Michigan Computer Science Standards Alignment with CodeX Curriculum

1B (Grades 3-5)	Unit 1	Unit 2	Unit 3
Computing Systems			
1B-CS-01 Describe how internal and external parts of computing devices function to form a system.	[1]		
1B-CS-02 Model how computer hardware and software work together as a system to accomplish tasks.			
1B-CS-03 Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies.	[2]		
Networks & the Internet			
1B-NI-04 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.			
1B-NI-05 Discuss real-world cybersecurity problems and how personal information can be protected.			
Data and Analysis			
1B-DA-06 Organize and present collected data visually to highlight relationships and support a claim.			
1B-DA-07 Use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate an idea.			
Algorithms and Programming			
1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.			
1B-AP-09 Create programs that use variables to store and modify data.	[3]		
1B-AP-10 Create programs that include sequences, events, loops, and conditionals.	[4]		
1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.			
1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.	[5]		
1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.			
1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.			
1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.	[6]		
1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.			
1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.	[7]		
Impacts of Computing			
1B-IC-18 Discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices.			
1B-IC-19 Brainstorm ways to improve the accessibility and usability of technology products for the diverse needs and wants of users.			
1B-IC-20 Seek diverse perspectives for the purpose of improving computational artifacts.			
1B-IC-21 Use public domain or creative commons media, and refrain from copying or using material created by others without permission.			

Michigan Computer Science Standards Alignment with CodeX Curriculum

2 (Grades 6-8)	Unit 1	Unit 2	Unit 3
Computing Systems			
2-CS-01 Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices.			
2-CS-02 Design projects that combine hardware and software components to collect and exchange data.			
2-CS-03 Systematically identify and fix problems with computing devices and their components.	[8]		
Networks & the Internet			
2-NI-04 Model the role of protocols in transmitting data across networks and the Internet.			
2-NI-05 Explain how physical and digital security measures protect electronic information.			
2-NI-06 Apply multiple methods of encryption to model the secure transmission of information.			
Data and Analysis			
2-DA-07 Represent data using multiple encoding schemes.			
2-DA-08 Collect data using computational tools and transform the data to make it more useful and reliable.			
2-DA-09 Refine computational models based on the data they have generated.			
Algorithms and Programming			
2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms.	[9]		
2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.	[10]		
2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	[11]		
2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.			
2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.			[12]
2-AP-15 Seek and incorporate feedback from team members and users to refine a solution that meets user needs.			
2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.	[13]		
2-AP-17 Systematically test and refine programs using a range of test cases.			
2-AP-18 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.			
2-AP-19 Document programs in order to make them easier to follow, test, and debug.	[14]		
Impacts of Computing			
2-IC-20 Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.			
2-IC-21 Discuss issues of bias and accessibility in the design of existing technologies.			
2-IC-22 Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.			
2-IC-23 Describe tradeoffs between allowing information to be public and keeping information private and secure.			

Michigan Computer Science Standards Alignment with CodeX Curriculum

3A (Grades 9-10)	Unit 1	Unit 2	Unit 3
Computing Systems			
3A-CS-01 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.			
3A-CS-02 Compare levels of abstraction and interactions between application software, system software, and hardware layers.			
3A-CS-03 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.	[15]		
Networks & the Internet			
3A-NI-04 Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.			
3A-NI-05 Give examples to illustrate how sensitive data can be affected by malware and other attacks.			
3A-NI-06 Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts.			
3A-NI-07 Compare various security measures, considering tradeoffs between the usability and security of a computing system.			
3A-NI-08 Explain tradeoffs when selecting and implementing cybersecurity recommendations.			
Data and Analysis			
3A-DA-09 Translate between different bit representations of real-world phenomena, such as characters, numbers, and images.			
3A-DA-10 Evaluate the tradeoffs in how data elements are organized and where data is stored.			
3A-DA-11 Create interactive data visualizations using software tools to help others better understand real-world phenomena.			
3A-DA-12 Create computational models that represent the relationships among different elements of data collected from a phenomenon or process.			
Algorithms and Programming			
3A-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.	[16]		
3A-AP-14 Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.		[17]	
3A-AP-15 Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.			
3A-AP-16 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.		[18]	
3A-AP-17 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.			
3A-AP-18 Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.		[19]	
3A-AP-19 Systematically design and develop programs for broad audiences by incorporating feedback from users.			
3A-AP-20 Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries.			
3A-AP-21 Evaluate and refine computational artifacts to make them more usable and accessible.			
3A-AP-22 Design and develop computational artifacts working in team roles using collaborative tools.			
3A-AP-23 Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.	[20]		
Impacts of Computing			
3A-IC-24 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.			
3A-IC-25 Test and refine computational artifacts to reduce bias and equity deficits.			
3A-IC-26 Demonstrate ways a given algorithm applies to problems across disciplines.			
3A-IC-27 Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields.			
3A-IC-28 Explain the beneficial and harmful effects that intellectual property laws can have on innovation.			
3A-IC-29 Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users.			
3A-IC-30 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