California Computer Science

Grades 3-5

Computing Systems

1 Describe how computing devices connect to other components to form a system. 3-5.CS.1

Computing devices often depend on other devices or components. Students describe physical and wireless connections to other components, including both input devices (e.g., keyboards, sensors, remote controls, microphones) and output devices (e.g., 3D printers, monitors, speakers).

2 Demonstrate how computer hardware and software work together as a system to accomplish tasks. 3-5.CS.2

Hardware and software are both needed to accomplish tasks with a computing device. Students create a model to illustrate ways in which hardware and software work as a system. Students could draw a model on paper or in a drawing program, program an animation to demonstrate it, or demonstrate it by acting this out in some way. At this level, a model should only include the basic elements of a computer system, such as input, output, processor, sensors, and storage.

3 Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies. 3-5.CS.3

Although computing systems vary, common troubleshooting strategies can be used across many different systems. Students use troubleshooting strategies to identify problems that could include a device not responding, lacking power, lacking a network connection, an app crashing, not playing sounds, or password entry not working. Students use and develop various solutions to address these problems. Solutions may include rebooting the device, checking for power, checking network availability, opening and closing an app, making sure speakers are turned on or headphones are plugged in, and making sure that the caps lock key is not on.

Networks & Internet

4 Model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the Internet, and reassembled at the destination. 3-5.NI.4

Information is sent and received over physical or wireless paths. It is broken down into smaller pieces called packets, which are sent independently and reassembled at the destination. Students demonstrate their understanding of this flow of information by, for instance, drawing a model of the way packets are transmitted, programming an animation to show how packets are transmitted, or demonstrating this through an unplugged activity in which they physically act this out.

Skip to main content

5 Describe physical and digital security measures for protecting personal information. 3-5.NI.5

Personal information can be protected physically and digitally. Cybersecurity is the protection from unauthorized use of electronic data, or the measures taken to achieve this. Students identify what personal information is and the reasons for protecting it. Students describe physical and digital approaches for protecting personal information such as using strong passwords and biometric scanners.

6 **Create patterns to protect information from unauthorized access.** `3-5.NI.6`

Encryption is the process of converting information or data into a code, especially to prevent unauthorized access. At this level, students use patterns as a code for encryption, to protect information. Patterns should be decodable to the party for whom the message is intended, but difficult or impossible for those with unauthorized access.

Data & Analysis

7 **Explain that the amount of space required to store data differs based on the type of data and/or level of detail.** `3-5.DA.7`

All saved data requires space to store it, whether locally or not (e.g., on the cloud). Music, images, video, and text require different amounts of storage. Video will often require more storage and different format than music or images alone because video combines both. The level of detail represented by that data also affects storage requirements. For instance, two pictures of the same object can require different amounts of storage based upon their resolution, and a high-resolution photo could require more storage than a low-resolution video. Students select appropriate storage for their data.

8 **Organize and present collected data visually to highlight relationships and support a claim.** `3-5.DA.8`

Raw data has little meaning on its own. Data is often sorted or grouped to provide additional clarity. Organizing data can make interpreting and communicating it to others easier. Data points can be clustered by a number of commonalities. The same data could be manipulated in different ways to emphasize particular aspects or parts of the data set.

9 **Use data to highlight and/or propose relationships, predict outcomes, or communicate ideas.** `3-5.DA.9`

The accuracy of data analysis is related to how the data is represented. Inferences or predictions based on data are less likely to be accurate if the data is insufficient, incomplete, or inaccurate or if the data is incorrect in some way. Additionally, people select aspects and subsets of data to be transformed, organized, and categorized. Students should be able to refer to data when communicating an idea, in order to highlight and/or propose relationships, predict outcomes, highlight different views and/or communicate insights and ideas.

Algorithms & Programming

10 **Compare and refine multiple algorithms for the same task and determine which is the most appropriate.** `3-5.AP.10`

Skip to main content

Different algorithms can achieve the same result, though sometimes one algorithm might be more appropriate for a specific solution. Students examine different ways to solve the same task and decide which would be the better solution for the specific scenario.

11 Create programs that use variables to store and modify data. 3-5.AP.11

Variables are used to store and modify data. Students use variables in programs they create. At this level, students may need guidance in identifying when to create variables (i.e., performing the abstraction).

12 Create programs that include events, loops, and conditionals. 3-5.AP.12

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. Events allow portions of a program to run based on a specific action. Conditionals allow for the execution of a portion of code in a program when a certain condition is true. Loops allow for the repetition of a sequence of code multiple times.

13 Decompose problems into smaller, manageable tasks which may themselves be decomposed. 3-5.AP.13

Decomposition is the act of breaking down tasks into simpler tasks. This manages complexity in the problem solving and program development process. For example, students could create an animation to represent a story they have written. Students write a story and then break it down into different scenes. For each scene, they would select a background, place characters, and program actions in that scene.

14 Create programs by incorporating smaller portions of existing programs, to develop something new or add more advanced features. 3-5.AP.14

Programs can be broken down into smaller parts, which can be incorporated into new or existing programs. Students incorporate predefined functions into their original designs. At this level, students do not need to understand all of the underlying implementation details of the abstractions that they use. For example, students could use code from a ping pong animation to make a ball bounce in a new basketball game. They could also incorporate code from a single-player basketball game to create a two-player game with slightly different rules.

15 Use an iterative process to plan and develop a program by considering the perspectives and preferences of others. 3-5.AP.15

Planning is an important part of the iterative process of program development. Students gain a basic understanding of the importance and process of planning before beginning to write code for a program. They plan the development of a program by outlining key features, time and resource constraints, and user expectations. Students should document the plan as, for example, a storyboard, flowchart, pseudocode, or story map.

Skip to main content

Observe intellectual property rights and give appropriate attribution when creating, remixing, or combining programs. 3-5.AP.16

Intellectual property rights can vary by country, but copyright laws give the creator of a work a set of rights and prevents others from copying the work and using it in ways that they may not like. Students consider common licenses that place limitations or restrictions on the use of others' work, such as images and music downloaded from the Internet. When incorporating the work of others, students attribute the work. At this level, students could give attribution by including credits or links directly in their programs, code comments, or separate project pages.

17 **Test and debug a program or algorithm to ensure it accomplishes the intended task.** `3-5.AP.17`

Programs do not always run properly. Students need to understand how to test and make necessary corrections to their programs to ensure they run properly. Students successfully identify and fix errors in (debug) their programs and programs created by others. Debugging strategies at this level may include testing to determine the first place the solution is in error and fixing accordingly, leaving "breadcrumbs" in a program, and soliciting assistance from peers and online resources.

18 **Perform different roles when collaborating with peers during the design, implementation, and review stages of program development.** `3-5.AP.18`

Collaborative computing is the process of creating computational artifacts by working in pairs or on teams. It involves asking for the contributions and feedback of others. Effective collaboration can often lead to better outcomes than working independently. With teacher guidance, students take turns in different roles during program development, such as driver, navigator, notetaker, facilitator, and debugger, as they design and implement their program.

19 **Describe choices made during program development using code comments, presentations, and demonstrations.** `3-5.AP.19`

People communicate about their code to help others understand and use their programs. Explaining one's design choices gives others a better understanding of one's work. Students may explain their step-by-step process of creating a program in a presentation or demonstration of their personal code journals. They describe how comments within code organize thought and process during the develpment of the program.

Impacts of Computing

20 **Discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices.** `3-5.IC.20`

New computing technologies are created and existing technologies are modified for many reasons, including to increase their benefits, decrease their risks, and meet societal needs. Students, with guidance from their teacher, discuss topics that relate to the history of computing technologies and changes in the world due to these technologies. Topics could be based on current news content, such as robotics, wireless Internet, mobile computing devices, GPS systems, wearable computing, and how social media has influenced social and political changes.

Skip to main content

21 **Propose ways to improve the accessibility and usability of technology products for the diverse needs and wants of users.** 3-5.IC.21

The development and modification of computing technology is driven by people's needs and wants and can affect groups differently. Students anticipate the needs and wants of diverse end users and propose ways to improve access and usability of technology, with consideration of potential perspectives of users with different backgrounds, ability levels, points of view, and disabilities.

22 **Seek and explain the impact of diverse perspectives for the purpose of improving computational artifacts.** 3-5.IC.22

Computing technologies enable global collaboration and sharing of ideas. Students solicit feedback from a diverse group of users and creators and explain how this input improves their computational artifacts.

23 **Describe reasons creators might limit the use of their work.** 3-5.IC.23

Ethical complications arise from the opportunities provided by computing. With the ease of sending and receiving copies of media on the Internet, in formats such as video, photos, and music, students consider the opportunities for unauthorized use, such as online piracy and disregard of copyrights. The license of a downloaded image or audio file may restrict modification, require attribution, or prohibit use entirely.