5th Grade CS Lessons

The computer literate student independently and collaboratively constructs programs and uses algorithms to accomplish real world tasks. Students continue to decompose larger problems into smaller tasks, recognize the impacts of computing and computing devices and model how computing systems work. The accurate use of terminology as well as the responsible use of technology will continue with emphasis on intellectual property rights.

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

COMPUTING SYSTEMS

5.CS.D.1 Computing Systems Devices	**Analyze and model how internal and external parts of computing devices communicate as a system**. Computing devices often depend on other devices or components. A robot depends on a physically attached light sensor to detect changes in brightness, whereas the light sensor depends on the robot for power. A smartphone can use wirelessly connected headphones to send audio information, and the headphones require a music source. *Practice(s): Communicating About Computing, Recognizing and Defining Computational Problems, Creating Computational Artifacts, Testing and Refining Computational Artifacts: 7.2, 3.2, 5.2, 6.3*
You're already doing it!!	A computing system is everything that connects to the "computer" which includes computers, gaming devices, vehicles, phones, remote controls for the television, etc. They are connected through wires, wirelessly or through the network. Discussing and demonstrating these devices is modeling this process! Include discussions about connectivity, how-to, etc. while using devices in the classroom daily!
Extensions	Have students create a step-by-step quick reference guide for a device of their choosing Students create a video demonstration connecting devices or describing parts of the computer

5.CS.D.2 Computer Systems Devices	**Explain how computing devices affect humans in positive and negative ways.** The use of computing devices has potential consequences, especially with regard to privacy and security. *Practice(s): Fostering an Inclusive Computing Culture, Communicating About Computing: 1.1, 7.2*
You're already doing it!!	Current events in technology constantly address the positive and negative effects Discussions about positive and negative effects in other subjects can transition into tech discussions
Extensions	Video about pros and cons to spark discussion Video about benefits of technology - Have students identify the cons to the pros addressed in video

	Students create an infographic about the positives and negatives using Canva Some discussion points to identify positive and negative that could be used as collaborative projects between students. Create a class slideshow where all students contribute a portion: ● Impacts on environment ● Impacts on society ● Social interactions ● Access to knowledge

5.CS.HS.1 Computer Systems Hardware & Software	**Model how information is transformed into binary digits to be stored or processed.** Hardware and software communicate in binary digits commonly represented in 0s and 1s. Information is transformed into binary digits, for example a song is stored as more binary digits than a photo. *Practice(s): Communicating About Computing, Creating Computational Artifacts: 7.2, 5.2*
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	Bits, Bytes, etc Video explaining bits, bytes, kb, mb, etc. Have students create scale models comparing different units Students create comparisons to other objects in life (example 1 cheerio, 8 cheerios (snack), 1000 cheerios in a bowl (kb), 1000000 in the box (mb), 1000000000 in the grocery store (gb) File Size lesson plan with visuals, videos, and activities - Priestlands Computer Science

5.CS.HS.2 Computer Systems Hardware & Software	**Demonstrate and explain how hardware can accomplish different tasks depending on the software**. In order for a person to accomplish tasks with a computer, both hardware and software are needed. At this stage, a model should only include the basic elements of a computer system, such as input, output, processor, sensors, and storage. Students could draw a model on paper or in a drawing program, program an animation to demonstrate it, or demonstrate it by acting it out in some way. *Practice(s): Communicating About Computing, Creating Computational Artifacts: 7.2, 5.3*
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

	CPU, input and output - Khan Academy

5.CS.T.1 Computer Systems Troubleshooting	**Apply potential solutions and solve simple hardware and software problems using common troubleshooting strategies**. Although computing systems may vary, common troubleshooting strategies such as checking connections in 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 computer machine. Computing devices are composed of an interconnected system of hardware and software, troubleshooting strategies may need to address both. In fifth grade students begin troubleshooting complex problems through networks, routers, and switches. *Practice(s): Recognizing and Defining Computational Problems, Developing and Using Abstractions: 3.2, 4.1*
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?" When troubleshooting your OWN issues, explain the steps you take while you do it.
Extensions	Create a classroom job for Student Onsite Support (Tech Expert). Review the Student Onsite Support document with the class at the beginning of the year so students are aware of basic troubleshooting tips.

NETWORKS & THE INTERNET

5.NI.C.1 Networks & Internet Cybersecurity	**Identify solutions to 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 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, 3.1*
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 Clickbait 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

5.NI. NCO.1 Networks & Internet Networks, Communication Organization	**Analyze the advantages and disadvantages of various network types**. There are physical paths for communicating information, such as Ethernet cables, and wireless paths, such as Wi-Fi or cellular data. The choice of device and type of connection will affect the path information travels and the potential bandwidth (the capacity to transmit data or bits in a given timeframe). *Practice(s): Developing and Using Abstractions, Collaborating Around Computing: 4.1, 2.4*
You're already doing it!!	Devices are connected either to each other or the internet in the classroom both wired and wireless. Explaining why we would hard wire vs. wireless ex. ● Computers are connected to Wi-Fi, if there is a problem with the Wi-Fi connection we would connect to the Ethernet cable. If we want something to download/upload more quickly we would connect to Ethernet ● Difficulty connecting wirelessly to projector - hardwire it. Playing video to projector will work better with it hard wired ● Turning cell phones off to eliminate some disruption in the wi-fi when large number of devices are connecting all at once (AzMerit testing!)
Extensions	The Internet - Unplugged activity by Code.org How the Internet Works - Video Networking Video- Learn with Purpose Students create a model of a network using common items Students make comparisons between networks and other common things (example, the nervous system, a road system - how the city streets differ from the freeway and how does that relate to networks?, etc)

DATA & ANALYSIS

5.DA.CVT.1 Data & Analysis	**Select tools to collect, organize, manipulate, and present data visually through multiple representations 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

Collection Visualization & Transformation	easier. Data points can be clustered by a number of commonalities. The same data could be manipulated and displayed in different formats to emphasize particular aspects or parts of the data set. *Practice(s): Developing and Using Abstractions, Creating Computational Artifacts: 4.1, 5.1*
You're already doing it!!	Providing different options for data presentation - Google docs, sheets, slides, canva, video, 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.) Categorizing objects based on commonalities
Extensions	Collect data using a Google form then using the sort, filter and graphing features in Google Sheets

5.DA.S.1 Data & Analysis Storage	**Discuss different file extensions and how they are stored and retrieved on a computing device**. 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. For example, two pictures of the same object can require different amounts of storage based upon their resolution. *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

5.DA.IM. 1 Data & Analysis Inference & Models	**Use data to propose cause-and-effect relationships, predict outcomes, or communicate an idea.** 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, Developing and Using Abstractions, Collaborate around Computing: 7.1, 4.3, 2.4*
You're already doing it!!	Data Analysis, Statistics and Probability standards in Math Inferences in Language Arts while reading Collecting weather data and predicting forecast in Science
Extensions	Data Analysis Statistics and Probability links to many lessons ideas!

ALGORITHMS & PROGRAMMING

5.AP.A.1 Algorithms & Programming Algorithms	**Compare, test, 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 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 shortest or fastest or would avoid a problem. Students might compare algorithms that describe how to get ready for school. 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. Students test their algorithms to verify their effectiveness. *Practice(s): Testing and Refining Computational Artifacts, Recognizing and Defining Computational Problems: 6.1, 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

5.AP.V.1 Algorithms & Programming Variables	**Recognizing that the data type determines the values that can be stored and the operations that can be performed on the data**. Variables are the vehicle through which computer programs store different types of 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). Data types vary by programming language, but many have types for numbers and text. Examples of operations associated with those types include multiplying numbers and combining text. Some visual, block-based languages do not have explicitly declared types but still have certain operations that apply only to particular types of data in a program. Programs can imply either digital or paper-based designs. Students create programs that use variables to store and modify data. *Practice(s): Creating Computational Artifacts: 5.2*
You're already doing it!!	Identifying people by name, student number, etc. all refers to variables Discussion in math can include CS variables Variable types (boolean, numeric, text, etc.) all occur naturally in the real world. Connecting boolean variables back to binary (0,1 - true, false) Fill in the blanks = entering information for a variable. Will it be numeric, single word, paragraph?
Extensions	Variable unplugged lesson - Code.org How to teach variables to kids (article) - Tynkr.org

5.AP.C.1 Algorithms & Programming Control	**Create programs that include sequences, events, loops, and 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. For example, 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. For example, 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. For example, 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. For example, 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: 5.1*
You're already doing it!!	Sequencing, placing things in order Cause and Effect Timelines Cycles
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

5.AP.M.1 Algorithms & Programming Modularity	**Decompose problems into 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. 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!!	
Extensions	

5.AP.M.2 Algorithms & Programming Modularity	**Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or 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!!	
Extensions	

5.AP.PD.1 Algorithms & Programming Programming 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!!	
Extensions	

5.AP.PD.2 Algorithms & Programming Programming 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!!	
Extensions	

5.AP.PD.3 Algorithms & Programming Program Development	**Test and debug (identify and fix errors) a program 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!!	
Extensions	

5.AP.PD.4 Algorithms & Programming Program Development	**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!!	
Extensions	

5.AP.PD.5 Algorithms & Programming Program Development	**Describe choices made during program development using code comments, presentations, and 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!!	
Extensions	

IMPACTS OF COMPUTING

5.IC.C.1 Impacts of Computing Culture	**Discuss computing technologies that have changed the world.** New computing technology is created and existing technologies are modified for many reasons, including in order to increase their benefits, decrease their risks, and meet societal needs. Students 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, cultural and political changes. *Practice(s): Recognizing and Defining Computational Problems: 3.1*
You're already doing it!!	
Extensions	

5.IC.C.2 Impacts of Computing Culture	**Design 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 shares their own tastes. *Practice(s): Fostering an Inclusive Computing Culture: 1.2*
You're already doing it!!	
Extensions	

5.IC.SI.1 Impacts of Computing Social Interactions	**Seek opportunities for local and global 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!!	
Extensions	

5.IC. SLE.1 Impacts of Computing Safety, Law & Ethics	**Use public domain or creative commons media, and refrain from copying or using material created by others without permission**. 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!!	
Extensions	

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!

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.