

Computer Science Discoveries

Kentucky Academic Standards (KAS) for Computer Science

This document aligns the Code.org Computer Science Discoveries (CSD) course with the KAS for Computer Science. At least one KAS for Computer Science standard is aligned with each CSD lesson. The structure is as follows:

- **Standard Identifier** (Identifier1, Identifier2) reflects consistent coding for the identification of a standard representing the grade (or grade band), the concept area and the numerated standard number per concept.

- **Example**

Grade Band	Concept	Standard (number)
H	-NI-	03

E = Grades K-5 ES M = Grades 6-8 MS H = Grades 9-12 HS	Algorithms & Programming (AP) Computing Systems (CS) Data & Analysis (DA) Impacts of Computing (IC) Networks & the Internet (NI)	Standard Number per Concept
---	--	--------------------------------

-

- **Standard** (Standard1, Standard2) outlines what students are expected to know and be able to do.

For more information about Kentucky Academic Standards, visit <https://kystandards.org/>.

Standards Alignment

Unit 1: Problem Solving and Computing

Lesson	Identifier1	Standard1	Identifier2	Standard2
Chapter 1: The Problem Solving Process				
01: Intro to Problem Solving	M-AP-02	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	M-AP-01	Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.
02: The Problem Solving Process	M-AP-02	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	M-AP-09	Systematically test and refine programs using a range of test cases.
03: Exploring Problem Solving	M-AP-02	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	M-AP-01	Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.
Chapter 2: Computers and Problem Solving				
04: What is a Computer?	M-CS-02	Design projects that combine hardware and software components to collect and exchange data.		
05: Input and Output	M-CS-02	Design projects that combine hardware and software components to collect and exchange data.		
06: Processing	M-AP-04	Create flowcharts and/or pseudocode to address complex problems as algorithms.	M-CS-02	Design projects that combine hardware and software components to collect and exchange data.
07: Apps and Storage	M-AP-04	Create flowcharts and/or pseudocode to address complex problems as algorithms.	M-IC-02	Compare the positive & negative effects of computing technologies on society.
08: Propose an App	M-AP-04	Create flowcharts and/or pseudocode to address complex problems as algorithms.	M-AP-03	Seek and incorporate feedback from team members and users to refine a solution that meets user needs.

Unit 2: Web Development

Lesson	Identifier1	Standard1	Identifier2	Standard2
Chapter 1: Web Content and HTML				
01: Exploring Websites	M-IC-02	Compare the positive & negative effects of computing technologies on society.		
02: Websites for Expression	M-AP-02	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	M-IC-02	Compare the positive & negative effects of computing technologies on society.
03: Intro to HTML	M-AP-02	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.		
04: Headings	M-AP-02	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.		
05: Digital Footprint	M-IC-02	Compare the positive & negative effects of computing technologies on society.	M-IC-04	Discuss the benefits and consequences of making information either public or private.
06: Lists	M-AP-08	Incorporate existing code, media, and libraries into original programs, and give attribution.	M-AP-09	Systematically test and refine programs using a range of test cases.
07: Intellectual Property and Images	M-AP-08	Incorporate existing code, media, and libraries into original programs, and give attribution.	M-IC-04	Discuss the benefits and consequences of making information either public or private.
08: Clean Code and Debugging	M-AP-10	Document programs in order to make them easier to follow, test, and debug.		
09: Project - Multi-Page Websites	M-AP-09	Systematically test and refine programs using a range of test cases.	M-AP-08	Incorporate existing code, media, and libraries into original programs, and give attribution.

Lesson	Identifier1	Standard1	Identifier2	Standard2
Chapter 2: Styling and CSS				
10: Styling Text with CSS	M-AP-08	Incorporate existing code, media, and libraries into original programs, and give attribution.	M-AP-10	Document programs in order to make them easier to follow, test, and debug.
11: Styling Elements with CSS	M-AP-08	Incorporate existing code, media, and libraries into original programs, and give attribution.	M-AP-10	Document programs in order to make them easier to follow, test, and debug.
12: Sources and Search Engines	M-IC-02	Compare the positive & negative effects of computing technologies on society.		
13: RGB Colors and Classes	M-AP-08	Incorporate existing code, media, and libraries into original programs, and give attribution.	M-AP-10	Document programs in order to make them easier to follow, test, and debug.
14: Project - Final Personal Website	M-AP-03	Seek and incorporate feedback from team members and users to refine a solution that meets user needs.	M-AP-01	Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.

Unit 3: Interactive Animations and Games

Lesson	Identifier1	Standard1	Identifier2	Standard2
Chapter 1: Images and Animations				
01: Programming for Entertainment	M-IC-01	Discuss issues of bias and accessibility in existing technologies.	M-IC-02	Compare the positive & negative effects of computing technologies on society.
02: Plotting Shapes	M-AP-02	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	M-AP-04	Create flowcharts and/or pseudocode to address complex problems as algorithms.
03: Drawing in Game Lab	M-NI-02	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	M-AP-10	Document programs in order to make them easier to follow, test, and debug.

Lesson	Identifier1	Standard1	Identifier2	Standard2
04: Shapes and Randomization	M-AP-09	Systematically test and refine programs using a range of test cases.	M-AP-02	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
05: Variables	M-AP-05	Create clearly named variables that represent different data types and perform operations on their values.	M-AP-09	Systematically test and refine programs using a range of test cases.
06: Sprites	M-AP-08	Incorporate existing code, media, and libraries into original programs, and give attribution.	M-AP-02	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
07: The Draw Loop	M-AP-06	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	M-AP-11	Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries.
08: Counter Pattern Unplugged	M-AP-04	Create flowcharts and/or pseudocode to address complex problems as algorithms.	M-AP-05	Create clearly named variables that represent different data types and perform operations on their values.
09: Sprite Movement	M-AP-06	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	M-AP-10	Document programs in order to make them easier to follow, test, and debug.
10: Booleans Unplugged	M-AP-04	Create flowcharts and/or pseudocode to address complex problems as algorithms.		
11: Booleans and Conditionals	M-AP-06	Create procedures with parameters to organize code and make it easier to reuse.	M-AP-09	Systematically test and refine programs using a range of test cases.
12: Conditionals and User Input	M-AP-08	Incorporate existing code, media, and libraries into original programs, and give attribution.	M-AP-07	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
13: Other Forms of Input	M-AP-02	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	M-AP-08	Incorporate existing code, media, and libraries into original programs, and give attribution.
14: Project - Interactive Card	M-AP-01	Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.	M-AP-12	Develop a process creating a computational artifact that leads to a minimum viable product followed by reflection, analysis, and iteration.

Lesson	Identifier1	Standard1	Identifier2	Standard2
Chapter 2: Building Games				
15: Velocity	M-DA-02	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	M-AP-05	Create clearly named variables that represent different data types and perform operations on their values.
16: Collision Detection	M-AP-06	Create procedures with parameters to organize code and make it easier to reuse.	M-AP-08	Incorporate existing code, media, and libraries into original programs, and give attribution.
17: Complex Sprite Movement	M-AP-05	Create clearly named variables that represent different data types and perform operations on their values.	M-AP-07	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
18: Collisions	M-AP-07	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	M-AP-09	Systematically test and refine programs using a range of test cases.
19: Functions	M-AP-06	Create procedures with parameters to organize code and make it easier to reuse.	M-AP-10	Document programs in order to make them easier to follow, test, and debug.
20: The Game Design Process	M-AP-05	Create clearly named variables that represent different data types and perform operations on their values.	M-AP-09	Systematically test and refine programs using a range of test cases.
21: Using the Game Design Process	M-AP-02	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	M-AP-07	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
22: Project - Design a Game	M-AP-01	Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.	M-AP-03	Seek and incorporate feedback from team members and users to refine a solution that meets user needs.

Unit 4: The Design Process

Lesson	Identifier1	Standard1	Identifier2	Standard2
Chapter 1: User Centered Design				
01: Analysis of Design	M-CS-01	Recommend improvements to the design of computing devices based on an analysis of how users interact with the devices.		
02: Understanding Your User	M-CS-01	Recommend improvements to the design of computing devices based on an analysis of how users interact with the devices.		
03: User-Centered Design Micro Activity	M-CS-02	Design projects that combine hardware and software components to collect and exchange data.	M-IC-02	Compare the positive & negative effects of computing technologies on society.
04: User Interfaces	M-AP-04	Create flowcharts and/or pseudocode to address complex problems as algorithms.	M-CS-01	Recommend improvements to the design of computing devices based on an analysis of how users interact with the devices.
05: Feedback and Testing	M-AP-04	Create flowcharts and/or pseudocode to address complex problems as algorithms.	M-AP-03	Seek and incorporate feedback from team members and users to refine a solution that meets user needs.
06: Identifying User Needs	M-AP-04	Create flowcharts and/or pseudocode to address complex problems as algorithms.	M-AP-03	Seek and incorporate feedback from team members and users to refine a solution that meets user needs.
07: Project - Paper Prototype	M-AP-04	Create flowcharts and/or pseudocode to address complex problems as algorithms.	M-AP-03	Seek and incorporate feedback from team members and users to refine a solution that meets user needs.
Chapter 2: App Prototyping				
08: Designing Apps for Good	M-IC-02	Compare the positive & negative effects of computing technologies on society.	M-IC-01	Discuss issues of bias and accessibility in existing technologies.
09: Market Research	M-IC-02	Compare the positive & negative effects of computing technologies on society.	M-IC-01	Discuss issues of bias and accessibility in existing technologies.

Lesson	Identifier1	Standard1	Identifier2	Standard2
10: Paper Prototypes	M-AP-04	Create flowcharts and/or pseudocode to address complex problems as algorithms.	M-AP-02	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
11: Prototype Testing	M-AP-04	Create flowcharts and/or pseudocode to address complex problems as algorithms.	M-AP-03	Seek and incorporate feedback from team members and users to refine a solution that meets user needs.
12: Digital Design	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.
13: Linking Screens	M-AP-02	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	M-AP-06	Create procedures with parameters to organize code and make it easier to reuse.
14: Testing the App	M-AP-03	Seek and incorporate feedback from team members and users to refine a solution that meets user needs.	M-AP-09	Systematically test and refine programs using a range of test cases.
15: Improving and Iterating	M-AP-02	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.		
16: Project - App Presentation	M-CS-01	Recommend improvements to the design of computing devices based on an analysis of how users interact with the devices.		

Unit 5: Data and Society

Lesson	Identifier1	Standard1	Identifier2	Standard2
Chapter 1: Representing Information				
01: Representation Matters	M-DA-01	Store data using multiple encoding methods.		

Lesson	Identifier1	Standard1	Identifier2	Standard2
02: Patterns and Representation	M-DA-01	Store data using multiple encoding methods.		
03: ASCII and Binary Representation	M-DA-01	Store data using multiple encoding methods.		
04: Representing Images	M-DA-01	Store data using multiple encoding methods.		
05: Representing Numbers	M-DA-01	Store data using multiple encoding methods.		
06: Keeping Data Secret	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.
07: Combining Representations	M-DA-01	Store data using multiple encoding methods.		
08: Create a Representation	M-AP-04	Create flowcharts and/or pseudocode to address complex problems as algorithms.	M-AP-02	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
Chapter 2: Solving Data Problems				
09: Problem Solving and Data	M-DA-02	Collect data using computational tools and transform the data to make it more useful and reliable.		
10: Problem Solving with Big Data	M-IC-02	Compare the positive & negative effects of computing technologies on society.	M-IC-04	Discuss the benefits and consequences of making information either public or private.
11: Structuring Data	M-DA-02	Collect data using computational tools and transform the data to make it more useful and reliable.		
12: Making Decisions with Data	M-DA-02	Collect data using computational tools and transform the data to make it more useful and reliable.		

Lesson	Identifier1	Standard1	Identifier2	Standard2
13: Interpreting Data	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.
14: Automating Data Decision	M-AP-04	Create flowcharts and/or pseudocode to address complex problems as algorithms.	M-DA-02	Collect data using computational tools and transform the data to make it more useful and reliable.
15: Project - Make a Recommendation	M-AP-04	Create flowcharts and/or pseudocode to address complex problems as algorithms.	M-AP-03	Seek and incorporate feedback from team members and users to refine a solution that meets user needs.

Unit 6: Physical Computing

Lesson	Identifier1	Standard1	Identifier2	Standard2
Chapter 1: Programming with Hardware				
01: Innovations in Computing	M-IC-02	Compare the positive & negative effects of computing technologies on society.		
02: Designing Screens with Code	M-AP-05	Create clearly named variables that represent different data types and perform operations on their values.	M-AP-10	Document programs in order to make them easier to follow, test, and debug.
03: The Circuit Playground	M-AP-05	Create clearly named variables that represent different data types and perform operations on their values.	M-AP-02	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
04: Input Unplugged	M-AP-04	Create flowcharts and/or pseudocode to address complex problems as algorithms.		
05: Board Events	M-AP-05	Create clearly named variables that represent different data types and perform operations on their values.	M-CS-01	Recommend improvements to the design of computing devices based on an analysis of how users interact with the devices.
06: Getting Properties	M-AP-05	Create clearly named variables that represent different data types and perform operations on their values.		

Lesson	Identifier1	Standard1	Identifier2	Standard2
07: Analog Input	M-AP-05	Create clearly named variables that represent different data types and perform operations on their values.	M-CS-01	Recommend improvements to the design of computing devices based on an analysis of how users interact with the devices.
08: The Program Design Process	M-AP-06	Create procedures with parameters to organize code and make it easier to reuse.	M-CS-03	Identify and fix problems with computing devices and their components systematically.
09: Project: Make a Game	M-CS-03	Identify and fix problems with computing devices and their components systematically.	M-AP-12	Develop a process creating a computational artifact that leads to a minimum viable product followed by reflection, analysis, and iteration.
Chapter 2: Building Physical Prototypes				
10: Arrays and Color LEDs	M-CS-02	Design projects that combine hardware and software components to collect and exchange data.	M-AP-05	Create clearly named variables that represent different data types and perform operations on their values.
11: Making Music	M-AP-07	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	M-CS-02	Design projects that combine hardware and software components to collect and exchange data.
12: Arrays and For Loops	M-CS-01	Recommend improvements to the design of computing devices based on an analysis of how users interact with the devices.	M-AP-06	Create procedures with parameters to organize code and make it easier to reuse.
13: Accelerometer	M-CS-02	Design projects that combine hardware and software components to collect and exchange data.		
14: Functions with Parameters	M-AP-05	Create clearly named variables that represent different data types and perform operations on their values.	M-AP-06	Create procedures with parameters to organize code and make it easier to reuse.
15: Circuits and Physical Prototypes	M-IC-01	Discuss issues of bias and accessibility in existing technologies.	M-CS-02	Design projects that combine hardware and software components to collect and exchange data.
16: Project: Prototype an Innovation	M-AP-01	Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.	M-CS-01	Recommend improvements to the design of computing devices based on an analysis of how users interact with the devices.

