

Skill Struck's alignment to

Kentucky Academic Standards for Computer Science




Legend



✓ = Standard aligned



♦ = Not currently aligned



⚠ = Partially aligned (parts of the standard we align with are highlighted in yellow)



Standard	Status
Elementary (K-5)	
E-NI-01 Understand the basic components of how networks operate to protect physical and digital information. Subconcept: Network Communication & Organization <u>Grade-by-Grade Indicators:</u> K - Describe how usernames and passwords protect personal information. 1 - Demonstrate how to log in and log out of digital device using age appropriate usernames and passwords. 2 - Describe the characteristics of a strong password. 3 - Explain the effects of password misuse. 4 - Explain how acceptable use policies help protect physical devices and digital information. 5 - Demonstrate an understanding of digital security (i.e. use strong passwords; use usernames; protect personal digital information)	<div>✓</div>



<p>E-NI-02</p> <p>Model how information is broken down into smaller pieces (data packets), transmitted over various paths (physical and/or wireless), and reassembled at the destination</p> <p>Subconcept: Cybersecurity</p> <p><u>Grade-by-Grade Indicators:</u></p> <p>K - Not introduced until 4th grade</p> <p>1 - Not introduced until 4th grade</p> <p>2 - Not introduced until 4th grade</p> <p>3 - Not introduced until 4th grade</p> <p>4 - Describe how computers break down information.</p> <p>5 - Use a model to represent how digital information is sent and received over physical or wireless paths.</p>	
<p>E-DA-01</p> <p>Appropriately store and modify digital files.</p> <p>Subconcept: Storage</p> <p><u>Grade-by-Grade Indicators:</u></p> <p>K - Open and close digital files with prompting and support.</p> <p>1 - Open, close and save digital files with prompting and support.</p> <p>2 - Open, close and save digital files.</p> <p>3 - Search, modify and delete digital files with prompting and support.</p> <p>4 - Search, modify, and delete digital files.</p> <p>5 - Convert digital files.</p>	
<p>E-DA-02</p> <p>Collect and visually display data using appropriate applications.</p> <p>Subconcept: Collection, Visualization & Transformation</p>	



<p><u>Grade-by-Grade Indicators:</u></p> <p>K – Not introduced until 1st Grade.</p> <p>1 – Ask questions to collect and visually represent data with prompting and support.</p> <p>2 – Collect and visually represent data using one digital format with prompting and support.</p> <p>3 – Collect and visually represent data in tables or graphical displays using one application or digital format.</p> <p>4 – Collect data and determine an appropriate application or format to visually display the data.</p> <p>5 – Collect and represent data in graphical displays using one or more application to determine the benefits of using more than one visual display type.</p>	
<p>E-DA-03</p> <p>Analyze data for trends and relationships</p> <p>Subconcept: Inference & Models</p> <p><u>Grade-by-Grade Indicators:</u></p> <p>K – Use observations to describe patterns in organized data with prompting and support.</p> <p>1 – Use observations to describe patterns in organized data.</p> <p>2 – Use observations to describe patterns that can be predicted in organized data.</p> <p>3 – Analyze and interpret data using digital tools.</p> <p>4 – Analyze and interpret data to describe patterns using digital tools.</p> <p>5 – Represent data in graphical displays and describe cause and effect relationships, communicate ideas or predict outcomes.</p>	
<p>E-AP-01</p> <p>Create, follow, compare and refine algorithms for a task.</p> <p>Subconcept: Algorithms</p>	



<p><u>Grade-by-Grade Indicators:</u></p> <p>K – Use simple algorithms to complete everyday tasks.</p> <p>1 – Create and use simple algorithms to complete everyday tasks.</p> <p>2 – Create and use simple algorithms using images, text or visual programming blocks to complete everyday tasks.</p> <p>3 – Compare two sets of algorithms for the same task to determine the best solution.</p> <p>4 – Create and compare two sets of algorithms for the same task to determine the best solution.</p> <p>5 – Modify a set of algorithms and discuss how multiple paths can lead to the same solution.</p>	
<p>E-AP-02</p> <p>Explore and use variables in a program.</p> <p>Subconcept: Variables</p> <p><u>Grade-by-Grade Indicators:</u></p> <p>K – Describe ways people represent data.</p> <p>1 – Explain how numbers are used to represent data.</p> <p>2 – Create a simple model to show how a computer stores information using numbers or symbols.</p> <p>3 – Identify ways variables are used in programs.</p> <p>4 – Modify or remix an existing program that uses a variable.</p> <p>5 – Create a program that uses a variable.</p>	
<p>E-AP-03</p> <p>Routinely create programs using a variety of tools to express ideas, address a problem or create an artifact, individually and collaboratively.</p> <p>Subconcept: Control</p> <p><u>Grade-by-Grade Indicators:</u></p>	



<p>K - Routinely create simple programs, independently OR collaboratively.</p> <p>1 - Routinely create simple programs, independently AND collaboratively.</p> <p>2 - Routinely create simple programs with sequences using a variety of tools, independently and collaboratively.</p> <p>3 - Routinely create simple programs with sequences or events using a variety of tools, independently and collaboratively.</p> <p>4 - Routinely create simple programs with sequences, events or loops routinely using a variety of tools, independently and collaboratively.</p> <p>5 - Routinely create simple programs with sequences, events, loops, variables or conditionals routinely using a variety of tools, independently and collaboratively.</p>	
<p>E-AP-04</p> <p>Decompose precise steps needed to solve a problem.</p> <p>Subconcept: Modularity</p> <p>Grade-by-Grade Indicators:</p> <p>K - Generate the steps needed to solve a simple problem with prompting and support.</p> <p>1 - Generate the steps needed to solve a simple problem.</p> <p>2 - Generate and correctly order the steps needed to solve a simple problem.</p> <p>3 - Generate and correctly order the steps needed to solve a complex problem.</p> <p>4 - Decompose a problem into parts to facilitate program development.</p> <p>5 - Decompose a problem into parts and subparts to facilitate program development.</p>	
<p>E-AP-05</p> <p>Use a process when creating programs or computational artifacts.</p> <p>Subconcept: Modularity</p>	

<p><u>Grade-by-Grade Indicators:</u></p> <p>K – Use a process when creating simple programs, individually OR collaboratively, with prompting and support.</p> <p>1 – Use a process to create simple programs, individually AND collaboratively, with prompting and support.</p> <p>2 – Use a process to create simple programs that include sequences.</p> <p>3 – Use a process to create programs that include sequences and events.</p> <p>4 – Use a process to create programs that includes loops, sequences or events.</p> <p>5 – Use a process to create programs that include loops, sequences, events, variables or conditions.</p>	
<p>E-AP-06</p> <p>Modify, remix or reuse part of an existing program to create a new program, giving attribution to others.</p> <p>Subconcept: Program Development</p> <p><u>Grade-by-Grade Indicators:</u></p> <p>K – Not introduced until 3rd grade</p> <p>1 – Not introduced until 3rd grade</p> <p>2 – Not introduced until 3rd grade</p> <p>3 – Modify or add features to an existing program, with prompting and support, to create a new program, giving attribution.</p> <p>4 – Modify, remix or reuse parts of an existing program to create a new program, giving attribution.</p> <p>5 – Modify, remix, reuse parts or add features to an existing program to create a new program, giving attribution.</p>	
<p>E-AP-07</p> <p>Document, share and reflect when creating programs using correct terminology.</p>	

<p>Subconcept: Program Development</p> <p><u>Grade-by-Grade Indicators:</u></p> <p>K - Document simple programs, using pictures, in order to share process with others.</p> <p>1 - Document simple programs, using pictures, in order to share with others and reflect on the process.</p> <p>2 - Document simple programs, with pictures and/or text, to share with others and reflect on the process.</p> <p>3 - Document programs and discuss development process with peers.</p> <p>4 - Document programs and discuss development process with peers, using correct terminology.</p> <p>5 - Document programs using correct terminology and incorporate peer feedback in the development process.</p>	
<p>E-AP-08</p> <p>Identify and correct errors in an algorithm.</p> <p>Subconcept: Program Development</p> <p><u>Grade-by-Grade Indicators:</u></p> <p>K - Analyze and debug simple algorithms with prompting and support.</p> <p>1 - Analyze and debug simple algorithms which includes sequencing.</p> <p>2 - Analyze and debug algorithms which includes simple loops.</p> <p>3 - Analyze and debug algorithms which includes sequencing and loops.</p> <p>4 - Analyze and debug algorithms which includes sequencing, loops and events.</p> <p>5 - Analyze and debug algorithms which includes sequencing, loops, events and conditionals.</p>	
<p>E-IC-01</p> <p>Discuss how computing has impacted society.</p> <p>Subconcept: Culture</p>	

<p><u>Grade-by-Grade Indicators:</u></p> <p>K – Make observations to describe ways computing devices are used daily life.</p> <p>1 – Describe computing devices used in different careers.</p> <p>2 – Demonstrate how some tasks can be completed with or without a computing device.</p> <p>3 – Describe how computing technology impacts the way people live, work, and interact.</p> <p>4 – Compare and contrast how computing has changed society from the past to the present.</p> <p>5 – Describe the positive and negative impacts of computing on society.</p>	
<p>E-IC-02</p> <p>Discover how computing devices have affected the way people communicate.</p> <p>Subconcept: Social Interactions</p> <p><u>Grade-by-Grade Indicators:</u></p> <p>K – Describe different computing devices used for communication.</p> <p>1 – Describe ways people can communicate using computing devices.</p> <p>2 – Compare similarities and differences between in person and online communications.</p> <p>3 – Describe ways in which computing devices could be made more accessible to all users.</p> <p>4 – Use online collaborative spaces ethically and safely to work with other students to solve a problem or reach a goal.</p> <p>5 – Compare diverse perspectives, synchronously or asynchronously, to improve a project.</p>	
<p>E-IC-03</p> <p>Evaluate the relevance and appropriateness of electronic information sources and digital media.</p>	

<p>Subconcept: Safety, Law & Ethics</p> <p><u>Grade-by-Grade Indicators:</u></p> <p>K - Describe characteristics of a website, with prompting and support.</p> <p>1 - Describe the purpose of different websites, with prompting and support.</p> <p>2 - Use and cite sources from approved digital materials.</p> <p>3 - Describe the relevance and appropriateness of various electronic information sources and digital media.</p> <p>4 - Compare the relevance and appropriateness of various electronic information sources and digital media.</p> <p>5 - Use relevant and appropriate electronic information sources and digital media, citing resources, for various tasks.</p>	
<p>E-IC-04</p> <p>Understand the importance of proper use of data and information in a computing society.</p> <p>Subconcept: Safety, Law & Ethics</p> <p><u>Grade-by-Grade Indicators:</u></p> <p>K - Describe characteristics of private information.</p> <p>1 - Identify harmful behaviors when using a connected device.</p> <p>2 - Demonstrate appropriate behavior when sending messages online.</p> <p>3 - Describe positive qualities of a digital citizen.</p> <p>4 - Describe potential strategies to manage and eliminate cyberbullying.</p> <p>5 - Understand consequences for sending or receiving inappropriate content.</p>	
<p>E-CS-01</p> <p>Identify, select and operate appropriate software and hardware to perform a variety of tasks and recognize that users have different needs and preferences for the technology they use.</p>	

<p>Subconcept: Devices</p> <p><u>Grade-by-Grade Indicators:</u></p> <p>K – Describe ways people use digital devices to perform tasks.</p> <p>1 – Use the appropriate device and application or software to complete a given task, with prompting and support.</p> <p>2 – Describe and use the appropriate device and application or software to complete a given task.</p> <p>3 – Compare and contrast various types and functions of software or applications.</p> <p>4 – Describe the capabilities and limitations of various software and applications for a particular use.</p> <p>5 – Justify selection of a particular computing device based on a desired application or task.</p>	
<p>E-CS-02</p> <p>Identify and describe the function of common physical components of computing systems (hardware) using appropriate terminology.</p> <p>Subconcept: Hardware & Software</p> <p><u>Grade-by-Grade Indicators:</u></p> <p>K – Use appropriate terminology to identify basic hardware.</p> <p>1 – Use appropriate terminology to identify basic software.</p> <p>2 – Describe the function of common hardware and software.</p> <p>3 – Compare and contrast features of different digital devices.</p> <p>4 – Describe the capabilities and limitations of various digital devices.</p> <p>5 – Describe the function of major hardware components of a digital device.</p>	
<p>E-CS-03</p> <p>Describe basic hardware and software problems using accurate terminology.</p> <p>Subconcept: Troubleshooting</p>	

<p><u>Grade-by-Grade Indicators:</u></p> <p>K – Identify a simple hardware problem.</p> <p>1 – Describe simple hardware and software problems.</p> <p>2 – Use observations to distinguish between simple hardware and software problems.</p> <p>3 – Demonstrate common troubleshooting strategies to solve simple hardware and software problems.</p> <p>4 – Describe the causes of hardware, software and connectivity problems.</p> <p>5 – Demonstrate an appropriate response to various error messages and identify the component and/or application causing the error.</p>	
Middle School (6–8)	
<p>M-NI-01</p> <p>Model how different sets of rules (protocols) are used to transmit different types of data across networks and the Internet.</p>	✓
<p>M-NI-02</p> <p>Model how information is disguised using different methods of encryption to secure it during transmission from one point to another.</p>	✓
<p>M-NI-03</p> <p>Explain how physical and digital security practices and measures proactively address the threat of breaches to personal and private data.</p>	✓
<p>M-DA-01</p> <p>Store data using multiple encoding methods.</p>	✓
<p>M-DA-02</p> <p>Collect data using computational tools and transform the data to make it more useful and reliable.</p>	✓
<p>M-DA-3</p> <p>Refine computational models based on the data they have generated.</p>	✓

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.	✓
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.	✓
M-AP-07 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	✓
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.	✓
M-AP-10	✓

Document programs in order to make them easier to follow, test, and debug.	
M-AP-11 Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries.	✓
M-AP-12 Discuss issues of bias and accessibility in existing technologies.	✓
M-IC-01 Select and operate commonly used devices to perform a variety of tasks.	✓
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.	✓
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.	✓

High School (9-12)	
H-NI-01 Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, end devices, topology, and addressing.	✓
H-NI-02 Give examples to illustrate how sensitive data can be affected by viruses, malware and other attacks.	✓
H-NI-03 Recommend security measures to address various scenarios based on factors such as usability, efficiency, feasibility, and ethical impacts.	✓
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.	✓
H-DA-01 Evaluate the tradeoffs in how data elements are organized and where data is stored.	✓
H-DA-02 Collect data using appropriate data collection tools and techniques to support a claim or to communicate information.	✓
H-DA-03 Understand and design database structures to optimize search and retrieval.	♦
H-DA-04	✓

Explain the privacy concerns related to the collection and generation of data.	
H-DA-05 Use data analysis tools and techniques to identify patterns and analyze data represented in complex systems.	✓
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.	✓
H-DA-09 Evaluate the ability of models and simulations to test and support the refinement of hypotheses.	✓
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.	✓

H-AP-04 Design and iteratively develop event-driven computational artifacts for practical intent, personal expression, or to address a societal issue.	✓
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.	✓
H-AP-08 Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.	✓
H-AP-09 Evaluate and refine computational artifacts to make them more usable and accessible using systematic testing and debugging.	✓
H-AP-10 Systematically design and develop programs for broad audiences by incorporating feedback from users.	✓
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.	✓
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.	✓
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.	✓
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).	✓
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.	✓
H-IC-01 Reduce bias and equity deficits through the design of accessible computational artifacts.	✓
H-IC-02 Evaluate and assess how 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 laws governing data (intellectual property, privacy etc.) can have on innovation.	✓
H-IC-05 Evaluate and design computational artifacts to maximize their benefit to society.	✓
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-07 Demonstrate ways computational design (i.e. algorithms, abstractions and analysis) can apply to problems across disciplines.*	✓

H-IC-08 Debate laws and regulations that impact the development and use of software and the protection of privacy.	✓
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.	✓
H-CS-04 Categorize the roles of operating system software.	✓
H-CS-05 Illustrate ways computing systems implement logic, input, and output through hardware components.	✓