4th Grade CS Lessons

By the end of fourth grade, the computer literate student refines their skills as they construct programs and use algorithms to accomplish a task. Analyzing a variety of hardware and software tools, students further develop their computational thinking and problem solving skills. Working independently and collaboratively students decompose larger problems into smaller tasks. Students understand that responsible computing use includes protecting personal information and respecting the rights of others.

Completing the Course in Code.org will fulfill all of the CS standards!

COMPUTING SYSTEMS

4.CS.D.1 Computing Systems Devices	**With teacher guidance, model how internal and external parts of computing connect multiple devices in a computing system**. Computing devices may be connected to other devices or components to extend their capabilities, such as sensing and sending information. Connections can take many forms, such as physical or wireless. Together, devices and components form a system of interdependent parts that interact for a common purpose. Students model the process that happens when multiple devices form a system *Practice(s): Communicating About Computing, Recognizing and Defining Computational Problems, Creating Computational Artifacts: 7.3, 3.1, 5.2*
You're already doing it!!	A computing system is everything that connects to the computer - printer, mouse, keyboard, projector, etc. They are connected through wires, wirelessly or through the network. Discussing and demonstrating these devices is modeling this process!
Extensions	Tutorial - Inside a Computer Tutorial - Basic parts of a Computer If you have access to an old computer that is no longer used, take it apart and show them the inside parts! Students LOVE this!!

4.CS.HS.1 Computer Systems Hardware & Software	**Recognize that bits serve as the basic unit of data in computing systems and can represent a variety of information.** Hardware and software communicate in binary digits commonly represented in 0s and 1s. Students discuss how bits are a unit of data. *Practice(s): Communicating About Computing: 7.2*
You're already doing it!!	0s and 1s relate to on and off, checked or unchecked and many other math concepts. Incorporating a discussion of binary could be used as an extension to many mathematical concepts!
Extensions	Video Binary Explained - Khan Academy Binary Bracelets activity - Code.org Binary Numbers Activities - CSUnplugged (great math extension!) TONS of Binary resources - CSunplugged

4.CS.HS.2 Computer Systems Hardware :& Software	**Recognize that a single piece of hardware can accomplish different tasks depending on its software**. A photo filter application (software) works with a camera (hardware) to produce a variety of effects that change the appearance of an image. This image is transmitted and stored as bits, or binary digits, which are commonly represented as 0s and 1s. All information, including instructions, is encoded as bits. Students discuss a variety of software and hardware that work together. *Practice(s): Communicating About Computing: 7.2*
You're already doing it!!	Students are using their Chromebooks (hardware) to do a number of different tasks A smartphone (hardware) can be used for much more than a phone based on the apps (software)
Extensions	Video about Hardware and Software Series of How Computers Work - Khan Academy

4.CS.T.1 Computer Systems Troubleshoot	**Develop and apply simple troubleshooting strategies to solve simple hardware and software problems.** Although computing systems may vary, common troubleshooting strategies such as checking connections and power, or swapping a working part in place of a potentially defective part, can be used to restore functionality. Restarting a device (rebooting) is commonly effective because it resets the machine. Because computing devices are composed of an interconnected system of hardware and software, troubleshooting strategies may need to address both. *Practice(s): Recognizing and Defining Computational Problems, Collaborating Around Computing: 3.1, 2.4*
You're already doing it!!	Teaching students basic troubleshooting in the beginning of the year When they have issues asking them what they've already tried to resolve the issue Teach students to SHUT DOWN after every use! (not just log off) If the computer is "frozen" have them restart Have students make sure to charge the computers. When their computer isn't working have THEM be specific in identifying the problem - "my monitor is black" and a possible solution "It needs to be plugged in". "I can't log in" - "Is the caps lock on? Do you have a WiFi connection?"
Extensions	Create a classroom job for Student Onsite Support (Tech Expert). Review the Student Onsite Support document with the class in the beginning of the year so students are aware of basic troubleshooting tips.

NETWORKS & THE INTERNET

4.NI.C.1 Networks &	**Discuss real-world cybersecurity problems and how personal information can be protected.** Just as we protect our personal property online, we also need to protect our

Internet Cybersecurity	devices and the information stored on them. Information can be protected using various security measures. These measures can be physical and/or digital. For example, discussion topics could be based on current events related to cybersecurity or topics that are applicable to students and the programs/devices they use. *Practice(s): Communicating about Computing, Recognizing and Defining Computational Problems: 7.2, 3.3*
You're already doing it!!	These are discussions that can take place normally while using technology in class. ● Reminding students not to click on pop-ups ● Not posting personal information in pubic forums ● Discussing with students why they shouldn't click on things they don't know, visit websites they aren't directed to, open emails and documents that may contain viruses. ● Discussing the reasons students are unable to download things onto their computers
Extensions	Private and Personal Information Lesson Plan - Common Sense Media Cybersecurity Lesson Plan - Khan Academy (with videos) EXCELLENT Cybersecurity lab by Nova (lesson plans, materials and activities!) Reading materials about CyberSecurity or Order Free copies! Safe Online Surfing Interactive provided by FBI

4.NI. NCO.1 Networks & Internet Network, Communication Organization	**Model how information is decomposed, transmitted as packets through multiple devices over networks and reassembled at the destination**. There are physical paths for communicating information, such as Ethernet cables, and wireless paths, such as Wi-Fi. Often, information travels on a combination of physical and wireless paths. Information is broken down into smaller pieces called packets, which are sent over the network and reassembled at the destination. Routers and switches are used to properly send packets across paths to their destinations. *Practice(s): Developing and Using Abstractions: 4.4*
You're already doing it!!	Discussing connectivity when connecting to WiFi or the WiDi projector Demonstrating how information travels while playing "telephone" Comparing information flow to flow through circuits (science)
Extensions	Packets routers and Reliability Video - Khan Academy

DATA & ANALYSIS

4.DA. CVT.1 Data & Analysis Collection Visualization	**Select tools to collect, organize, and present data visually to highlight relationships and support a claim**. Tools are chosen based upon the type of measurement they use as well as the type of data people wish to observe. Organizing data can make interpreting and communicating it to others easier. *Practice(s): Developing and Using Abstractions, Creating Computational Artifacts: 4.1, 5.1*

Transportation	
You're already doing it!!	Providing different options for data collection - paper/pencil, Google sheet, Whiteboard, easel paper, etc. - and allowing them to choose which would be best for the task at hand Collect, organize and present data lessons in other subjects (math, science, social studies, etc.)
Extensions	Extend your regular data collection (science, ss, etc) lessons to include incorporate the use of technology (collect - Google Forms, Organize - Sheets, Present - Slides) Create infographics of collected data using Canva

4.DA.S.1 Data & Analysis Storage	**Recognize different file extensions and the different amounts of storage required for each type.** Music, images, video, and text require different amounts of storage. Video will often require more storage than music or images alone because video combines both. Students discuss common file extensions, such as .doc, .pdf, and .jpeg and give examples of files that require different amounts of storage. *Practice(s): Communicating About Computing: 7.2*
You're already doing it!!	Identify file extensions when opening different file types Identify which software programs open extensions Categorize different file types (image, video, document) Discussion of file size when emailing files
Extensions	Use different file types in place of a spelling/vocab

4.DA.IM. 1 Data & Analysis Inference & Models	**Use a computational tool to manipulate data to draw conclusions, make predictions, and answer questions.** People use data to highlight or propose cause-and-effect relationships and predict outcomes. Basing inferences or predictions on data does not guarantee their accuracy; the data must be relevant and of sufficient quantity. *Practice(s): Communicating about Computing, Creating Computational Artifacts, Collaborate around Computing: 7.2, 5.2, 2.4*
You're already doing it!!	Graphing in math - "based on the graph what do you think…." Inferences in Language Arts while reading Collecting weather data and predicting forecast in Science
Extensions	Answering questions from data lesson plan - Education.com Entering ANY data collection into a Google sheet and create charts Collect data using google forms

ALGORITHMS & PROGRAMMING

4.AP.A.1 Algorithms & Programming Storage	**Compare and refine multiple algorithms for the same task and determine which is the most effective**. Different algorithms can achieve the same result, though sometimes one algorithm might be most appropriate for a specific situation. Students should be able to look at different ways to solve the same task and decide which would be the best solution. For example, students might compare algorithms that describe how to get ready for school or how to tie their shoes. Students could use a map and plan multiple algorithms to get from one point to another. They could look at routes suggested by mapping software and change the route to something that would be better, based on which route is the shortest or fastest or would avoid a problem. Another example might be to write different algorithms to draw a regular polygon and determine which algorithm would be the easiest to modify or repurpose to draw a different polygon. *Practice(s): Testing and Refining Computational Artifacts, Recognizing and Defining Computational Problems: 6.3*
You're already doing it!!	Discussing how students can get the same answer using different processes Discussing step by step instructions and which is the best solution Identifying the best solution from multiple correct solutions - what is "best" based on? Quickest? Most thorough? Efficiency? Cost effective, etc.
Extensions	TONS of unplugged activities for Algorithms from code.org Explain Algorithms to kids by Tynker

4.AP.V.1 Algorithms & Programming Variables	**Create programs that use variables to store and modify data Variables are used to store and modify data**. At this level, understanding how to use variables is sufficient, without a fuller understanding of the technical aspects of variables (such as identifiers and memory locations. Students may use mathematical operations to add to the score of a game or subtract from the number of lives in a game. Programs can imply either digital or paperbased designs. *Practice(s): Creating Computational Artifacts: 5.2*
You're already doing it!!	Using variables in Math Video about Math Variables
Extensions	Variables from code.org Unplugged Activity for Variables from code.org

4.AP.C.1 Algorithms & Programming Controls	**Create programs that include sequences, events, loops, and/or conditionals.** Control structures specify the order (sequence) in which instructions are executed within a program and can be combined to support the creation of more complex programs. If dialogue is not sequenced correctly when programming a simple animated story, the story will not make sense. Events allow portions of a program to run based on a specific action. Students could write a program to explain the water cycle and when a specific component

	is clicked (event), the program would show information about that part of the water cycle. Loops allow for the repetition of a sequence of code multiple times. In a program that produces an animation about a famous historical character, students could use a loop to have the character walk across the screen as they introduce themselves. Conditionals allow for the execution of a portion of code in a program when a certain condition is true. Students could write a math game that asks multiplication fact questions and then uses a conditional to check whether or not the answer that was entered is correct. Practice(s): Creating Computational Artifacts: 5.2
You're already doing it!!	Sequencing, placing things in order Cause and Effect
Extensions	Tynker - hour of code activities - 1 hour self guided lessons with complete instruction! Including videos, etc. Code.org - hour of code activities - 1 hour self guided lessons If-Then Adventure Stories in Google Slides Unplugged Conditional Activity - Code.org

4.AP.M.1 Algorithms & Programming Modularity	**Decompose problems into smaller, manageable subproblems to facilitate the program development process.** Decomposition is the act of breaking down a task into multiple simpler tasks. Decomposition also enables different people to work on different parts at the same time. For example, students could create an animation by separating a story into different scenes. For each scene, they would select a background, place characters, and program actions. *Practice(s): Recognizing and Defining Computational Problems: 3.2*
You're already doing it!!	Chunking words Deconstructing text Breaking a math problem into smaller manageable pieces Breaking a project into smaller parts
Extensions	Any collaborative project using Google apps - slides, docs, sheets, etc. or Canva or any other collaborative app where students work together on the same document Pair programming

4.AP.M.2 Algorithms & Programming Modularity	**Modify, remix, or incorporate portions of an existing program into one's own work to add more advanced features**. Programs can be broken down into smaller parts, which can be incorporated into new or existing programs. For example, students could modify prewritten code from a single-player game to create a two-player game with slightly different rules, remix and add another scene to an animated story, use code to make a ball bounce from another program in a new basketball game, or modify an image created by another student. *Practice(s): Creating Computational Artifacts: 5.3*

You're already doing it!!	Gathering research information for a report Using a template to complete a project Peer editing
Extensions	Create a game in Tynker - self guided easy lesson

4.AP.PD.1 Algorithms & Programming Program Development	**Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences**. Planning is an important part of the iterative process of program development. Students outline key features, time and resource constraints, and user expectations. Students should document the plan as, for example, a storyboard, flowchart, pseudocode, or story map. Practice(s): Fostering an Inclusive Computing Culture, Creating Computational Artifacts: 1.1, 5.1
You're already doing it!!	Planning a school project - deadlines, documentation, rough draft, etc. Pre-writing using thinking maps Working collaboratively then sharing their preferences Planning, executing, evaluating, planning for improvement Goal setting Sequencing events in a story
Extensions	Programming in code.org with angry birds! Great computational thinking game that ties with Math!

4.AP.PD.2 Algorithms & Programming Program Development	**Observe intellectual property rights and give appropriate attribution when creating or remixing programs**. Intellectual property rights can vary by country but copyright laws give the creator of a work a set of rights that prevents others from copying the work and using it in ways that the creator may not like. Students should identify instances of remixing, when ideas are borrowed and iterated upon, and credit the original creator. Students should also consider common licenses that place limitations or restrictions on the use of computational artifacts, such as images and music downloaded from the Internet. At this stage, attribution should be written in the format required by the teacher and should always be included on any programs shared online. *Practice(s): Creating Computational Artifacts, Communicating About Computing: 5.2, 7.3*
You're already doing it!!	Talking about copyright, patents, trademarks, etc. Citing sources when doing research
Extensions	Intellectual Property lesson plan Intellectual Property lesson plan #2

4.AP.PD.3 Algorithms & Programming Program Development	**Test and debug (identify and fix errors) a program/app or algorithm to ensure it runs as intended.** As students develop programs they should continuously test those programs to see that they do what was expected and fix (debug), any errors. Students should also be able to successfully debug simple errors in programs created by others. *Practice(s): Testing and Refining Computational Artifacts: 6.1, 6.2*
You're already doing it!!	Cause and effect Solving problems Doing test corrections Persevering, continuing until there is a correct solution
Extensions	Debugging - Code.org self guided lesson Debugging activities from Code for Fun GREAT debugging challenges from Scratch!

4.AP.PD.4 Algorithms & Programming Programming Development	**With teacher guidance, students take on varying roles when collaborating with peers during the design, implementation, and review stages of program development.** Collaborative computing is the process of performing a computational task by working in pairs or on teams. Because it involves asking for the contributions and feedback of others, effective collaboration can lead to better outcomes than working independently. Students should take turns in different roles during program development, such as note taker, facilitator, program tester, or "driver" of the computer. *Practice(s): Collaborating Around Computing: 2.2*
You're already doing it!!	When doing group tasks alternating jobs of the group self-Evaluating participation in group projects
Extensions	Any collaborative project using Google apps - slides, docs, sheets, etc. or Canva or any other collaborative app where students work together on the same document Pair programming - students work together to create code at code.org

4.AP.PD.5 Algorithms & Programming Programming Development	**Describe choices made during program development using code comments, presentations, and/or demonstrations**. People communicate about their code to help others understand and use their programs. Another purpose of communicating one's design choices is to show an understanding of one's work. These explanations could manifest themselves as in-line code comments for collaborators and assessors, or as part of a summative presentation, such as a code walk-through or coding journal. *Practice(s): Communicating About Computing: 7.2*
You're already doing it!!	Having students explain the process they took to solve a problem. Identifying that there are multiple ways to solve problems and have students evaluate the

	efficiency/effectiveness of their method and explain. Using comments in Google docs to correspond with collaborators Using presenter notes in Google slides Describing the process used to solve a given problem Being systematic in problem solving and describing actions taken
Extensions	Hour of Code activities by Code.org

IMPACTS OF COMPUTING

4.IC.C.1 Impacts of Computing Culture	**Identify and discuss computing technologies that have changed the world.** New computing technology is created and existing technologies are modified for many reasons, including to order to increase their benefits, decrease their risks, and meet societal needs. Students, with guidance from their teacher, should discuss topics that relate to the history of technology and the changes in the world due to technology. Students discuss how culture influences changes in technology. Topics could be based on current news content in areas, such as robotics, wireless Internet, mobile computing devices, GPS systems, wearable computing, or how social media has influenced social and political changes. *Practice(s): Recognizing and Defining Computational Problems: 3.1*
You're already doing it!!	History and social studies Changes in technology Current events
Extensions	Show examples of the evolution of different technology - phones, computers, cameras, etc. Create a timeline of the evolution of technology! Display it in your room!! - Give each student an era/date and have them identify different technologies. Their findings could be included in a collaborative slide show! (example - telephones, students each given a specific era 1920, 25, 30, 35, 40…. And do the research for the phone. Enter their findings on a specific slide to show progression) Discuss advantages and disadvantages of old and new technology Kids react to old computers video (7:41)

4.IC.C.2 Impacts of Computing Culture	**Brainstorm ways to improve the accessibility and usability of technology products for the diverse needs and wants of users**. The development and modification of computing technology are driven by people's needs and wants and can affect groups differently. Anticipating the needs and wants of diverse end users requires students to purposefully consider potential perspectives of users with different backgrounds, ability levels, points of view, and disabilities. For example, students may consider using both speech and text when they wish to convey information in a game. They may also wish to

	vary the types of programs they create, knowing that not everyone share their own tastes. *Practice(s): Fostering an Inclusive Computing Culture: 1.2*
You're already doing it!!	Brainstorming ways to problem solve technology issues in the classroom Providing requirements for projects and explaining why (neutral background, not too busy, etc.)
Extensions	Demonstrate Voice Typing option in Google Docs Show Closed Caption option in Google Slides Present mode

4.IC.SI.1 Impacts of Computing Social Interactions	**Seek opportunities for local and nationally collaboration to facilitate communication and innovation.** Computing influences many social institutions such as family, education, religion, and the economy. People can work in different places and at different times to collaborate and share ideas when they use technologies that reach across the globe. Computing provides the possibility for collaboration and sharing of ideas and allows the benefit of diverse perspectives. These social interactions affect how local and global groups interact with each other, and alternatively, these interactions can change the nature of groups. For example, a class can discuss ideas in the same class, school, or in another state or nation through interactive webinars. *Practice(s): Fostering an Inclusive Computing Culture: 1.1*
You're already doing it!!	Any collaborative project! Sharing Google docs
Extensions	Set up a collaborative project with another teacher (another grade level teacher at your school, a teacher in a different grade at your school, a teacher at a different GPS school, a teacher at a different district altogether!!) where students work in their respective classrooms using Google collaboration tools to complete a task/assignment

4.IC.SLE. 1 Impacts of Computing Safety, Law & Ethics	**Use material that is publicly available and/or permissible to use.** Ethical complications arise from the opportunities provided by computing. The ease of sending and receiving copies of media, such as video, photos, and music, on the Internet, creates the opportunity for unauthorized use, such as online piracy and disregard of copyrights. Students should consider the licenses for the computational artifacts that they wish to use. For example, the license on a downloaded image or audio file may have restrictions that prohibit modification, require attribution, or prohibit use entirely. *Practice(s): Communicating About Computing: 7.3*
You're already doing it!!	Discussing copyright and plagiarism with students
Extensions	Teach students to search for images labeled for reuse in Google Images Advanced

4th Grade CS Lessons

	Search

TIPS and TRICKS

AZ CS Standards Full document
Computational Thinking for Educators course offered by Google! This is a free self paced offering by Google to introduce teachers to delivering computational thinking lessons to their students of all ages!

The core of Computer Science is really computational thinking! You do this with your students all day every day!! An excellent way of getting your students to think like computer scientists is to speak to them about how your everyday routines relate to computer science!

Notice that none of these standards refer to a specific program or app! They are written to develop computational thinkers! So, keep doing what you're doing but help them make the connections to computer science through your discussions! Talk to your students using appropriate technological language to expose them to the terms that are used. When you give them an assignment on the computer have in depth discussions about the hardware, software and the processes to make them all work! When a student has an issue or trouble with their computer or the program, discuss with them what you are doing to fix it!

USEFUL WEBSITES AND RESOURCES

Computer Science Principles - An entire course offered by Code.org for High School
Self Paced 30 hour Computer Science course offered by code.org for Middle and High school
Code.org Course B
All you need is code - many premade lessons/activities
Complete lesson plans, presentations, etc. to deliver to your classes by Code for Fun
Many different lesson plans on different technology topics
How Computer work Khan Academy Unit
K-12 Digital Citizenship from Common Sense Media
Educators Guide to Copyright, Fair use and Creative Commons - a blog with great info about copyright, etc.