K12 COMPUTER SCIENCE FRAMEWORK

VIEW BY GRADE BAND: Abridged

The Concepts and Practices of the K–12 Computer Science Framework

Core Concepts

1. Computing Systems
2. Networks and the Internet
3. Data and Analysis
4. Algorithms and Programming
5. Impacts of Computing

Core Practices

1. Fostering an Inclusive Computing Culture
2. Collaborating Around Computing
3. Recognizing and Defining Computational Problems
4. Developing and Using Abstractions
5. Creating Computational Artifacts
6. Testing and Refining Computational Artifacts
7. Communicating About Computing

Suggested citation: K–12 Computer Science Framework. (2016). *Framework view by grade band, abridged.* Retrieved from http://www.k12cs.org

Suggested attribution: "The K–12 Computer Science Framework, led by the Association for Computing Machinery, Code.org, Computer Science Teachers Association, Cyber Innovation Center, and National Math and Science Initiative in partnership with states and districts, informed the development of this work."

How to refer to the concepts: [Grade Band].[Core Concept].[Subconcept]
Example: K–2.Algorithms and Programming.Program Development

How to refer to the practices: P[Practice Number].[Core Practice].[Practice Statement Number]
Example: P4.Developing and Using Abstractions.1

Practices

Practice 1. Fostering an Inclusive Computing Culture

Overview: Building an inclusive and diverse computing culture requires strategies for incorporating perspectives from people of different genders, ethnicities, and abilities. Incorporating these perspectives involves understanding the personal, ethical, social, economic, and cultural contexts in which people operate. Considering the needs of diverse users during the design process is essential to producing inclusive computational products.

By the end of Grade 12, students should be able to

1. **Include the unique perspectives of others** and reflect on one's own perspectives when designing and developing computational products.
2. **Address the needs of diverse end users** during the design process to produce artifacts with broad accessibility and usability.
3. **Employ self- and peer-advocacy** to address bias in interactions, product design, and development methods.

Practice 2. Collaborating Around Computing

Overview: Collaborative computing is the process of performing a computational task by working in pairs and on teams. Because it involves asking for the contributions and feedback of others, effective collaboration can lead to better outcomes than working independently. Collaboration requires individuals to navigate and incorporate diverse perspectives, conflicting ideas, disparate skills, and distinct personalities. Students should use collaborative tools to effectively work together and to create complex artifacts.

By the end of Grade 12, students should be able to

1. **Cultivate working relationships** with individuals possessing diverse perspectives, skills, and personalities.
2. **Create team norms, expectations, and equitable workloads** to increase efficiency and effectiveness.
3. **Solicit and incorporate feedback** from, and provide constructive feedback to, team members and other stakeholders.
4. **Evaluate and select technological tools** that can be used to collaborate on a project.

Practice 3. Recognizing and Defining Computational Problems

Overview: The ability to recognize appropriate and worthwhile opportunities to apply computation is a skill that develops over time and is central to computing. Solving a problem with a computational approach requires defining the problem, breaking it down into parts, and evaluating each part to determine whether a computational solution is appropriate.

By the end of Grade 12, students should be able to

1. **Identify complex, interdisciplinary, real-world problems** that can be solved computationally.
2. **Decompose complex real-world problems** into manageable subproblems that could integrate existing solutions or procedures.
3. **Evaluate whether it is appropriate and feasible** to solve a problem computationally.

Practice 4. Developing and Using Abstractions

Overview: Abstractions are formed by identifying patterns and extracting common features from specific examples to create generalizations. Using generalized solutions and parts of solutions designed for broad reuse simplifies the development process by managing complexity.

By the end of Grade 12, students should be able to

1. **Extract common features** from a set of interrelated processes or complex phenomena.
2. **Evaluate existing technological functionalities** and **incorporate t**hem into new designs.
3. **Create modules** and **develop points of interaction** that can apply to multiple situations and reduce complexity.
4. **Model phenomena and processes** and **simulate systems** to understand and evaluate potential outcomes.

Practice 5. Creating Computational Artifacts

Overview: The process of developing computational artifacts embraces both creative expression and the exploration of ideas to create prototypes and solve computational problems. Students create artifacts that are personally relevant or beneficial to their community and beyond. Computational artifacts can be created by combining and modifying existing artifacts or by developing new artifacts. Examples of computational artifacts include programs, simulations, visualizations, digital animations, robotic systems, and apps.

By the end of Grade 12, students should be able to

1. **Plan the development** of a computational artifact using an iterative process that includes reflection on and modification of the plan, taking into account key features, time and resource constraints, and user expectations.
2. **Create a computational artifact** for practical intent, personal expression, or to address a societal issue.
3. **Modify an existing artifact** to improve or customize it.

Practice 6. Testing and Refining Computational Artifacts

Overview: Testing and refinement is the deliberate and iterative process of improving a computational artifact. This process includes debugging (identifying and fixing errors) and comparing actual outcomes to intended outcomes. Students also respond to the changing needs and expectations of end users and improve the performance, reliability, usability, and accessibility of artifacts.

By the end of Grade 12, students should be able to

1. **Systematically test** computational artifacts by considering all scenarios and using test cases.
2. **Identify and fix errors** using a systematic process.
3. **Evaluate and refine** a computational artifact multiple times to enhance its performance, reliability, usability, and accessibility.

Practice 7. Communicating About Computing

Overview: Communication involves personal expression and exchanging ideas with others. In computer science, students communicate with diverse audiences about the use and effects of computation and the appropriateness of computational choices. Students write clear comments, document their work, and communicate their ideas through multiple forms of media. Clear communication includes using precise language and carefully considering possible audiences.

By the end of Grade 12, students should be able to

1. **Select, organize, and interpret** large data sets from multiple sources to support a claim.
2. **Describe, justify, and document** computational processes and solutions using appropriate terminology consistent with the intended audience and purpose.
3. **Articulate ideas responsibly** by observing intellectual property rights and giving appropriate attribution.

Concepts

By the end of Grade 2

Computing Systems	Networks and the Internet	Data and Analysis	Algorithms and Programming	Impacts of Computing
DEVICES People use computing devices to perform a variety of tasks accurately and quickly. Computing devices interpret and follow the instructions they are given literally. **HARDWARE AND SOFTWARE** A computing system is composed of hardware and software. Hardware consists of physical components, while software provides instructions for the system. These instructions are represented in a form that a computer can understand. **TROUBLESHOOTING** Computing systems might not work as expected because of hardware or software problems. Clearly describing a problem is the first step toward finding a solution.	**NETWORK COMMUNICATION AND ORGANIZATION** Computer networks can be used to connect people to other people, places, information, and ideas. The Internet enables people to connect with others world-wide through many different points of connection. **CYBERSECURITY** 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.	**COLLECTION** Everyday digital devices collect and display data over time. The collection and use of data about individuals and the world around them is a routine part of life and influences how people live. **STORAGE** Computers store data that can be retrieved later. Identical copies of data can be made and stored in multiple locations for a variety of reasons, such as to protect against loss. **VISUALIZATION AND TRANSFORMATION** Data can be displayed for communication in many ways. People use computers to transform data into new forms, such as graphs and charts. **INFERENCE AND MODELS** Data can be used to make inferences or predictions about the world. Inferences, statements about something that cannot be readily observed, are often based on observed data. Predictions, statements about future events, are based on patterns in data and can be made by looking at data visualizations, such as charts and graphs.	**ALGORITHMS** People follow and create processes as part of daily life. Many of these processes can be expressed as algorithms that computers can follow. **VARIABLES** Information in the real world can be represented in computer programs. Programs store and manipulate data, such as numbers, words, colors, and images. The type of data determines the actions and attributes associated with it. **CONTROL** Computers follow precise sequences of instructions that automate tasks. Program execution can also be nonsequential by repeating patterns of instructions and using events to initiate instructions. **MODULARITY** Complex tasks can be broken down into simpler instructions, some of which can be broken down even further. Likewise, instructions can be combined to accomplish complex tasks. **PROGRAM DEVELOPMENT** People develop programs collaboratively and for a purpose, such as expressing ideas or addressing problems.	**CULTURE** Computing technology has positively and negatively changed the way people live and work. Computing devices can be used for entertainment and as productivity tools, and they can affect relationships and lifestyles. **SOCIAL INTERACTIONS** Computing has positively and negatively changed the way people communicate. People can have access to information and each other instantly, anywhere, and at any time, but they are at the risk of cyberbullying and reduced privacy. **SAFETY, LAW, AND ETHICS** People use computing technology in ways that can help or hurt themselves or others. Harmful behaviors, such as sharing private information and interacting with strangers, should be recognized and avoided.

By the end of Grade 5

Computing Systems	Networks and the Internet	Data and Analysis	Algorithms and Programming	Impacts of Computing
DEVICES 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. **HARDWARE AND SOFTWARE** Hardware and software work together as a system to accomplish tasks, such as sending, receiving, processing, and storing units of information as bits. Bits serve as the basic unit of data in computing systems and can represent a variety of information. **TROUBLESHOOTING** Computing systems share similarities, such as the use of power, data, and memory. Common troubleshooting strategies, such as checking that power is available, checking that physical and wireless connections are working, and clearing out the working memory by restarting programs or devices, are effective for many systems.	**NETWORK COMMUNICATION AND ORGANIZATION** Information needs a physical or wireless path to travel to be sent and received, and some paths are better than others. Information is broken into smaller pieces, called packets, that are sent independently and reassembled at the destination. Routers and switches are used to properly send packets across paths to their destinations. **CYBERSECURITY** Information can be protected using various security measures. These measures can be physical and/or digital.	**COLLECTION** People select digital tools for the collection of data based on what is being observed and how the data will be used. For example, a digital thermometer is used to measure temperature and a GPS sensor is used to track locations. **STORAGE** Different software tools used to access data may store the data differently. The type of data being stored and the level of detail represented by that data affect the storage requirements. **VISUALIZATION AND TRANSFORMATION** People select aspects and subsets of data to be transformed, organized, clustered, and categorized to provide different views and communicate insights gained from the data. **INFERENCE AND MODELS** The accuracy of inferences and predictions is related to how realistically data is represented. Many factors influence the accuracy of inferences and predictions, such as the amount and relevance of data collected.	**ALGORITHMS** Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others. **VARIABLES** Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data. **CONTROL** Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions. **MODULARITY** Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created.	**CULTURE** The development and modification of computing technology is driven by people's needs and wants and can affect groups differently. Computing technologies influence, and are influenced by, cultural practices. **SOCIAL INTERACTIONS** Computing technology allows for local and global collaboration. By facilitating communication and innovation, computing influences many social institutions such as family, education, religion, and the economy. **SAFETY, LAW, AND ETHICS** Ethical complications arise from the opportunities provided by computing. The ease of sending and receiving copies of media on the Internet, such as video, photos, and music, creates the opportunity for unauthorized use, such as online piracy, and disregard of copyrights, such as lack of attribution.

Table continued on next page

Table continued from previous page

			PROGRAM DEVELOPMENT People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.	

By the end of Grade 8

Computing Systems	Networks and the Internet	Data and Analysis	Algorithms and Programming	Impacts of Computing
DEVICES The interaction between humans and computing devices presents advantages, disadvantages, and unintended consequences. The study of human–computer interaction can improve the design of devices and extend the abilities of humans. **HARDWARE AND SOFTWARE** Hardware and software determine a computing system's capability to store and process information. The design or selection of a computing system involves multiple considerations and potential tradeoffs, such as functionality, cost, size, speed, accessibility, and aesthetics. **TROUBLESHOOTING** Comprehensive troubleshooting requires knowledge of how computing devices and components work and interact. A systematic process will identify the source of a problem, whether within a device or in a larger system of connected devices.	**NETWORK COMMUNICATION AND ORGANIZATION** Computers send and receive information based on a set of rules called protocols. Protocols define how messages between computers are structured and sent. Considerations of security, speed, and reliability are used to determine the best path to send and receive data. **CYBERSECURITY** The information sent and received across networks can be protected from unauthorized access and modification in a variety of ways, such as encryption to maintain its confidentiality and restricted access to maintain its integrity. Security measures to safeguard online information proactively address the threat of breaches to personal and private data.	**COLLECTION** People design algorithms and tools to automate the collection of data by computers. When data collection is automated, data is sampled and converted into a form that a computer can process. For example, data from an analog sensor must be converted into a digital form. The method used to automate data collection is influenced by the availability of tools and the intended use of the data. **STORAGE** Applications store data as a representation. Representations occur at multiple levels, from the arrangement of information into organized formats (such as tables in software) to the physical storage of bits. The software tools used to access information translate the low-level representation of bits into a form understandable by people. **VISUALIZATION AND TRANSFORMATION** Data can be transformed to remove errors, highlight or expose relationships, and/or make it easier for computers to process.	**ALGORITHMS** Algorithms affect how people interact with computers and the way computers respond. People design algorithms that are generalizable to many situations. Algorithms that are readable are easier to follow, test, and debug. **VARIABLES** Programmers create variables to store data values of selected types. A meaningful identifier is assigned to each variable to access and perform operations on the value by name. Variables enable the flexibility to represent different situations, process different sets of data, and produce varying outputs. **CONTROL** Programmers select and combine control structures, such as loops, event handlers, and conditionals, to create more complex program behavior.	**CULTURE** Advancements in computing technology change people's everyday activities. Society is faced with tradeoffs due to the increasing globalization and automation that computing brings. **SOCIAL INTERACTIONS** People can organize and engage around issues and topics of interest through various communication platforms enabled by computing, such as social networks and media outlets. These interactions allow issues to be examined using multiple viewpoints from a diverse audience. **SAFETY, LAW, AND ETHICS** There are tradeoffs between allowing information to be public and keeping information private and secure. People can be tricked into revealing personal information when more public information is available about them online.

Table continued on next page

Table continued from previous page

		INFERENCE AND MODELS Computer models can be used to simulate events, examine theories and inferences, or make predictions with either few or millions of data points. Computer models are abstractions that represent phenomena and use data and algorithms to emphasize key features and relationships within a system. As more data is automatically collected, models can be refined.	**MODULARITY** Programs use procedures to organize code, hide implementation details, and make code easier to reuse. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability. **PROGRAM DEVELOPMENT** People design meaningful solutions for others by defining a problem's criteria and constraints, carefully considering the diverse needs and wants of the community, and testing whether criteria and constraints were met.	

By the end of Grade 12

Computing Systems	Networks and the Internet	Data and Analysis	Algorithms and Programming	Impacts of Computing
DEVICES Computing devices are often integrated with other systems, including biological, mechanical, and social systems. These devices can share data with one another. The usability, dependability, security, and accessibility of these devices, and the systems they are integrated with, are important considerations in their design as they evolve. **HARDWARE AND SOFTWARE** Levels of interaction exist between the hardware, software, and user of a computing system. The most common levels of software that a user interacts with include system software and applications. System software controls the flow of information between hardware components used for input, output, storage, and processing. **TROUBLESHOOTING** Troubleshooting complex problems involves the use of multiple sources when researching, evaluating, and implementing potential solutions. Troubleshooting also relies on experience, such as when people	**NETWORK COMMUNICATION AND ORGANIZATION** Network topology is determined, in part, by how many devices can be supported. Each device is assigned an address that uniquely identifies it on the network. The scalability and reliability of the Internet are enabled by the hierarchy and redundancy in networks. **CYBERSECURITY** Network security depends on a combination of hardware, software, and practices that control access to data and systems. The needs of users and the sensitivity of data determine the level of security implemented.	**COLLECTION** Data is constantly collected or generated through automated processes that are not always evident, raising privacy concerns. The different collection methods and tools that are used influence the amount and quality of the data that is observed and recorded. **STORAGE** Data can be composed of multiple data elements that relate to one another. For example, population data may contain information about age, gender, and height. People make choices about how data elements are organized and where data is stored. These choices affect cost, speed, reliability, accessibility, privacy, and integrity. **VISUALIZATION AND TRANSFORMATION** People transform, generalize, simplify, and present large data sets in different ways to influence how other people interpret and understand the underlying information. Examples include visualization, aggregation, rearrangement, and application of mathematical operations.	**ALGORITHMS** People evaluate and select algorithms based on performance, reusability, and ease of implementation. Knowledge of common algorithms improves how people develop software, secure data, and store information. **VARIABLES** Data structures are used to manage program complexity. Programmers choose data structures based on functionality, storage, and performance tradeoffs. **CONTROL** Programmers consider tradeoffs related to implementation, readability, and program performance when selecting and combining control structures. **MODULARITY** Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose. These modules can be procedures within a program; combinations of data and procedures; or independent, but interrelated, programs. Modules allow for better management of complex tasks.	**CULTURE** The design and use of computing technologies and artifacts can improve, worsen, or maintain inequitable access to information and opportunities. **SOCIAL INTERACTIONS** Many aspects of society, especially careers, have been affected by the degree of communication afforded by computing. The increased connectivity between people in different cultures and in different career fields has changed the nature and content of many careers. **SAFETY, LAW, AND ETHICS** Laws govern many aspects of computing, such as privacy, data, property, information, and identity. These laws can have beneficial and harmful effects, such as expediting or delaying advancements in computing and protecting or infringing upon people's rights. International differences in laws and ethics have implications for computing.

Table continued on next page

Table continued from previous page

recognize that a problem is similar to one they have seen before or adapt solutions that have worked in the past.		**INFERENCE AND MODELS** The accuracy of predictions or inferences depends upon the limitations of the computer model and the data the model is built upon. The amount, quality, and diversity of data and the features chosen can affect the quality of a model and ability to understand a system. Predictions or inferences are tested to validate models.	**PROGRAM DEVELOPMENT** Diverse teams can develop programs with a broad impact through careful review and by drawing on the strengths of members in different roles. Design decisions often involve tradeoffs. The development of complex programs is aided by resources such as libraries and tools to edit and manage parts of the program. Systematic analysis is critical for identifying the effects of lingering bugs.	