Mississippi College- and Career-Readiness Standards for Computer Science Alignment with CodeX Curriculum

1B (Grades 3-5)	Unit 1	Unit 2	Unit 3
Computing Systems			
CS.1B.1 Describe how internal and external parts of computing devices function to form a system.	[1]		
CS.1B.2 Model how computer hardware and software work together as a system to accomplish tasks.			
CS.1B.3 Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies.	[2]		
Networks & the Internet			
NI.1B.1 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.			
NI.1B.2 Discuss real-world cybersecurity problems and how personal information can be protected.			
Data and Analysis			
DA.1B.1 Organize and present collected data visually to highlight relationships and support a claim.			
DA.1B.2 Use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate an idea.			
Algorithms and Programming			
AP.1B.1 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.			
AP.1B.2 Create programs that use variables to store and modify data.	[3]		
AP.1B.3 Create programs that include sequences, events, loops, and conditionals.	[4]		
AP.1B.4 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.			
AP.1B.4 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.	[5]		
AP.1B.5 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.			
AP.1B.6 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.			
AP.1B.7 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.	[6]		
AP.1B.8 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.			
AP.1B.9 Describe choices made during program development using code comments, presentations, and demonstrations.	[7]		
Impacts of Computing			
IC.1B.1 Discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices.			
IC.1B.2 Brainstorm ways to improve the accessibility and usability of technology products for the diverse needs and wants of users.			
IC.1B.3 Seek diverse perspectives for the purpose of improving computational artifacts.			
IC.1B.4 Use public domain or creative commons media, and refrain from copying or using material created by others without permission.			

Mississippi College- and Career-Readiness Standards for Computer Science Alignment with CodeX Curriculum

2 (Grades 6-8)	Unit 1	Unit 2	Unit 3
Computing Systems			
CS.2.1 Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices.			
CS.2.2 Design projects that combine hardware and software components to collect and exchange data.			
CS.2.3 Systematically identify and fix problems with computing devices and their components.	[8]		
Networks & the Internet			
NI.2.1 Model the role of protocols in transmitting data across networks and the Internet.			
NI.2.2 Explain how physical and digital security measures protect electronic information.			
NI.2.3 Apply multiple methods of encryption to model the secure transmission of information.			
Data and Analysis			
DA.2.1 Represent data using multiple encoding schemes.			
DA.2.2 Collect data using computational tools and transform the data to make it more useful and reliable.			
DA.2.3 Refine computational models based on the data they have generated.			
Algorithms and Programming			
AP.2.1 Use flowcharts and/or pseudocode to address complex problems as algorithms.	[9]		
AP.2.2 Create clearly named variables that represent different data types and perform operations on their values.	[10]		
AP.2.3 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	[11]		
AP.2.4 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.			
AP.2.5 Create procedures with parameters to organize code and make it easier to reuse.			[12]
AP.2.6 Seek and incorporate feedback from team members and users to refine a solution that meets user needs.			
AP.2.7 Incorporate existing code, media, and libraries into original programs, and give attribution.	[13]		
AP.2.8 Systematically test and refine programs using a range of test cases.			
AP.2.9 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.			
AP.2.10 Document programs in order to make them easier to follow, test, and debug.	[14]		
Impacts of Computing			
IC.2.1 Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.			
IC.2.2 Discuss issues of bias and accessibility in the design of existing technologies.			
IC.2.3 Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.			
IC.2.4 Describe tradeoffs between allowing information to be public and keeping information private and secure.			

Mississippi College- and Career-Readiness Standards for Computer Science Alignment with Codex Curriculum

3A (Grades 9-10)	Unit 1	Unit 2	Unit 3
Computing Systems			
CS.3A.1 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.			
CS.3A.2 Compare levels of abstraction and interactions between application software, system software, and hardware layers.			
CS.3A.3 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.	[15]		
Networks & the Internet			
NI.3A.1 Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.			
NI.3A.2 Give examples to illustrate how sensitive data can be affected by malware and other attacks.			
NI.3A.3 Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts.			
NI.3A.4 Compare various security measures, considering tradeoffs between the usability and security of a computing system.			
NI.3A.5 Explain tradeoffs when selecting and implementing cybersecurity recommendations.			
Data and Analysis			
DA.3A.1 Translate between different bit representations of real-world phenomena, such as characters, numbers, and images.			
DA.3A.2 Evaluate the tradeoffs in how data elements are organized and where data is stored.			
DA.3A.3 Create interactive data visualizations using software tools to help others better understand real-world phenomena.			
DA.3A.4 Create computational models that represent the relationships among different elements of data collected from a phenomenon or process.			
Algorithms and Programming			
AP.3A.1 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.	[16]		
AP.3A.2 Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.		[17]	
AP.3A.3 Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.			
AP.3A.4 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.		[18]	
AP.3A.5 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.			
AP.3A.6 Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.		[19]	
AP.3A.7 Systematically design and develop programs for broad audiences by incorporating feedback from users.			
AP.3A.8 Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries.			
AP.3A.9 Evaluate and refine computational artifacts to make them more usable and accessible.			
AP.3A.10 Design and develop computational artifacts working in team roles using collaborative tools.			
AP.3A.11 Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.	[20]		
Impacts of Computing			
IC.3A.1 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.			
IC.3A.2 Test and refine computational artifacts to reduce bias and equity deficits.			
IC.3A.3 Demonstrate ways a given algorithm applies to problems across disciplines.			
IC.3A.4 Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields.			
IC.3A.5 Explain the beneficial and harmful effects that intellectual property laws can have on innovation.			
IC.3A.6 Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users.			
IC.3A.7 Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.			

[1] Mission 2 begins teaching this

[2] Mission 2 and the teachers' manual discuss troubleshooting techniques

[3] 3.8 begins the use of variables

[4] Mission 4 begins the use of all of these

[5] These are the remixes that begin in Mission 4

[6] 3.5 introduces the debugger

[7] 5.5 introduces the use of comments

[8] Troubleshooting is taught in Mission 2 and the teachers' manual

[9] Flowcharts and pseudocodes are introduced in the teachers' manual

[10] 3.8 begins the use of variables
5.5 discusses descriptive naming of variables

[11] Mission 6 begins the use of nested loops but does not describe them
Mission 9 begins the use of compound conditionals

[12] 9.3 begins the technique of creating your own functions
Mission 10 discusses organizing code for resuse

[13] All missions use libraries and every time a new one is introduced, they are explained.

[14] 5.5 introduces the use of comments

[15] Code Tracing Charts can accomplish this and they are introduced in the teachers' manual

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

[17] 7.5 begins the use of lists

[18] This can be the remixes depending on the rubric the teacher provides the students

[19] These are accomplished in the remixes

[20] 5.5 introduces the use of comments