways thatdata is transferred across a networkand the role of protocols intransmitting data.			
68-NI-02 Explain how physical anddigital security measures protectelectronic information.			
68-NI-03 Explain permission andauthorizations to access resources tocomputer systems online.			
68-NI-04 Apply multiple methods ofencryption to model the securetransmission of information.			
Data and Analysis			
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 reliable.			
68-DA-04 Refine computational models based on the data they have generated and/or data collected.			
Algorithms and Programming			
68-AP-01 Implement flowcharts and/or pseudocode to address complex problems as algorithms.	[3]		
68-AP-02 Create clearly named variables that represent different data types.	[4]		
68-AP-03 Design and iteratively develop programs that combine control structures including nested loops and compound conditionals.	[5]		
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.	[6]		
68-AP-07 Create procedures with parameters to organize code and make it easier to reuse groups of instructions.	[7]		
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.	[8]		
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.	[9]		
Impacts of Computing			
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.

North Carolina Standards Alignment with CodeX Curriculum

	Unit 1	Unit 2	Unit 3
Computing Systems			
ICS-CS-01 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.			
ICS-CS-02 Compare levels of abstraction and interactions between application software, system software, and hardware layers.			
ICS-CS-03 Explain the roles of operating systems including memory management, data storage/retrieval, process management, and access control.			
ICS-CS-04 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.	[10]		
Networks & the Internet			
ICS-NI-01 Evaluate the relationship between routers, switches, servers, and topology with regard to networks.			
ICS-NI-02 Identify examples to illustrate how sensitive data can be affected by malware and other attacks.			
ICS-NI-03 Recommend cybersecurity measures to address 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 and Analysis			
ICS-DA-01 Compare different binary representations of data, including text, sound, images, and numbers.			
ICS-DA-02 Evaluate the tradeoffs in how data elements are organized and where data is stored.			
ICS-DA-03 Create interactive data visualizations using software tools to help others better understand real-world phenomena.			
ICS-DA-04 Create computational models that represent the relationships among different elements of data collected.			
Algorithms and Programming			
ICS-AP-01 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.	[11]		
ICS-AP-02 Explain the use of artificial intelligence within computing systems.			
ICS-AP-03 Utilize lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.		[12]	
ICS-AP-04 Justify the selection of specific control structures, considering implementation, readability, and program performance.			
ICS-AP-05 Iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.		[13]	
ICS-AP-06 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.		[14]	
ICS-AP-07 Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.			
ICS-AP-08 Systematically design programs for broad audiences.			
ICS-AP-09 Refine programs by incorporating feedback from users			
ICS-AP-10 Evaluate licenses that limit or restrict use of computational artifacts when using resources such as software libraries.			
ICS-AP-11 Evaluate computational artifacts for usability.			
ICS-AP-12 Modify computational artifacts to increase usability and accessibility.			
ICS-AP-13 Develop computational artifacts working in team roles using collaborative tools.			
ICS-AP-14 Explain design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.	[15]		
Impacts of Computing			
ICS-IC-01 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.			
ICS-IC-02 Elaborate how computational innovations have and may continue to impact society.			
ICS-IC-03 Evaluate how equity, access, and influence impact distribution of computing resources in a global society.			
ICS-IC-04 Test computational artifacts to reduce bias and equity deficits.			
ICS-IC-05 Demonstrate ways a given algorithm applies to problems across disciplines.			
ICS-IC-06 Utilize tools and methods for collaboration on a project to increase connectivity of peers.			
ICS-IC-07 Explain the beneficial and harmful effects that intellectual property laws can have on innovation.			
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.			

North Carolina Standards Alignment with CodeX Curriculum

	Unit 1	Unit 2	Unit 3
Computing Systems			
HS-CS-01 Describe the use of artificial intelligence within computing systems.			
HS-CS-02 Explain how computing devices manage and allocate shared resources.			
HS-CS-03 Illustrate the ways computing systems implement logic, input, and output through hardware components			
HS-CS-04 Utilize guidelines that convey systematic troubleshooting strategies that debug computer systems.	[16]		
Networks & the Internet			
HS-NI-01 Identify issues of network functionality in computational artifact design			
HS-NI-02 Analyze issues of network functionality in computational artifact design.			
HS-NI-03 Identify issues of unauthorized access and cybersecurity in computational artifact design.			
HS-NI-04 Analyze issues of unauthorized access and cybersecurity in computational artifact design.			
HS-NI-05 Explain tradeoffs when selecting and implementing cybersecurity recommendations for various scenarios based on factors such as efficiency, feasibility, and ethical impacts.			
Data and Analysis			
HS-DA-01 Identify patterns in data representing complex systems with select data analysis tools and techniques.			
HS-DA-02 Select appropriate data collection tools and techniques.			
HS-DA-03 Compile data sets that support a claim or communicate information.			
HS-DA-04 Identify the abilty of models and simulations to test hypotheses.			
HS-DA-05 Formulate hypotheses with select models and simulations.			
Algorithms and Programming			
HS-AP-01 Identify artificial intelligence algorithms.			
HS-AP-02 Solve computational problems with classic algorithms.			
HS-AP-03 Evaluate algorithms in terms of their efficiency, correctness, and clarity.			
HS-AP-04 Select an appropriate data structure for information of a given problem.	[17]		
HS-AP-05 Illustrate the flow of execution of a recursive algorithm.	[18]		
HS-AP-06 Identify a large-scale computational problem.			
HS-AP-07 Analyze general patterns applicable to a solution.			
HS-AP-08 Create computational artifacts with pre-existing procedures, external components, libraries and APIs.	[19]		
HS-AP-09 Create a computational artifact through an industry-standard process.			
HS-AP-10 Justify that a computational artifact meets design specifications with systematic testing and debugging methods.			
HS-AP-11 Construct a computational artifact as a team through industry appropriate collaborative tools and processes.			
HS-AP-12 Compose standard documentation for computational artifacts to make it easier to follow, test, and debug.	[20]		
HS-AP-13 Modify an existing computational artifact for additional functionality.		[21]	
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			
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.			
HS-IC-03 Predict how computational innovations that revolutionized aspects of our culture might evolve.			
HS-IC-04 Evaluate how equity, access, and influence impact distribution of computing resources in a global society.			
HS-IC-05 Create computational artifacts to ensure accessibility and reduce computational bias.			
HS-IC-06 Utilize tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields.			

[1] All missions utilize hardware and software that collect and exchange data

[2] 3.5 introduces the use of the debugger
Code tracing Charts are introduced in the teachers' manual

[3] The teachers' manual introduces pseudocodes and flowcharts

[4] 3.8 introduces the use of variables
5.5 discusses descriptive naming of variables
Mission 4 introduces different data types

[5] Mission 6 uses but does not discuss nested loops
Mission 9 introduces compound conditionals

[6] 5.5 introduces the use of comments
Code Tracing charts, Flowcharts and pseudocodes are all introduced in the teachers' manual and the use of all of these would accomplish this objective

[7] 5.5 introduces the use of comments
7.6 and 10.6 discuss readability and maintenance of codes

[8] All missions use libraries and any time a new one is introduced it is explained

[9] 5.5 introduces the use of comments in the codes

[10] Mission 2 and the teachers' manual discuss troubleshooting techniques
Code Tracing Charts are introduced in the teachers' manual as well and a collection of these could be used by all to help identify and fix errors

[11] These are the remixes that are introduced in Mission 4

[12] 7.5 begins the use of lists

[13] These can be the remixes depending on the rubric the teacher gives

[14] Pseudocodes and Flowcharts are introduced in the teachers' manual and would accomplish this objective

[15] Pseudocodes and Flowcharts are introduced in the teachers' manual and would accomplish this objective

[16] 3.5 introduces the debugger
Mission 2 and the teachers' manual discuss troubleshooting techniques

[17] Mission 4 introduces data types

Mission 7 introduces the use of lists

[18] Flowcharts are introduced in the teachers' manual

[19] All missions use libraries and any time a new one is introduced it is explained

[20] 5.5 introduces the use of comments

[21] This is accomplished with the remixes that begin in Mission 4