Illinois CS Standards Alignment with CodeX Curriculum

Grades 3-5	Unit 1	Unit 2	Unit 3
Computing Systems			
3-5.CS.01 Describe how internal and external parts of computing devices function to form a system.	[1]		
3-5.CS.02 Model how computer hardware and software work together as a system to accomplish tasks. Discuss task specific embedded systems.			
3-5.CS.03 Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies.	[2]		
Networks & the Internet			
3-5.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.			
3-5.NI.05 Discuss real-world cybersecurity problems and how personal information can be protected.			
Data and Analysis			
3-5.DA.06 Organize and present collected data visually to highlight relationships and support a claim.			
3-5.DA.07 Use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate an idea.			
Algorithms and Programming			
3-5.AP.08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.			
3-5.AP.09 Create programs that use variables to store and modify data.	[3]		
3-5.AP.10 Create programs that include sequences, events, loops, and conditionals.	[4]		
3-5.AP.11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.	[5]		
3-5.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.	[6]		
3-5.AP.13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.			
3-5.AP.14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.			
3-5.AP.15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.	[7]		
3-5.AP.16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.			
3-5.AP.17 Describe choices made during program development using code comments, presentations, and demonstrations.	[8]		
Impacts of Computing			
3-5.IC.18 Discuss computing technologies that have changed the world and express how those technologies influence, and are influenced by, cultural practices.			
3-5.IC.19 Brainstorm ways to improve the accessibility and usability of technology products for the diverse needs and wants of users.			
3-5.IC.20 Seek diverse perspectives for the purpose of improving computational artifacts.			
3-5.IC.21 Use public domain or Creative Commons media and refrain from copying or using material created by others without permission.			

Illinois CS Standards Alignment with CodeX Curriculum

Grades 6-8	Unit 1	Unit 2	Unit 3
Computing Systems			
6-8.CS.01 Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices.			
6-8.CS.02 Design projects that combine hardware and software components to collect and exchange data.			[9]
6-8.CS.03 Systematically identify and fix problems with computing devices and their components.	[10]		
Networks & the Internet			
6-8.NI.04 Model the role of protocols in transmitting data across networks and the internet.			
6-8.NI.05 Explain how physical and digital security measures protect electronic information.			
6-8.NI.06 Apply multiple methods of encryption to model the secure transmission of information.			
Data and Analysis			
6-8.DA.07 Represent data using multiple encoding schemes.			
6-8.DA.08 Collect data using computational tools and transform the data to make it more useful and reliable.			
6-8.DA.09 Refine computational models based on the data they have generated.			
6-8.DA.10 Evaluate the misuse of data and impact of distorted outcomes.			
Algorithms and Programming			
6-8.AP.11 Use flowcharts or pseudocode to address complex problems as algorithms.	[11]		
6-8.AP.12 Perform operations on student-created variables that possess descriptive names and represent different data types.	[12]		
6-8.AP.13 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	[13]		
6-8.AP.14 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.		[14]	
6-8.AP.15 Create procedures with parameters to organize code and make it easier to reuse.			[15]
6-8.AP.16 Seek and incorporate feedback from team members and users to refine a solution that meets user needs.			
6-8.AP.17 Incorporate existing code, media, and libraries into original programs and give attribution.	[16]		
6-8.AP.18 Systematically test and refine programs using a range of test cases.			
6-8.AP.19 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.			
6-8.AP.20 Document programs to make them easier to follow, test, and debug.	[17]		
Impacts of Computing			
6-8.IC.21 Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.			
6-8.IC.22 Discuss issues of bias and accessibility in the design of existing technologies.			
6-8.IC.23 Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.			
6-8.IC.24 Describe tradeoffs between allowing information to be public and keeping information private and secure.			

Illinois CS Standards Alignment with CodeX Curriculum

Grades 9-10	Unit 1	Unit 2	Unit 3
Computing Systems			
9-10.CS.01 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.			
9-10.CS.02 Compare levels of abstraction and interactions between application software, system software, and hardware layers.			
9-10.CS.03 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.	[18]		
Networks & the Internet			
9-10.NI.04 Evaluate the scalability and reliability of networks by describing the relationship between routers, switches, servers, topology, and addressing.			
9-10.NI.05 Give examples to illustrate how sensitive data can be affected by malware and other attacks.			
9-10.NI.06 Compare various security measures, considering tradeoffs between the usability and security of a computing system.			
9-10.NI.07 Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts.			
9-10.NI.08 Explain tradeoffs when selecting and implementing cybersecurity recommendations.			
Data and Analysis			
9-10.DA.09 Translate between different bit representations of real-world phenomena, such as characters, numbers, and images.			
9-10.DA.10 Evaluate the tradeoffs in how data elements are organized and stored.			
9-10.DA.11 Create interactive data visualizations using software tools to help others better understand real-world phenomena.			
9-10.DA.12 Create computational models that represent the relationships among different elements of data collected from a phenomenon or process.			
Algorithms and Programming			
9-10.AP.13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.	[19]		
9-10.AP.14 Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.		[20]	
9-10.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.			
9-10.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.		[21]	
9-10.AP.17 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, or objects.		[22]	
9-10.AP.18 Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.	[23]		
9-10.AP.19 Systematically design and develop programs for broad audiences by incorporating feedback from users.			
9-10.AP.20 Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries.			
9-10.AP.21 Evaluate and refine computational artifacts to make them more usable and accessible.			
9-10.AP.22 Design and develop computational artifacts working in team roles using collaborative tools.			
9-10.AP.23 Document design decisions using text, graphics, presentations, or demonstrations in the development of complex programs.	[24]		
9-10.AP.24 Describe the characteristics and evaluate the impact of human computer interaction.			
Impacts of Computing			
9-10.IC.25 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.			
9-10.IC.26 Test and refine computational artifacts to reduce bias and equity deficits.			
9-10.IC.27 Demonstrate ways a given algorithm applies to problems across disciplines.			
9-10.IC.28 Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields.			
9-10.IC.29 Explain the beneficial and harmful effects that intellectual property laws can have on innovation.			
9-10.IC.30 Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users.			
9-10.IC.31 Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.			

Illinois CS Standards Alignment with CodeX Curriculum

Grades 11-12	Unit 1	Unit 2	Unit 3
Computing Systems			
11-12.CS.01 Compare the characteristics and uses of traditional and emerging computing devices and systems.			
11-12.CS.02 Categorize the roles of operating system software.			
11-12.CS.03 Illustrate ways computing systems implement logic, input, and output through hardware components.	[25]		
Networks & the Internet			
11-12.NI.04 Describe the issues that impact network functionality (e.g., bandwidth, load, delay, topology).			
11-12.NI.05 Compare ways software developers protect devices and information from unauthorized access.			
Data and Analysis			
11-12.DA.06 Use data analysis tools and techniques to identify patterns in data representing complex systems.			
11-12.DA.07 Select data collection tools and techniques to generate data sets that support a claim or communicate information.			
11-12.DA.08 Analyze the ways in which automated data collection is utilized in society.			
11-12.DA.09 Evaluate the ability of models and simulations to test and support the refinement of hypotheses.			
Algorithms and Programming			
11-12.AP.10 Describe how artificial intelligence drives many software and physical systems.			
11-12.AP.11 Implement an artificial intelligence algorithm to play a game against a human opponent or solve a problem.			
11-12.AP.12 Use and adapt classic algorithms to solve computational problems.			
11-12.AP.13 Evaluate algorithms in terms of their efficiency, correctness, and clarity.			
11-12.AP.14 Compare and contrast fundamental data structures and their uses.			
11-12.AP.15 Illustrate the flow of execution of a recursive algorithm.	[26]		
11-12.AP.16 Construct solutions to problems using student-created components, such as procedures, modules, or objects.		[27]	
11-12.AP.17 Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.			
11-12.AP.18 Demonstrate code reuse by creating programming solutions using libraries and application programming interfaces.	[28]		
11-12.AP.19 Plan and develop programs for broad audiences using a software life cycle process.			
11-12.AP.20 Demonstrate conversion of source code into machine code using compliers or interpreters.			
11-12.AP.21 Explain security issues that might lead to compromised computer programs.			
11-12.AP-22 Develop programs for multiple computing platforms.			
11-12.AP-23 Use version control systems, integrated development environments, and collaborative tools and practices (code documentation) in a group software project.			
11-12.AP.24 Develop and use a series of test cases to verify that a program performs according to its design specifications.			
11-12.AP.25 Discuss social, economic, and ethical consequences of malfunctional software and software updates.			
11-12.AP.26 Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality).	[29]		
11-12.AP.27 Evaluate key qualities of a program through a process such as a code review.	[30]		
11-12.AP.28 Compare multiple programming languages and discuss how their features make them suitable for solving different types of problems.			
Impacts of Computing			
11-12.IC.29 Evaluate computational artifacts to maximize their beneficial effects and minimize harmful effects on society.			
11-12.IC.30 Evaluate the impact of equity, access, and influence on the distribution of computing resources in a global society.			
11-12.IC.31 Predict how computational innovations that have revolutionized aspects of our culture might evolve.			
11-12.IC.32 Debate laws and regulations that impact the development and use of software.			

[1] Mission 2 describes this and anytime a new sensor is introduced it discuss this

[2] Troubleshooting techniques are introduced in the teachers' manual as well as in Mission 2

[3] 3.8 begins the use of variables

[4] Mission 4 begins the use these

[5] flowcharts, pseudocodes and Code Tracing Charts do this

[6] Mission 4 begins the use of remixes

[7] 3.5 introduces the debugger

[8] 5.5 introduces the use of comments

[9] The use of our sensors does this

[10] Mission 2 discusses troubleshooting as does the teachers' manual

[11] This is discussed in the teachers' manual

[12] 5.5 discusses using descriptive naming for variables
4.2 discusses different data types

[13] Mission 6 uses nested loops but does not discuss it.
Mission 9 uses compound conditionals

[14] Flowcharts, pseudocodes and Code Tracing Charts do this

[15] Use of comments and Code Tracing Charts accomplish this

[16] All mission use and explain the use of libraries

[17] 5.5 begins the use of comments

[18] Code Tracing Charts do this

[19] Remixes do this

[20] 7.5 begins the use of lists

[21] This can be covered with remixes depending on the teacher's rubric

[22] Flowcharts, pseudocodes and Code Tracing Charts do this

[23] These are the remixes

[24] Flowcharts, pseudocodes and Code Tracing Charts do this
Comments added also help with this

[25] Mission 2 introduces this
Any time a new sensor is introduced this is explained

[26] Flowcharts do this

[27] The student created Remixes with their flowcharts and pseudocodes accomplishes this

[28] All missions use and explain the use of libraries

[29]
Remixes are introduced in Mission 4

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