

Kentucky Academic Standards for Computer Science Alignment with CodeX Curriculum

By the end of Grade 8, students will be able to:	Unit 1	Unit 2	Unit 3
Networks & the Internet			
M-NI-01 Model the role of protocols in transmitting data across networks and the Internet.			
M-NI-02 Model how information is disguised using different methods of encryption to secure it during transmission from one point to another.			
M-NI-03 Explain how physical and digital security practices and measures proactively address the threat of breaches to personal and private data.			
Data and Analysis			
M-DA-01 Store data using multiple encoding methods.			
M-DA-02 Collect data using computational tools and transform the data to make it more useful and reliable.			
M-DA-03 Refine computational models based on the data they have generated.			
Algorithms and Programming			
M-AP-01 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.			
M-AP-02 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.			
M-AP-03 Seek and incorporate feedback from team members and users to refine a solution that meets user needs.			
M-AP-04 Create flowcharts and/or pseudocode to address complex problems as algorithms.	[1]		
M-AP-05 Create clearly named variables that represent different data types and perform operations on their values.	[2]		
M-AP-06 Create procedures with parameters to organize code and make it easier to reuse.		[3]	
M-AP-07 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	[4]		
M-AP-08 Incorporate existing code, media, and libraries into original programs, and give attribution.	[5]		
M-AP-09 Systematically test and refine programs using a range of test cases.			
M-AP-10 Document programs in order to make them easier to follow, test, and debug.	[6]		
M-AP-11 Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries.			
M-AP-12 Develop a process creating a computational artifact that leads to a minimum viable product followed by reflection, analysis, and iteration.			
Impacts of Computing			
M-IC-01 Discuss issues of bias and accessibility in the design of existing technologies.			
M-IC-02 Compare the positive & negative effects of computing technologies on society.			
M-IC-03 Collaborate with others using appropriate tools at the local, national, and/or international levels.			
M-IC-04 Discuss the benefits and consequences of making information either public or private.			
Computing Systems			
M-CS-01 Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices.			
M-CS-02 Design projects that combine hardware and software components to collect and exchange data.			
M-CS-03 Identify and fix problems with computing devices and their components systematically.	[7]		

Kentucky Academic Standards for Computer Science Alignment with CodeX Curriculum

<i>By the end of Grade 12, students will be able to:</i>	Unit 1	Unit 2	Unit 3
Networks & the Internet			
H-NI-01 Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.			
H-NI-02 Give examples to illustrate how sensitive data can be affected by malware and other attacks.			
H-NI-03 Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts.			
Data and Analysis			
H-DA-02 Collect data using appropriate data collection tools and techniques to support a claim or to communicate information.			
H-DA-04 Explain the privacy concerns related to the collection and generation of data.			
H-DA-05 Use data analysis tools (e.g. formulas and other software data / statistical tools) to process and transform the data to make it more useful and reliable.			
H-DA-06 Use data analysis tools and techniques to identify patterns and analyze data represented in complex systems.			
H-DA-07 Create computational models that represent the relationships among different elements of data.			
H-DA-08 Create interactive data visualizations using software tools to help others better understand real-world phenomena.			
Algorithms and Programming			
H-AP-01 Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries.			
H-AP-02 Use a development process in creating a computational artifact that leads to a minimum viable product followed by reflection, analysis, and iteration.			
H-AP-03 Use functions, data structures or objects to simplify solutions, generalizing computational problems instead of repeated use of simple variables.	[8]		
H-AP-04 Design and iteratively develop event-driven computational artifacts for practical intent, personal expression, or to address a societal issue.	[9]		
H-AP-05 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.			
H-AP-06 Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.			
H-AP-07 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.	[10]		
H-AP-08 Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.		[11]	
H-AP-09 Evaluate and refine computational artifacts to make them more usable and accessible.			
H-AP-10 Systematically design and develop programs for broad audiences by incorporating feedback from users.			
Impacts of Computing			
H-IC-01 Reduce bias and equity deficits through the design of accessible computational artifacts.			
H-IC-02 Evaluate and assess the ways computing impacts personal, ethical, social, economic, and cultural practices.			
H-IC-03 Research how computational innovations that have revolutionized aspects of our culture might have evolved from a need to solve a problem.			
H-IC-04 Explain the beneficial and harmful effects that intellectual property laws can have on innovation.			
H-IC-06 Evaluate the impact of the digital divide (i.e. inequity of computing access, education and influence) on the development of local communities and society.			
H-IC-08 Debate laws and regulations that impact the development and use of software and the protection of privacy.			
Computing Systems			
H-CS-01 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.			
H-CS-02 Compare levels of abstraction and interactions between application software, system software, and hardware layers.			
H-CS-03 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.	[12]		
H-CS-04 Categorize the roles of operating system software.			

Kentucky Academic Standards for Computer Science Alignment with CodeX Curriculum

Standards marked with a (*) for grades 9-12 represent challenging computer science learning expectations (referred to as Challenge Standards) for students with aspirations toward careers and postsecondary studies in computing disciplines.	Unit 1	Unit 2	Unit 3
Networks & the Internet			
H-NI-04 Describe the issues that impact network functionality (e.g., bandwidth, load, delay, topology). *			
H-NI-05 Compare ways software developers protect devices and information from unauthorized access. *			
Data and Analysis			
H-DA-01 Evaluate the tradeoffs in how data elements are organized and where data is stored.*			
H-DA-03 Understand and design database structures to optimize search and retrieval.*			
H-DA-09 Evaluate the ability of models and simulations to test and support the refinement of hypotheses.*			
Algorithms and Programming			
H-AP-11 Design and develop computational artifacts working in team roles using collaborative tools.*			
H-AP-12 Describe how artificial intelligence drives many software and physical systems.*			
H-AP-13 Use and adapt classic algorithms to solve computational problems.*		[13]	
H-AP-14 Evaluate algorithms in terms of their efficiency, correctness, and clarity.*			
H-AP-15 Compare and contrast fundamental data structures and their uses.*			
H-AP-16 Illustrate the flow of execution of a recursive algorithm.*		[14]	
H-AP-17 Construct solutions to problems using student-created components, such as procedures, modules and/or objects.*			
H-AP-18 Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.*			
H-AP-19 Select and employ an appropriate component or library to facilitate programming solutions.*		[15]	
H-AP-20 Develop programs for multiple computing platforms.*			
H-AP-21 Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project.*			
H-AP-22 Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., introducing errors).*	[16]		
H-AP-23 Evaluate key qualities (including correctness, usability, readability, and efficiency) of a program.*			
H-AP-24 Compare multiple programming languages and discuss how their features make them suitable for solving different types of problems.*			
Impacts of Computing			
H-IC-05 Evaluate and design computational artifacts to maximize their benefit to society.*			
H-IC-07 Demonstrate ways a given algorithm applies to problems across disciplines.*			
Computing Systems			
H-CS-05 Illustrate ways computing systems implement logic, input, and output through hardware components.*			

[1] These are introduced in the teachers' manual

[2] 3.8 introduces the use of variables
5.5 discusses descriptive naming of variables
Mission 4 introduces different data types

[3] 7.6 discusses readability and maintenance for reuse of code

[4] Mission 6 uses nested loops but does not describe them
Mission 9 introduces compound conditionals

[5] All lessons use libraries and any time a new library is introduced it is explained

[6] 5.5 introduces the use of comments

[7] 3.5 introduces the debugger

[8] mission 4 begins the use of functions
Mission 7 introduces lists

[9] Remixes can cover this depending on their rubrics the teachers give them

[10] These are the remixes

[11] These are the remixes

[12] Code Tracing Charts can do this

[13] Mission 7 begins the use of algorithms

[14] Flowcharts of loops covers this and flowcharts are introduced in the teachers manual

[15] All of our missions require libraries and beginning in the CodeTrek, sometimes figuring out which library to use is a TODO

[16] These are the remixes