# 2nd Grade CS Lessons

Students expand basic knowledge of digital literacy and computer science skills. By the end of second grade, students will be able to select the appropriate device and software to perform specific tasks. Students will understand that computing systems use both hardware and software to process information and will be able to use basic troubleshooting strategies with teacher guidance. Students will develop a deeper understanding of the importance and use of strong passwords to protect private information and discuss how computer networks can connect people globally. Second grade students will independently collect, transform, and display data using digital devices. Students can store, copy, search, retrieve, modify, and delete information using a computing device. Second grade students will create and follow algorithms and programs using loops and variables to solve a problem. Students will decompose steps into simpler tasks and give credit to the ideas and creations of others. Second grade students will discuss the impacts of computing by comparing how people lived and worked before and after the implementation of new technology, how to work responsibly online, the importance of keeping login information private, and logging off devices appropriately.

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

## COMPUTING SYSTEMS

| | |
|---|---|
| **2.CS.D.1**<br>Computing Systems<br>Devices | **Recognize that users have different needs and preferences for technology they used by selecting and operating appropriate devices.**<br>People use computing devices to perform a variety of tasks accurately and quickly. Students should be able to select the appropriate app/program to use for tasks they are required to complete. For example, if students are asked to draw a picture, they should be able to open and use a drawing app/program to complete this task, or if they are asked to create a presentation, they should be able to open and use presentation software. In addition, with teacher guidance, students should identify and discuss preferences for software with the same primary functionality. *Practice(s): Fostering an inclusive Computing Culture, Communicating About Computing: 1.1, 7.3* |
| You're already doing it!! | When guiding students to perform a task, indicate the program you want them to use then indicate why. Additionally, you can ask them which program would best complete the task at hand. Here are some examples:<br>● We're going to open a Google Doc because we use Google Docs to create documents.<br>● We're going to open Google Drawing because it is a program used for drawing<br>● Click on Google Chrome. Chrome is the browser we use to access the internet<br>● We're going to make a presentation. Do you think we should use Google Slides or Google Docs (both are correct so ask them why they chose it) |
| Extensions | Have students decide which program would work best to complete a task<br>Use a variety of programs for projects - create charts in Sheets, presentations in Slides, write papers in Docs, find images and do research using Google<br>Introduce students to new programs - canva.com to create posters<br>Have students begin using multiple programs for the same project! Create a chart in |

| | |
|---|---|
| | google sheets then copy and paste it into a presentation. |

| | |
|---|---|
| **2.CS.HS.1**<br>Computer Systems<br>Hardware & Software | **Understand how computing systems use both hardware (device) and software (program/app) to process information.**<br>A computing system is composed of hardware and software. Hardware consists of physical components. Students should be able to identify and describe the function of external hardware, such as desktop computers, laptop computers, tablet devices, monitors, keyboards, mice, and printers. *Practice(s): Communicating about Computing : 7.2* |
| You're already doing it!! | The best way to guide students to mastering this standard is to use proper terminology while talking to them about their computers. Include computer terms when they are learning words |
| Extensions | What is a Computer slideshow by Code for Fun<br>Create your own computer cut and paste by Code for Fun<br>Computer parts lesson plan (from Bloom Into EdTech blog)<br>Computer parts flash cards<br>Computer parts memory game (this picture dictionary could be cut out to make a memory game!)<br>Video about computer parts  (3:11)<br>Parts of a computer slideshow and game |

| | |
|---|---|
| **2.CS.T.1**<br>Computer Systems<br>Troubleshooting | **Explain basic hardware (device) and software (program/app) problems using accurate terminology.**<br>Problems with computing systems have different causes. Students at this level do not need to understand those causes, but they should be able to communicate a problem with accurate terminology (e.g., when an app or program is not working as expected, the device is frozen, the link doesn't work, the internet is not connecting. *Practice(s): Communicating About Computing: 7.2* |
| You're already doing it!! | When a student presents an issue "My ChormeBook isn't working", have them articulate the problem. Have them use descriptive words and appropriate terminology, then discuss possible solutions. For example:<br>Student:  My screen is black<br>Teacher:  Your monitor is black? Do you think the battery died? Should we plug it in?<br><br>Student: My game's not working!!<br>Teacher: You're not at the correct website. Maybe we should check to make sure you entered the URL correctly<br><br>Student: My computer is locked up<br>Teacher: Try restarting it (then show them how!) |

| | |
|---|---|
| Extensions | As part of your rotating classroom jobs, include an IT Customer Care Specialist (Information Technology) assignment. Have this student responsible for making sure the ChromeBooks are plugged in accurately at the end of class. Allow them to be the "expert" that students go to first if there is a problem<br>Identify experts within the classroom (in groups, at tables, etc.) that students should ask before coming to the teacher for help.<br>Come up with a "help desk ticket" system when using the technology so students submit their issue and it can be addressed AFTER you have finished teaching the lesson. |

| | |
|---|---|
| **2.CS.T.2**<br>Computer Systems<br>Troubleshooting | **With teacher guidance, use basic troubleshooting strategies.**<br>Students should be able to use simple troubleshooting strategies, including turning a device off and on to reboot it, closing and reopening an app, turning on speakers, or plugging in headphones and adjusting volume. Practice(s): Testing and Refining Computational Artifacts: 6.2 |
| You're already doing it!! | Having students adjust volume<br>When students articulate trouble with the device ask them "what have you tried to do to fix it?" or "what can you try to do to fix it?" Having students figure out basic issues - frozen apps, no sound, etc. |
| Extensions | Refresh a page<br>Check internet connection |

## NETWORKS & THE INTERNET

| | |
|---|---|
| **2.NI.C.1**<br>Networks & Internet<br>Cybersecurity | **Explain what passwords are and why we use them, and use strong passwords to protect devices and information from unauthorized access.**<br>Connecting devices to a network or the Internet provides great benefit, care must be taken to use authentication measures, such as strong passwords, to protect devices and information from unauthorized access. This is an essential first step in learning about cybersecurity. They should appropriately use and protect the passwords they are required to use. Usernames and passwords, such as those on computing devices or Wi-Fi networks, provide a way of authenticating a user's identity. For example, students learn to not share passwords and not use anyone else's password. *Practice(s): Communicating About Computing: 7.2* |
| You're already doing it!! | Discuss with students why we don't share passwords. Login information should be kept private to protect the documents on your computer. Additionally, it protects you from having someone do something they shouldn't be doing under YOUR name!! |
| Extensions | Strong Password video (1:46) |

| | |
|---|---|
| **2.NI.NCO.1**<br>Networks & Internet<br>Cybersecurity | **Students can discuss how computer networks can be used to connect people to other people, places, information, and ideas.**<br>Small, wireless devices, such as cell phones, communicate with one another through a series of intermediary connection points, such as cellular towers. This coordination among many computing devices allows a person to voice call a friend or video chat with a family member. Details about the connection points are not expected at this level. For example, students will participate in a class discussion about how different networks connect people, places, things and information, such as a phone call to grandma in another state, using conferencing software to connect with a content area expert, or accessing an online game via Wi-Fi. *Practice(s): Communicating About Computing: 7.2* |
| You're already doing it!! | When connecting wirelessly to the projector, using the chromebooks with WiFi, etc. explain what you are doing!  "I'm going to connect wirelessly to the projector!"<br>Discussions with students about how we are connected with people - phone, video calls, etc. |
| Extensions | What is the Internet (2:23)<br>The Internet! How it Works! (8:29)<br>Network Connections Lesson |

## DATA & ANALYSIS

| | |
|---|---|
| **2.DA.CVT.1**<br>Data & Analysis<br>Collection Visualization & Transformation | **Collect and transform data using digital devices; Display data for communication in various visual formats.**<br>The collection and use of data about the world around them is a routine part of life and influences how people live. Many everyday objects, such as cell phones, digital toys, and cars, can contain tools (such as sensors) and computers to collect and display data from their surroundings. Students could collect data on the weather, such as sunny days versus rainy days, the temperature at the beginning of the school day and end of the school day, or the inches of rain over the course of a storm. Students could count the number of pieces of each color of candy in a bag of candy, such as Skittles or M&Ms. Students could create surveys of things that interest them, such as favorite foods, pets, or TV shows, and collect answers to their surveys from their peers and others. The data collected could then be organized into two or more visualizations, such as a bar graph, pie chart, or pictograph. *Practice(s): Communicating About Computing, Developing and Using Abstractions: 7.3, 4.2* |
| You're already doing it!! | Counting, sorting and graphing in Math<br>Collecting data in Science<br>Recording reading minutes/log |
| Extensions | Students can create pictographs (abcya.com paint - using the shape, stamp or sticker tools)  Or with Google Drawings |

| | |
|---|---|
| | Enter data in a Google Sheet and create a chart (put a link to the sheet on the teacher directed website and students can easily access it!) make a different sheet for every student to enter their data (example) |

| | |
|---|---|
| **2.DA.S.1**<br>Data & Analysis<br>Storage | **Store, copy, search, retrieve, modify, and delete information using a computing device and define the information stored as data.**<br>All information stored and processed by a computing device is referred to as data. Data can be images, text documents, audio files, software programs or apps, video files, etc. It can be retrieved, copied, and stored in multiple places. As students use software to complete tasks on a computing device, they will be manipulating data. For example, students will learn to save files in specific locations, such as a folder, and retrieve those files for use later. *Practice(s): Developing and Using Abstractions: 4.1* |
| You're already doing it!! | Any time students create, open and reopen documents they are learning that the files created were saved!<br>When creating files teach students to NAME THE FILE and create folders to ORGANIZE THEIR DRIVE so they can locate files later!<br>Use the SEARCH bar in Google drive to locate files. Advanced search is also a possibility! |
| Extensions | Create documents in google Docs and open them again later! You can even create a template document and force a copy like this!! |

| | |
|---|---|
| **2.DA.IM.1**<br>Data & Analysis<br>Inference & Models | **Describe patterns in data to make inferences or predictions.**<br>Data can be used to make inferences or predictions about the world. Students could analyze a Graph and pie chart of the colors in a bag of candy identify the patterns for which colors are most and least represented, and then make a prediction as to which colors will have most and least in a new bag of candy. *Practice(s): Developing and Using Abstractions: 4.3* |
| You're already doing it!! | Analyzing charts (behavior, reading, etc)<br>Graphs in Math / Science<br>Predicting the end of a story<br>Using context clues when reading<br>Goal setting |
| Extensions | Create charts in Google Sheets from data students have collected. Have students compare their charts with each other!<br>Have students enter their reading log data into a premade spreadsheet and discuss trends (are grades improving when you are reading more?) (example) |

| | |
|---|---|
| | Have students create a survey in Google forms and discuss the data they collect in a spreadsheet. |

## ALGORITHMS & PROGRAMMING

| | |
|---|---|
| **2.AP.A.1**<br>Algorithms & Programming<br>Algorithms | **Model daily processes by creating and following algorithms (sets of step-by-step instructions to complete tasks.**<br>Routines, such as morning meeting, clean-up time, and dismissal, are examples of algorithms that are common in many early elementary classrooms. Just as people use algorithms to complete daily routines, they can program computers to use algorithms to complete different tasks. Algorithms are commonly implemented using a precise language that computers can interpret. For example, students begin to understand and model daily step-by-step processes, such as brushing teeth, implementing a morning procedure, or following a simple recipe as "algorithms" that lead to an end result. *Practice(s): Developing and Using Abstractions: 4.3* |
| You're already doing it!! | Clapping to get their attention!  When I clap, you clap!  This is an if/then command!<br>Math problems need to be completed in a specific order<br>Writing conventions (capital letters, periods)<br>Following a set of directions to complete a task<br>Sorting things in alpha order<br>Discussing recipes<br>Any repetitive task<br>Completing a word search or sudoku |
| Extensions | Discuss routines and how they relate to computer science. Use the word algorithm  to expose them to the vocabulary<br>Algorithm lesson by Code for Fun<br>Searching Algorithms lessons by CS Unplugged |

| | |
|---|---|
| **2.AP.V.1**<br>Algorithms & Programming<br>Variables | **Model the way programs store and manipulate data by using numbers or other symbols to represent information.**<br>Information in the real world can be represented in computer programs. Students could use thumbs up/down as representations of yes/no, use arrows when writing algorithms to represent direction, or encode and decode words using numbers, pictographs, or other symbols to represent letters or words. *Practice(s): Developing and Using Abstractions: 4.3* |
| You're already doing it!! | Decoding words in language arts<br>Following arrows in directions<br>Colors to represent the noise level<br>Hand signals to identify different actions they want to take (bathroom, drink, sharpen |

| | |
|---|---|
| | pencil, etc.)<br>Hand signals to identify understanding - thumbs up/down, fist to five, etc.<br>Secret code activities |
| Extensions | Happy Maps - code.org unplugged<br>Binary Bracelets - code.org unplugged<br>Navigate a maze using arrows - set a maze on the floor and hold up arrows for them to navigate<br>The Big Event - code.org unplugged<br>Use emojis to tell a story |

| | |
|---|---|
| **2.AP.C.1**<br>Algorithms & Programming<br>Control | **Develop programs with sequences and simple loops, to express ideas or address a problem.**<br>Programming is used as a tool to create products that reflect a wide range of interests. Control structures specify the order in which instructions are executed within a program. Computers follow instructions literally. Sequences are the order of instructions in a program. For example, sequences of instructions include steps for drawing a shape or moving a character across the screen. If the commands to program a robot are not in the correct order, the robot will not complete the task desired. Loops allow for the repetition of a sequence of code multiple times. For example, in a program to show the life cycle of a butterfly, a loop could be combined with move commands to allow continual but controlled movement of the character. For example: students independently identify loops and sequences in songs, rhymes, and games, such as the song Head, Shoulders, Knees and Toes. *Practice(s): Creating Computational Artifacts: 5.2* |
| You're already doing it!! | Discussing cycles (water cycle, life cycle,  etc.) is a sequence with loops!<br>Any time you tell students to "continue until…" it is a loop!  (musical chairs!!)<br>Giving step by step instructions that must be done in order - sequences!<br>Timelines in social studies<br>Experiments in Science (Scientific Method) |
| Extensions | Code.org Course B - premade curriculum that students can complete independently. It contains videos, activities and coding and walks them through step by step! |

| | |
|---|---|
| **2.AP.M.1**<br>Algorithms & Programming<br>Modularity | **Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.**<br>Decomposition is the act of breaking down tasks into simpler tasks. Students could break down the steps needed to make a peanut butter and jelly sandwich, to brush their teeth, to draw a shape, to move a character across the screen, or to solve a level of a coding App. *Practice(s): Recognizing and Defining Computational Problems: 3.2* |
| You're already doing it!! | Decoding words by sound<br>Step by step instructions in anything<br>Completing a jigsaw or any other type of puzzle |

| | |
|---|---|
| Extensions | Code.org Course B (this has multiple activities that students can navigate)<br>Here's an explanation of modularity and CS (not necessarily for students - maybe too advanced but a way to wrap your own head around it!)<br>Decompose lesson by BrainPop Educators |

| | |
|---|---|
| **2.AP.PD.1**<br>Algorithms & Programming<br>Program Development | **Develop plans that describe a program's sequence of events, goals, and expected outcomes.**<br>Programming is used as a tool to create products that reflect a wide range of interests, such as video games, interactive art projects, and digital stories. Students could create a planning document, such as a story map, a storyboard, or a sequential graphic organizer, to illustrate what an end product will do. Students at this stage may complete the planning process with help from their teachers. For example, students create a graphic organizer modeling the life cycle of a plant. *Practice(s): Creating Computational Artifacts, Communicating About Computing: 5.2, 7.2* |
| You're already doing it!! | Thinking maps<br>Goal setting<br>Sequencing events in a story |
| Extensions | Programming in code.org with angry birds!<br>Great computational thinking game that ties with Math! |

| | |
|---|---|
| **2.AP.PD.2**<br>Algorithms & Programming<br>Program Development | **Give attribution (credit) when using the ideas and creations of others while developing programs.**<br>Using computers comes with a level of responsibility. Students should credit artifacts that were created by others, such as pictures, music, and code. Proper attribution at this stage does not require a formal citation, such as in a bibliography or works cited document. For example, students can give attribution in written form, at a minimum, by listing a website where they got the information, picture, or music. *Practice(s): Communicating About Computing: 7.3* |
| You're already doing it!! | Discussing the author of a book or a piece of art<br>Students putting their name on their paper - gives THEM credit!<br>Not stealing other people's work<br>Having students identify where they got information (which book, website, person, etc.) |
| Extensions | Here's a good article about copyright and talking to kids about it.<br>Citations for elementary (video and resources!) |

| | |
|---|---|
| **2.AP.PD.3**<br>Algorithms & Programming<br>Programming Development | **Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops.**<br>Algorithms or programs may not always work correctly. Students should be able to use various strategies, such as changing the sequence of the steps, following the algorithm in a step-by-step manner, or trial and error to fix problems in algorithms and programs.<br>For example: Students could use arrows on a grid to map out a path to a specific coordinate, such as a treasure map. If the steps were repeated, it would create a loop.<br>*Practice(s): Testing and Refining Computational Artifacts: 6.2* |
| You're already doing it!! | Cause and effect<br>Solving problems<br>Doing test corrections |
| Extensions | Debugging activities from Code for Fun<br>GREAT debugging challenges from Scratch! |

| | |
|---|---|
| **2.AP.PD.4**<br>Algorithms & Programming<br>Programming Development | **Using correct terminology, describe steps taken and choices made during the iterative process of program (procedure) development.**<br>At this stage, students should be able to talk or write about the goals and expected outcomes of the programs they create and the choices that they made when creating programs. This could be done using coding journals, or discussions with a teacher.<br>Students work together to explain the steps in their procedure. *Practice(s): Communicating About Computing: 7.2* |
| You're already doing it!! | Solving and correcting math problems<br>Asking "how did you come up with that answer?"<br>Describing steps taken to complete a task |
| Extensions | Coding in Scratch Jr. |

## IMPACTS OF COMPUTING

| | |
|---|---|
| **2.IC.C.1**<br>Impacts of Computing<br>Culture | **Compare how people live and work before and after the implementation or adoption of new computing technology.**<br>Computing technology has positively and negatively changed the way people live and work. In the past, if students wanted to read about a topic, they needed access to a library to find a book about it. Today, students can view and read information on the Internet about a topic or they can download e-books about it directly to a device. Such information may be available in more than one language and could be read to a student, allowing for great accessibility. *Practice(s): Communicating About Computing: 7.1* |
| You're already doing it!! | History and social studies<br>Changes in technology<br>Current events |

| | |
|---|---|
| Extensions | Show examples of the evolution of different technology - phones, computers, cameras, etc.<br>Discuss advantages and disadvantages of old and new technology<br>Kids react to old computers video (7:41) |

| | |
|---|---|
| **2.IC.SI.1**<br>Impacts of Computing<br>Social Interaction | **Work respectfully and responsibly with others online.**<br>Online communication facilitates positive interactions, such as sharing ideas with many people, but the public and anonymous nature of online communication also allows intimidating and inappropriate behavior in the form of cyberbullying. Students could share their work on blogs or in other collaborative spaces online, taking care to avoid sharing information that is inappropriate or that could personally identify them to others. Students could provide feedback to others on their work in a kind and respectful manner. They should tell an adult if others are sharing things they should not share or are treating others in an unkind or disrespectful manner on online. Privacy should be considered when posting information online: such information can persist for a long time and be accessed by others, even unintended viewers. *Practice(s): Collaborating Around Computing: 2.1* |
| You're already doing it!! | Behavior expectations<br>Anti-bullying discussions |
| Extensions | Know the Rule video (1:53)<br>Use Your Netsmartz video (2:27)<br>What is Personal Information (2:41) |

| | |
|---|---|
| **2.IC. SLE.1**<br>Impacts of Computing<br>Safety, Law & Ethics | **Keep login information private, and log off of devices appropriately**.<br>Using computers comes with a level of responsibility, such as not sharing login information, keeping passwords private, and logging off when finished. Rules guiding personal interactions in the world apply to online environments as well. For example, students routinely practice logging in and logging out of online resources to protect their personal information. Students should also commit to interacting with only those they know in person in online environments. *Practice(s): Communicating About Computing: 7.2* |
| You're already doing it!! | Inform students that they need to log on and off to prevent others from using the computer under their log in. Because it saves files, etc.<br>Reminding students the importance of keeping passwords private so others can't access your information online. |
| Extensions | Password rap video (2:00)<br>NetSafe Tell an Adult video (2:02)<br>Understanding Online "Friends" video (2:12) |

## 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
K-12 Digital Citizenship  from Common Sense Media