Kansas CS Standards Alignment with CodeX Curriculum

	Unit 1	Unit 2	Unit 3
Computing Systems			
5.CS.D.01 Model and communicate how computing devices can be connected to other devices to extend their capabilities.			
5.CS.HS.01 Illustrate how information is translated into binary numbers between software and hardware.			
5.CS.IO.01 Demonstrate proper use of grade level appropriate input devices and produce digital artifacts selective publication based on audience/purpose.	[1]		
5.CS.T.01 Using accurate terminology, identify simple hardware and software problems that may occur during everyday use.			
Networks & the Internet			
5.NI.NCO.01 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.			
5.NI.C.01 Discuss real-world cybersecurity problems and how personal information can be protected.			
Data and Analysis			
5.DA.S.01 Evaluate trade-offs of file types, storage requirements, and sharing requirements, including comparisons of availability and quality.			
5.DA.C.01 Select the appropriate tool to collect relevant and reliable data that solves a problem.			
5.DA.CVT.01 Organize and present collected data to highlight comparisons and support a claim.			
5.DA.IM.01 Use data to discover or propose cause and effect relationships, predict outcomes, or communicate an idea.			
Algorithms and Programming			
5.AP.A.01 Analyze and refine multiple algorithms for the same task and determine which algorithm is the most efficient.			
5.AP.V.01 Utilize, create, and modify programs that use, modify, and combine variables with grade level appropriate data.	[2]		
5.AP.C.01 Create programs using a programming language that utilize sequencing, repetition, conditionals, event handlers, and variables to solve a problem or express ideas both independently and collaboratively.	[3]		
5.AP.M.01 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.			
5.AP.M.02 With grade appropriate complexity, modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.	[4]		
5.AP.PD.01 Create a plan using an iterative process for the development of a program that includes others' perspectives and user preferences while solving simple problems.			
5.AP.PD.02 Use proper citations and document when ideas are borrowed and changed for their own use (e.g., using pictures created by others, using music created by others, remixing programming projects).			
5.AP.PD.03 Analyze, debug (identify/fix errors), and create a program that includes sequencing, repetition and variables in a programming language.	[5]		
5.AP.PD.04 Take on varying roles collaborating with peers to give feedback at different stages of program development, including design and implementation.			
Impacts of Computing			
5.IC.C.01 Develop, test, and refine digital artifacts to improve accessibility and usability for a computing device or program.	[6]		
5.IC.SI.01 Develop a code of conduct, explain, and practice grade-level appropriate behavior and responsibilities while participating in an online community. Identify and report inappropriate behavior. (Digital Citizenship - review of all nine components, but focused on Digital Commerce, Digital Law, and Digital Security.			
5.IC.SI.02 As a team, collaborate with people and resources outside of your normal space to include diverse perspectives to improve computational products.			
5.IC.H.01 Identify and explain the evolution of computing technologies that have changed the world.			
5.IC.SLE.01 Observe intellectual property rights and give appropriate credit when using resources.			
5.IC.CP.01 Design a visual product depicting the connections between computer science and other fields.			

Kansas CS Standards Alignment with CodeX Curriculum

Grades 6-8	Unit 1	Unit 2	Unit 3
Computing Systems			
MG.CS.D.01 Develop and implement a process to evaluate existing computing devices and recommend improvements to design based on analysis of how other users interact with the device.			
MG.CS.HS.01 Model a computing system involving multiple considerations and potential tradeoffs of software and hardware, such as functionality, cost, size, speed, accessibility, and aesthetics.			
MG.CS.IO.01 Know and apply grade-level appropriate skills with input and output devices.	[7]		
MG.CS.T.01 Systematically identify, fix, and document increasingly complex software and hardware problems with computing devices and their components.	[8]		
Networks & the Internet			
MG.NI.NCO.01 Explain protocols and their importance to data transmission; model how packets are broken down into smaller pieces and how they are delivered.			
MG.NI.C.01 Evaluate physical and digital procedures that could be implemented to protect electronic data/information; explain the impacts of hacking, ransomware, scams, fake scans, and ethical/legal concerns.			
MG.NI.C.02 Compare the advantages and disadvantages of multiple methods of encryption to model the secure transmission of information.			
Data and Analysis			
MG.DA.S.01 Analyze multiple methods of representing data and choose the most appropriate method for representing data.			
MG.DA.C.01 Develop, implement, and refine a process that utilizes computational tools to collect meaningful data.			
MG.DA.CVT.01 Develop, implement, and refine a process to make data more useful and reliable.			
MG.DA.IM.01 Refine computational models based on the data generated by the models.			
Algorithms and Programming			
MG.AP.A.01 Design algorithms in natural language, flow and control diagrams, comments within code, and/or pseudocode to solve complex problems.	[9]		
MG.AP.V.01 Create programs using variables with purposeful and thoughtful naming conventions for identifiers to improve program readability.	[10]		
MG.AP.C.01 Develop programs that utilize combinations of nested repetition, compound conditionals, procedures without parameters, and the manipulation of variables representing different data types.	[11]		
MG.AP.M.01 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of complex programs.			
MG.AP.PD.01 Seek and incorporate feedback from team members and users to refine a solution to a problem that meets the needs of diverse users.			
MG.AP.PD.02 Incorporate existing code, media, and libraries into original programs of increasing complexity and give attribution.	[12]		
MG.AP.PD.03 Systematically test and refine programs using a range of student created inputs.			
MG.AP.PD.04 Explain how effective communication between participants is required for successful collaboration when developing computational artifacts.			
MG.AP.PD.05 Document text-based programs of increasing complexity in order to make them easier to follow, test, and debug.	[13]		
Impacts of Computing			
MG.IC.C.01 Describe the trade-offs associated with computing technologies (e.g. automation), explaining their effects on economies and global societies, and explore careers related to the field of computer science.			
MG.IC.C.02 Evaluate and improve the design of existing technologies to meet the needs of diverse users and increase accessibility and usability.			
MG.IC.SI.01 Communicate and publish key ideas and details individually or collaboratively in a way that informs, persuades, and/or entertains using a variety of digital tools and media-rich resources. Describe and use safe, appropriate, and responsible practices (netiquette) when participating in online communities (e.g., discussion groups, blogs, social networking sites).			
MG.IC.H.01 Identify and describe how the prominent figures in computer science have impacted and/or progressed the field.			
MG.IC.SLE.01 Discuss the social impacts and ethical considerations associated with cybersecurity, including the positive and malicious purposes of hacking.			
MG.IC.CP.01 Formulate a computer-science based solution for a problem or issue by gathering input from local / regional industry members.			

Kansas CS Standards Alignment with CodeX Curriculum

Grades 9-12, all students	Unit 1	Unit 2	Unit 3
Computing Systems			
L1.CS.D.01 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.			
L1.CS.HS.01 Compare levels of abstraction and interactions between application software, system software, and hardware layers.			
L1.CS.HS.02 Compare computer systems and determine advantages and drawbacks of each system.			
L1.CS.IO.01 Demonstrate efficient use of input and output devices.			
L1.CS.T.01 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.	[14]		
Networks & the Internet			
L1.NI.NCO.01 Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.			
L1.NI.NCO.02 Compare various security measures, considering tradeoffs between the usability and security of a computing system.			
L1.NI.C.01 Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts.			
L1.NI.C.02 Explain tradeoffs when selecting and implementing cybersecurity recommendations.			
Data and Analysis			
L1.DA.S.01 Analyze storage types and locations.			
L1.DA.S.02 Evaluate the tradeoffs in how data elements are organized and where data is stored.			
L1.DA.C.01 Collect and analyze data.			
L1.DA.CVT.01 Create interactive data visualizations using software tools to help others better understand real-world phenomena.			
L1.DA.IM.01 Create computational models that represent the relationships among different elements of data collected from a phenomenon or process.			
Algorithms and Programming			
L1.AP.A.01 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.	[15]		
L1.AP.V.01 Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.		[16]	
L1.AP.C.01 Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.			
L1.AP.C.02 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.		[17]	
L1.AP.C.03 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.			
L1.AP.M.01 Create computational artifacts by systematically organizing, manipulating and/or processing data.			
L1.AP.M.02 Systematically design and develop programs for broad audiences by incorporating feedback from users.			
L1.AP.PD.01 Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries.			
L1.AP.PD.02 Evaluate and refine computational artifacts to make them more usable and accessible.			
L1.AP.PD.03 Design and develop computational artifacts working in team roles using collaborative tools.			
L1.AP.PD.04 Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.	[18]		
Impacts of Computing			
L1.IC.C.01 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.			
L1.IC.C.02 Test and refine computational artifacts to reduce bias and equity deficits.			
L1.IC.C.03 Demonstrate how a given algorithm applies to problems across disciplines.			
L1.IC.SI.01 Compare and contrast the benefits and drawbacks of social media.			
L1.IC.H.01 Hypothesize the impact of the innovations of computing systems for the next decade.			
L1.IC.SLE.01 Explain the beneficial and harmful effects that intellectual property laws can have on innovation.			
L1.IC.SLE.02 Explain the privacy concerns related to the collection and generation of data through automated processes (e. g., how businesses, social media, and the government collects and uses data) that may not be evident to users.			
L1.IC.SLE.03 Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.			
L1.IC.CP.01 Explore computing, software, and data storage systems in local industries.			

Kansas CS Standards Alignment with CodeX Curriculum

Grades 9-12 (Students who wish to pursue computer science beyond what is expected of all students.)	Unit 1	Unit 2	Unit 3
Computing Systems			
L2.CS.D.01 Describe how internal and external parts of computing devices function to form a system.			
L2.CS.HS.01 Categorize the roles of operating system software.			
L2.CS.HS.02 Compare options for building a computer systems and determine advantages and drawbacks of each piece and how it will affect the overall performance.			
L2.CS.IO.01 Demonstrate use of course specific advanced input and output devices related to field.			
L2.CS.T.01 Illustrate ways computing systems implement logic, input, and output through hardware components.			
Networks & the Internet			
L2.NI.NCO.01 Describe the issues that impact network functionality (e.g., bandwidth, load, delay, topology).			
L2.NI.NCO.02 Give examples to illustrate how sensitive data can be affected by malware and other attacks.			
L2.NI.C.01 Compare ways software developers protect devices and information from unauthorized access.			
L2.NI.C.02 Use encryption and decryption algorithms to transmit/ receive an encrypted message.			
Data and Analysis			
L2.DA.S.01 Translate and compare different bit representations of data types, such as characters, numbers, and images.			
L2.DA.S.02 Analyze file systems created for keeping track of files on the hard disk.			
L2.DA.C.01 Select data collection tools and techniques to generate data sets that support a claim or communicate information.			
L2.DA.CVT.01 Use data analysis tools and techniques to identify patterns in data representing complex systems.			
L2.DA.IM.01 Evaluate the ability of models and simulations to test and support the refinement of hypotheses. (e.g., flocking behaviors, life cycles, etc.)			
Algorithms and Programming			
L2.AP.A.01 Describe how artificial intelligence algorithms drive many software and physical systems (e.g., digital advertising, autonomous robots, computer vision, pattern recognition, text analysis).			
L2.AP.A.02 Describe how artificial intelligence drives many software and physical systems.			
L2.AP.A.03 Critically examine and trace classic algorithms (e.g., selection sort, insertion sort, binary search, linear search).			
L2.AP.A.04 Implement an artificial intelligence algorithm to play a game against a human opponent or solve a problem.			
L2.AP.A.05 Use and adapt classic algorithms to solve computational problems.			
L2.AP.A.06 Evaluate algorithms in terms of their efficiency, correctness, and clarity.			
L2.AP.V.01 Compare and contrast simple data structures and their uses to simplify solutions, generalizing computational problems instead of repeatedly using primitive variables.		[19]	
L2.AP.C.01 Trace the execution of repetition (e.g., loops, recursion), illustrating output and changes in values of named variables.	[20]		
L2.AP.M.01 Construct solutions to problems using student-created components, such as procedures, modules and/or objects.	[21]		
L2.AP.M.02 Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.			
L2.AP.M.03 Demonstrate code reuse by creating programming solutions using libraries and APIs.	[22]		
L2.AP.PD.01 Plan and develop programs for broad audiences using a software life cycle process.	[23]		
L2.AP.PD.02 Explain security issues that might lead to compromised computer programs.			
L2.AP.PD.03 Develop programs for multiple computing platforms.			
L2.AP.PD.04 Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project.			
L2.AP.PD.05 Develop and use a series of test cases to verify that a program performs according to its design specifications.			
L2.AP.PD.06 Modify an existing program to add additional functionality and discuss intended and unintended implications (e. g., breaking other functionality).	[24]		
L2.AP.PD.07 Evaluate key qualities of a program through a process such as a code review.	[25]		
L2.AP.PD.08 Compare multiple programming languages and discuss how their features make them suitable for solving different types of problems.			
Impacts of Computing			
L2.IC.C.01 Evaluate the beneficial and harmful effects that computational artifacts and innovations have on society.			
L2.IC.C.02 Evaluate the impact of equity, access, and influence on the distribution of computing resources in a global society.			
L2.IC.C.03 Design and implement a study that evaluates or predicts how computing has revolutionized an aspect of our culture and how it might evolve (e.g., education, healthcare, art/entertainment, energy).			
L2.IC.SI.01 Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields.			
L2.IC.H.01 Analyze trends of computing and how those trends have changed over time.			
L2.IC.SLE.01 Debate laws and regulations that impact the development and use of software.			
L2.IC.SLE.02 Determine ways to test the validity of information located online.			
L2.IC.SLE.03 Evaluate the social and economic consequences of how law and ethics interact with digital aspects of privacy, data, property, information, and identity.			
L2.IC.CP.01 Collaborate with local industry partners to design and implement a viable mentorship.			

[1] The remixes that begin in Mission 4 accomplish this

[2] The remixes cover this

[3] Mission 3 begins the use of variables
Mission 4 begins the use of conditionals, sequencing, repetition and more

[4] These are the remixes at the end of the Missions

[5] 3.5 the debugger is introduced

[6] These are the remixes

[7] The CodeX is all about input and output skills

[8] Mission 2 discusses troubleshooting techniques as well as the teachers' manual

[9] Flowcharts and pseudocodes are introduced in the teachers' manual.
5.5 introduces the use of comments

[10] 3.8 introduces the use of variables
5.5 discusses descriptive names for variables

[11] Mission 6 uses nested loops but does not discuss them
Mission 9 introduces compound conditional
Mission 6 begins the manipulation of variables

[12] All missions use libraries and as new ones are introduced they are explained

[13] 5.5 introduces the use of comments in your codes

[14] Code Tracing Charts cover this and they are introduced in the teachers' manual

[15] These are the remixes

[16] 7.5 introduces the use of lists

[17] Remixes can cover this depending on the rubric the teacher uses

[18] These are the flowcharts and pseudocodes as well as comments in the program itself

[19] 7.5 introduces the use of lists

[20] The debugger can do this as well as Code Tracing Charts that are introduced in the teachers' manual

[21] These are the remixes

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

[23] These can be accomplished with the remixes depending on teacher directions

[24] These are the remixes

[25] Code Tracing Charts are introduced in the teachers' manual