Ohio’s Learning Standards

Computer Science

ADOPTED DECEMBER 2018

Ohio | Department of Education

Table of Contents

Introduction to Ohio's Learning Standards for Computer Science

Substitute House Bill Number 170 took effect in March 2018, requiring the State Board of Education of Ohio to adopt standards and a model curriculum for grade K-12 instruction in computer science. A team of Ohio educators came together to develop and write the computer science standards and model curriculum, and the State Board adopted these in December 2018.

Ohio's Standards in Computer Science are fully aligned to Ohio's five-year strategic plan for education, *Each Child, Our Future*. The strategic plan acknowledges a major education policy shift around technology. A student's ability to use technology strategically is now identified as foundational and just as important as mathematics and English language arts, from which all other learning is built.

GUIDING ASSUMPTIONS

The team of Ohio educators that developed the standards and model curriculum had a clear goal – to encourage districts and educators to give all Ohio students opportunities to learn computer science. Beginning in the earliest grades and continuing through grade 12, Ohio's students will develop a foundation of computer science knowledge and gain experiences in computational thinking and problem solving to become creators and innovators of computing technology. Ohio's Computer Science Standards and Model Curriculum will give students experiences that help them discover and take part in a world continually influenced by technology and to understand the role of computing in that world.

OVERVIEW OF THE COMPUTER SCIENCE STANDARDS CONTENT

The standards will support a progression of learning in each core concept or strand to provide computer science experiences for all Ohio students. The K-8 standards integrate computer science into instruction across subject areas including mathematics, science, history, English language arts, fine arts, world language and career and technology courses. The high school computer science standards provide both foundational and advanced opportunities districts can use to design as separate courses or, when appropriate, integrate into other disciplines.

Ohio's Computer Science Standards and Model Curriculum are organized in the following strands:

- **Computing Systems** - Addresses how devices, including hardware and software, interact to accomplish tasks and how students can troubleshoot computing systems when they do not work as intended.
- **Networks and the Internet** - Addresses how networks connect to share information and resources and how students can apply cybersecurity concepts to protect information.
- **Data and Analysis** - Addresses how data can be collected and stored; analyzed and communicated; and used to make more accurate predictions.
- **Algorithmic Thinking and Programming** - Addresses program development, including the use of algorithms, variables, control structures and modules.
- **Impacts of Computing** – Addresses computing's influence on our world by examining the relationship between computing and culture, computing's impact on social interaction, and legal and ethical implications of computing.

Computational Thinking is a problem-solving process that students use to engage with concepts in the computer science standards. This thinking involves formulating problems in a way that can be carried out by a computer. Using computational thinking to solve a problem includes breaking down the problem into manageable parts; recognizing patterns; excluding irrelevant details to abstract or identify general principles that generate these patterns; and developing step-by-step sequences or algorithms to solve the problem and similar problems. Computational thinking can be applied with or without computers, for example, through "unplugged" activities. While computational thinking is a focus in computer science, it also is used in content areas beyond computer science.

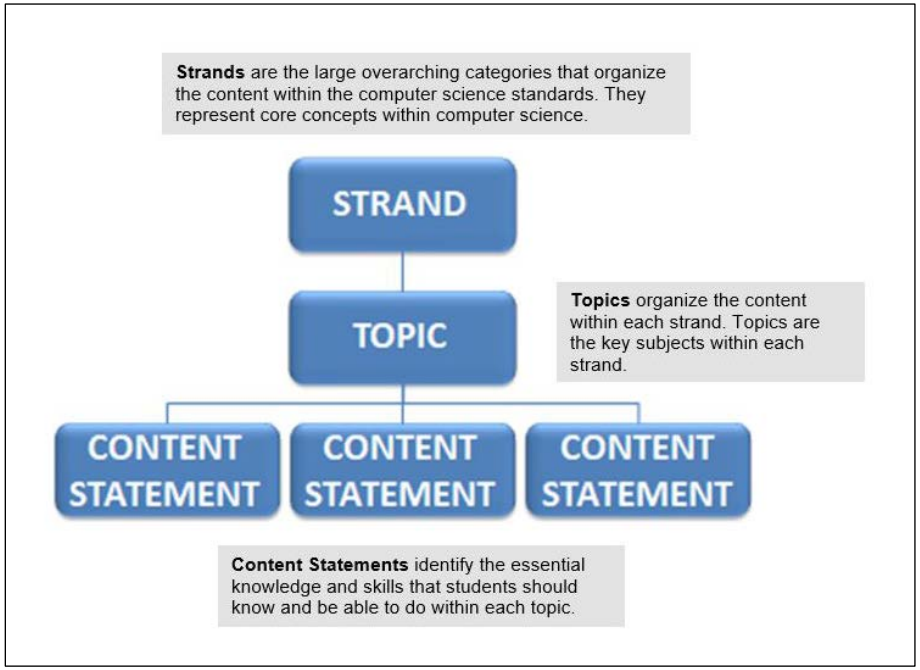
It is important that computer science not be confused with other aspects and uses of computer technology in schools, including:

- **Computer literacy "**refers to the general use of computers and programs, such as productivity software." Examples of computer literacy include performing an internet search and creating a digital presentation.
- **Educational (computer) technology "**applies computer literacy to school subjects. For example, students in an English class can use a web-based application to collaboratively create, edit and store an essay online."
- **Digital citizenship "**refers to the appropriate and responsible use of technology, such as choosing an appropriate password and keeping it secure."
- **Information technology "**often overlaps with computer science but is mainly focused on industrial applications of computer science, such as installing [and operating] software rather than creating it. Information technology professionals often have a background in computer science."

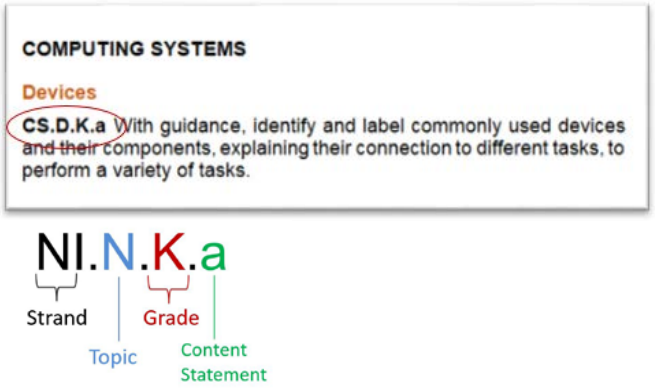
(K-12 Computer Science Framework, 2016, pp.13-14)

OVERVIEW OF THE COMPUTER SCIENCE STANDARDS FRAMEWORK

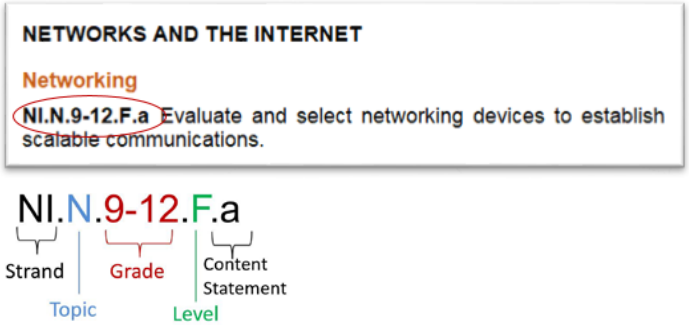
Ohio's Computer Science Standards are organized by strands, topics and content statements.



Kindergarten through Grade 8 - Content statements are organized *by grade level*. Below is an example of a content statement for kindergarten and its corresponding content statement code. This content statement addresses the topic of *Devices* within the *Computing Systems* strand.



Grades 9-12 - Content statements are organized *by grade band* into *two levels – Foundational and Advanced*. See an example of a content statement for high school and its corresponding content statement code below. This content statement addresses the topic of *Networking* within the *Networks and the Internet* strand, at the Foundational Level.



Equity and Computer Science[1]

COMPUTER SCIENCE FOR ALL

To help realize the vision *of computer science for all students,* equity must be at the forefront of the state's efforts to implement the computer science standards. *Equity is more than whether classes are available. It includes how those classes are taught, how students are recruited and how the classroom culture supports diverse learners and promotes retention***.** When equity exists, schools expect academic success for every student and makes that success accessible to every student. The result of such equity is a classroom of diverse students based on factors such as race, gender, disability, socioeconomic status and English language proficiency. All these students have high expectations and feel empowered to learn.

Computer science faces intense challenges related to access, opportunity and culture.

- The 2015 National Assessment of Educational Progress (NAEP) survey showed that only 44 percent of 12th grade students attend high schools that offer any computer science courses (Change the Equation, 2016). This data showed that students with the least access are Black, Latino and Native American; from lower income backgrounds; and from rural areas.

Even when computer science courses are available, there are wide gaps in participation and the level of instruction.

- For the 2015 Advanced Placement (AP) Computer Science exam, only 21.9 percent of students were female, 3.9 percent were Black or African American, 9 percent were Hispanic or Latino and 0.4 percent were American Indian (College Board, 2016). The potential impact of these gaps in participation is illustrated in the statistic that females who take high school AP Computer Science are 10 times more likely to major in computer science in college than students who do not take this course. Similarly, African American students are seven times more likely, and Hispanic students 8.5 times more likely, to major in computer science when they have taken high school AP Computer Science (Morgan & Klaric, 2007).
- Especially in schools with large numbers of African American and Latino students, computer classes too commonly offer only basic, rudimentary user skills rather than engaging students in the problem-solving and computational thinking practices that form the foundation of computer science (Margolis et al., 2012).

The lack of representation in computer science after K–12 reflects the lack of access and participation in grades K–12. In 2015, only 24.7 percent of workers employed in computer and mathematical occupations were female. Only 8.6 percent were Black or African American, and only 6.8 percent were Hispanic or Latino (Bureau of Labor Statistics, 2015).

EFFORTS TO INCREASE EQUITY

Even when schools have made computer science courses available to students, inequity can be perpetuated at the classroom level. Educators can work to ensure equity through changes in curriculum, instruction and classroom culture.

- Educators can reach *students with disabilities* using learning accommodations, curricular modifications and established techniques for differentiated instruction. For example, the Quorum programming language accommodates students with visual impairments by enabling the programming language to be read by computer screen readers (Quorum, 2019). Recent research shows ways to use Universal Design for Learning (UDL) to develop and refine introductory computer science experiences for a wider range of learners (Hansen et al., 2016). Educators also can apply instructional strategies used in other content areas to support struggling learners and

[1] The "Equity and Computer Science" section has been modified from chapter two of The K-12 Computer Science Framework, "Equity in Computer Science Education." (K–12 Computer Science Framework. (2016). Retrieved from http://www.k12cs.org.) This work is licensed under Creative Commons (CC BY-NC-SA 4.0).

students with disabilities. For example, if verbal prompting helps in math instruction, it will likely help in computer science instruction (Snodgrass, Israel, & Reese, 2016).

- A variety of approaches make programming more accessible *to young learners and beginners.* Visual, block-based programming languages allow students to program without the obstacle of syntax errors (errors in typing commands) found in traditional text-based languages. Programming environments on tablets have made programming even more accessible to younger children by reducing the number of available commands and the amount of reading required to navigate the options (Strawhacker & Bers, 2014).
- To address a *lack of computer and internet access*, educators can help students learn many computer science topics, such as algorithmic thinking, searching, sorting and logic through "unplugged" activities.
- To reach *females and underrepresented minorities*, teachers can use strategies to work against issues such as the threat of stereotyping or bias. For example, stereotype threats can be mitigated by altering the wording of test questions to be gender-neutral and using examples that are equally relevant to both males and females (Kumar, 2012). It also is important for students to have diverse role models in the field so they can imagine themselves as a computer scientists. Role models also help dispel stereotypes of how computer scientists look and act (Goode, 2008).

Below, are other practices that teachers can adopt and adapt to change classroom culture and broaden participation in computer science:

- Connect computer science to concepts that motivate children, like fairness and social justice (Denner et al., 2015).
- Practice culturally relevant pedagogy to tie computer science to students' experiences, culture and interests (Margolis et al., 2014). Designing projects and instruction to be socially relevant and meaningful for diverse students helps them "build personal relationships with CS concepts and applications -- an important process for discovering the relevance of CS for their own lives." (Margolis et. al, 2012, p. 76)
- Reflect on beliefs and actions to address stereotypes among students and teachers (Margolis et al., 2014).

EQUITY AND THE COMPUTER SCIENCE STANDARDS

The computer science standards reflect the writing team's considerations on equity. The standards describe concepts and skills all students can benefit from, regardless of whether they go on to postsecondary computer-science education or a career in computer science.

Equity is woven into the computer science concepts and skills across grade levels. This is especially apparent in the core concept or strand involving *Impacts of Computing*. Here, students examine the social implications of the digital world, including their impacts on equity and access to computing. Specific content statements address equity directly. For example, in grade 3, students identify diverse user needs and "how computing devices have built-in features to increase accessibility to all users." In grade 7, students "evaluate various technologies to identify issues of bias and accessibility." Students in grade 8 build on prior learning to work against existing inequities; they propose guidelines "to positively impact bias and accessibility in the design of future technologies."

As students design computational products, they engage in computer science practices that also directly involve consideration of equity, inclusion and diversity. Students foster inclusion as they develop products that "include the unique perspectives of others" and "address the needs of diverse end users." Students encourage diversity through working in teams "with individuals possessing diverse perspectives." Involving students in such practices stresses the need to practice equity when doing computer science. Through such practices, students can see the benefit of, for example, considering the products they develop from the perspectives of a diverse group of end-users, such as those with visual impairments and English language learners.

Kindergarten

COMPUTING SYSTEMS

Devices

CS.D.K.a With guidance, identify and label commonly used devices and their components, explaining their connection to different tasks, to perform a variety of tasks.

Hardware and Software

CS.HS.K.a With guidance and support, identify and use hardware and software necessary for accomplishing a task.

Troubleshooting

CS.T.K.a With guidance and support, use problem solving strategies to troubleshoot a problem.

NETWORKS AND THE INTERNET

Networking

NI.N.K.a With guidance and support, create a list of ways information can be shared electronically to gain a deeper understanding of how information is transmitted (e.g., email, social media).

Cybersecurity

NI.C.K.a With guidance and support, identify and use secure practices (e.g., passwords) to protect private information.

DATA AND ANALYSIS

Data Collection and Storage

DA.DCS.K.a Identify data to collect and sort.

DA.DCS.K.b With guidance and support, demonstrate how data can be collected and stored in a variety of ways.

Visualization and Communication

DA.VC.K.a With guidance, organize and present data in various formats to make observations.

Inference and Modeling

DA.IM.K.a With guidance, create a model of an object or process to identify patterns.

ALGORITHMIC THINKING AND PROGRAMMING

Algorithms

ATP.A.K.a With guidance and support, model a real-world process by constructing and following step-by-step directions (i.e., algorithms) to complete tasks.

Variables and Data Representation

ATP.VDR.K.a Recognize that a group of items (e.g., numbers, symbols or pictures) can be used to represent data.

Control Structures

ATP.CS.K.a With guidance and support, model a sequence of instructions (i.e., program) with a beginning, middle and end to solve a problem or express an idea.

Program Development

ATP.PD.K.a With guidance and support, plan or create an artifact to illustrate thoughts, ideas and problems in a sequential (step-by-step) manner (e.g., story map, storyboard, sequential graphic organizer).

IMPACTS OF COMPUTING

Culture

IC.Cu.K.a With guidance and support, identify technologies that impact one's own everyday life.

IC.Cu.K.b With guidance and support, recognize different ways computing devices are used regularly to understand technology's impact on one's own daily life.

Social Interactions

IC.SI.K.a With guidance and support, identify and use safe and responsible behaviors concerning information and technology.

Safety, Law and Ethics

IC.SLE.K.a With guidance, discuss appropriate uses of technology to support informed decisions.

Grade 1

COMPUTING SYSTEMS

Devices

CS.D.1.a Operate commonly used devices and their components to perform a variety of tasks.

Hardware and Software

CS.HS.1.a With guidance, describe and use hardware and software necessary for accomplishing a task.

Troubleshooting

CS.T.1.a With guidance, use problem solving strategies to troubleshoot a problem.

NETWORKS AND THE INTERNET

Networking

NI.N.1.a Create a list of ways information can be shared electronically to gain a deeper understanding of how information is transmitted (e.g., email, social media).

NI.N.1.b Recognize that computing devices can be connected to retrieve information from the global community.

Cybersecurity

NI.C.1.a Identify and use secure practices (e.g., passwords) to protect private information.

DATA AND ANALYSIS

Data Collection and Storage

DA.DCS.1.a With guidance, collect and organize data to retrieve for later use.

DA.DCS.1.b With guidance, demonstrate how data can be collected and stored in a variety of ways.

Visualization and Communication

DA.VC.1.a Organize and present data in various formats to make observations.

Inference and Modeling

DA.IM.1.a Create and explain a model of an object or process that includes patterns and key elements.

ALGORITHMIC THINKING AND PROGRAMMING

Algorithms

ATP.A.1.a With guidance, model a real-world process by constructing and following step-by-step directions (i.e., algorithms) to complete tasks.

Variables and Data Representation

ATP.VDR.1.a Categorize a group of items (e.g., numbers, symbols or pictures) based on the attributes or actions of each item, with or without a computing device.

Control Structures

ATP.CS.1.a With guidance, model a sequence of instructions (i.e., program) that includes repetition (i.e., loops) to solve a problem or express ideas.

Modularity

ATP.M.1.a With guidance, break down (i.e., decompose) a series of steps and separate the necessary from the unnecessary steps to create a precise sequence of instructions to solve a problem or express an idea.

Program Development

ATP.PD.1.a With guidance, plan and create an artifact to illustrate thoughts, ideas and problems in a sequential (step-by-step) manner (e.g., story map, storyboard, sequential graphic organizer).

ATP.PD.1.b With guidance, identify and fix (i.e., debug) a multi-step process that includes sequencing.

IMPACTS OF COMPUTING

Culture

IC.Cu.1.a Discuss different technologies and their impact on everyday life.

IC.Cu.1.b Identify how people use and are impacted by many types of technologies in their daily work and personal lives.

Social Interactions

IC.SI.1.a With guidance, describe safe and responsible behaviors for the use of information and technology.

Safety, Law and Ethics

IC.SLE.1.a With guidance, discuss appropriate and ethical uses of technology to guide informed decisions.

Grade 2

COMPUTING SYSTEMS

Devices

CS.D.2.a Select and operate commonly used devices to perform a variety of tasks.

Hardware and Software

CS.HS.2.a Select and use hardware and software necessary for accomplishing a task.

Troubleshooting

CS.T.2.a Use problem solving strategies to troubleshoot a problem.

NETWORKS AND THE INTERNET

Networking

NI.N.2.a Describe how information can be communicated electronically to gain a deeper understanding of how information is transmitted (e.g., email, social media).

NI.N.2.b Use computing devices that are connected to share and receive information from the global community.

Cybersecurity

NI.C.2.a Explain and demonstrate secure practices (e.g., creating strong passwords) to protect private information.

DATA AND ANALYSIS

Data Collection and Storage

DA.DCS.2.a Collect and organize data to store, retrieve and modify.

DA.DCS.2.b Manipulate data to perform various tasks.

Visualization and Communication

DA.VC.2.a Organize, analyze and present data in various formats.

Inference and Modeling

DA.IM.2.a Interpret and analyze data, graphs, models or charts.

ALGORITHMIC THINKING AND PROGRAMMING

Algorithms

ATP.A.2.a Model a real-world process by constructing and following step-by-step instructions (i.e., algorithms) to complete tasks.

Variables and Data Representation

ATP.VDR.2.a Construct a model that shows the way programs store and manipulate data by using numbers or other symbols to represent information.

Control Structures

ATP.CS.2.a Develop a program that uses sequencing and repetition (i.e., loops) to solve a problem or express ideas.

Modularity

ATP.M.2.a Break down (i.e., decompose) a series of steps and separate the necessary from the unnecessary steps to create a precise sequence of instructions to solve a problem or express an idea.

Program Development

ATP.PD.2.a Plan and create an artifact to illustrate thoughts, ideas and problems in a sequential (step-by-step) manner (e.g., story map, storyboard, sequential graphic organizer).

ATP.PD.2.b Identify and fix (i.e., debug) a multi-step process that includes sequencing.

IMPACTS OF COMPUTING

Culture

IC.Cu.2.a Compare and contrast how the use of technology has changed to understand its impact on everyday life.

IC.Cu.2.b Describe the ways people use technologies in their daily work and personal lives to understand technology's impact on one's community.

Social Interactions

IC.SI.2.a Compare and contrast safe and responsible behaviors to those that are not when using information and technology.

Safety, Law and Ethics

IC.SLE.2.a Discuss appropriate and ethical uses of technology to guide informed decisions.

Grade 3

COMPUTING SYSTEMS

Devices

CS.D.3.a Explore common components (i.e., parts) of a computing system and their function to understand and describe the role they play in a computer system.

Hardware and Software

CS.HS.3.a Identify and use digital learning tools/devices to support planning, implementing and reflecting upon a defined task.

Troubleshooting

CS.T.3.a Apply troubleshooting strategies given problems and solutions to resolve hardware and software problems.

NETWORKS AND THE INTERNET

Networking

NI.N.3.a Describe how communication occurs when information is sent and received over physical or wireless paths to explain communication systems (e.g., sending an email or visiting a website).

NI.N.3.b Recognize that every device on a network has a unique identification to share or receive information from the global community.

Cybersecurity

NI.C.3.a Explore digital safety concepts in order to explain that information can be both public and private, to determine what information can safely be shared and to know how to use passwords to protect information.

DATA AND ANALYSIS

Data Collection and Storage

DA.DCS.3.a Collect quantitative data over time from multiple sources to perform various tasks.

DA.DCS.3.b Identify different types of information to store in different formats.

Visualization and Communication

DA.VC.3.a Create a chart or graph to inform a target audience about observations and data collected.

Inference and Modeling

DA.IM.3.a Utilize data to make predictions and discuss whether there is adequate data to make reliable predictions.

ALGORITHMIC THINKING AND PROGRAMMING

Algorithms

ATP.A.3.a Construct and reflect on errors in an algorithm to accomplish a given task.

Variables and Data Representation

ATP.VDR.3.a Define and identify a variable, a placeholder for storing a value, to understand how it is used in a multi-step process (i.e., algorithm).

Control Structures

ATP.CS.3.a Create a program using sequences, events, loops and conditionals to solve a problem.

Modularity

ATP.M.3.a Decompose (i.e., break down) the steps needed or not needed (i.e., abstraction) into precise sequences of instructions to design an algorithm.

Program Development

ATP.PD.3.a Use a design process to plan the development of a program that solves problems.

ATP.PD.3.b Using a given program known to contain errors, identify and debug errors to ensure it works.

IMPACTS OF COMPUTING

Culture

IC.Cu.3.a Identify computing technologies that have changed the world and express how those technologies influence and are influenced by cultural practice.

IC.Cu.3.b Identify how computing devices have built-in features to increase accessibility to all users.

Social Interactions

IC.SI.3.a Collaborate and consider diverse perspectives to improve digital artifacts.

Safety, Law and Ethics

IC.SLE.3.a Use public domain or Creative Commons media, and refrain from copying or using material created by others without permission.

IC.SLE.3.b Determine whether information should be shared or kept private to protect student identity.

IC.SLE.3.c Communicate the importance of information security to protect one's own digital footprint.

Grade 4

COMPUTING SYSTEMS

Devices

CS.D.4.a Explore external components (i.e., parts) of a computing system and their function to understand and describe the role they play in a computer system.

Hardware and Software

CS.HS.4.a Select and use digital learning tools/devices to support planning, implementing and reflecting upon a defined task.

Troubleshooting

CS.T.4.a Diagnose problems and select an appropriate solution from a list of problems and solutions to resolve hardware and software issues.

NETWORKS AND THE INTERNET

Networking

NI.N.4.a Describe how information is broken down to be transmitted over a network to help students gain a better understanding of the internet and networks.

NI.N.4.b Describe network addresses, names and rules (i.e., protocols) to share or receive information from the global community.

Cybersecurity

NI.C.4.a Describe what information should be protected and the importance of a secure password to protect information.

DATA AND ANALYSIS

Data Collection and Storage

DA.DCS.4.a Gather and organize multiple quantitative data elements using a tool to perform various tasks.

DA.DCS.4.b Identify techniques and formats to store, process and retrieve different types of information.

Visualization and Communication

DA.VC.4.a Organize data into subsets to provide different views or commonalities and present insights gained using visual or other types of representations.

Inference and Modeling

DA.IM.4.a Utilize data to make predictions and discuss whether there is adequate data to make reliable predictions.

ALGORITHMIC THINKING AND PROGRAMMING

Algorithms

ATP.A.4.a Construct and refine an algorithm to accomplish a given task.

Variables and Data Representation

ATP.VDR.4.a Identify and use a variable, a placeholder for storing a value, to understand how it works in a multi-step process (i.e., algorithm).

Control Structures

ATP.CS.4.a Create a program using sequences, events, loops and conditionals to solve a problem.

Modularity

ATP.M.4.a Decompose (i.e., break down) the steps needed or not needed (i.e., abstraction) into precise sequences of instructions to design an algorithm.

Program Development

ATP.PD.4.a Use a design process to plan and develop a program that addresses a multi-step problem.

ATP.PD.4.b Using guided questions, work through a program to identify errors and discuss possible solutions to repair the program.

IMPACTS OF COMPUTING

Culture

IC.Cu.4.a List examples of computing technologies that have changed the global community to express how those technologies influenced and are influenced by cultural practice.

IC.Cu.4.b Identify and anticipate diverse user needs to increase accessibility to all users.

Social Interactions

IC.SI.4.a Collaborate and consider diverse perspectives to improve digital artifacts.

Safety, Law and Ethics

IC.SLE.4.a Use public domain or Creative Commons media, and refrain from copying or using material created by others without permission.

IC.SLE.4.b Explain why information should be shared or kept private to protect student identity.

IC.SLE.4.c Communicate the importance of protecting your digital footprint.

Grade 5

COMPUTING SYSTEMS

Devices

CS.D.5.a Explore the internal parts of the computing system and their function to understand and describe the role they play in a computer system.

Hardware and Software

CS.HS.5.a Evaluate digital learning tools/devices to support planning, implementing and reflecting across curricular areas.

Troubleshooting

CS.T.5.a Diagnose problems and develop strategies to resolve technology issues.

NETWORKS AND THE INTERNET

Networking

NI.N.5.a Model how information is broken down to be transmitted and then reassembled to help students gain a better understanding of the internet and networks.

NI.N.5.b Apply knowledge of network addresses, names and rules (i.e., protocols) to discuss real-world scenarios.

Cybersecurity

NI.C.5.a Demonstrate password creation techniques to develop and use a strong password used on personal accounts.

DATA AND ANALYSIS

Data Collection and Storage

DA.DCS.5.a Gather and organize multiple quantitative data elements using a tool to perform various tasks.

DA.DCS.5.b Compare and contrast file formats to demonstrate the advantages and disadvantages of each.

Visualization and Communication

DA.VC.5.a Organize and present collected data using visual or other types of representations to highlight relationships and support a claim.

Inference and Modeling

DA.IM.5.a Utilize data to propose cause and effect relationships and predict outcomes.

ALGORITHMIC THINKING AND PROGRAMMING

Algorithms

ATP.A.5.a Evaluate a multi-step process to diagram the proper steps to solve a problem.

Variables and Data Representation

ATP.VDR.5.a Create a variable, a placeholder for storing a value, to understand how it is used in a multi-step process (i.e., algorithm).

Control Structures

ATP.CS.5.a Create a program using sequences, events, loops and conditionals to solve a problem.

Modularity

ATP.M.5.a Decompose (i.e., break down) the steps needed or not needed (i.e., abstraction) into precise sequences of instructions to design an algorithm.

ATP.M.5.b With grade appropriate complexity, modify, remix or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.

Program Development

ATP.PD.5.a Use a design process to plan and develop a program that includes multiple steps and end user preferences.

ATP.PD.5.b Using guided questions, work through a program to identify errors and discuss possible solutions to repair the program.

IMPACTS OF COMPUTING

Culture

IC.Cu.5.a Explain how computing technologies have changed the global community and express how those technologies influence and are influenced by cultural practices.

IC.Cu.5.b Develop, test and refine digital artifacts to improve accessibility and usability.

Social Interactions

IC.SI.5.a Collaborate and consider diverse perspectives to improve digital artifacts.

Safety, Law and Ethics

IC.SLE.5.a Use public domain or Creative Commons media, and refrain from copying or using material created by others without permission.

IC.SLE.5.b Communicate the effects of sharing personal information on the safety of student identity to determine how to protect students.

IC.SLE.5.c Evaluate the need to keep personal information secure and protect the digital footprint.

Grade 6

COMPUTING SYSTEMS

Devices

CS.D.6.a Identify the benefits and limitations of a given computing device's functions (including individual components) to explain how the functions and components work together to create the computing system.

Hardware and Software

CS.HS.6.a Identify ways that hardware and software work together as a system to collect and exchange data.

Troubleshooting

CS.T.6.a Use a systematic process to identify and evaluate the source of a routine computing problem. Select the best solution to solve the computing problem and communicate the solution to others.

NETWORKS AND THE INTERNET

Networking

NI.N.6.a Identify the role of hardware components to understand the infrastructure of networks and the internet (including cloud servers).

NI.N.6.b Identify protocols (i.e., rules) and explain why they are used to transmit data across networks and the internet.

Cybersecurity

NI.C.6.a Identify cybersecurity concerns and measures needed to protect electronic information.

NI.C.6.b Identify the different types of malware to understand threats to data security.

NI.C.6.c Identify ways to protect private information.

DATA AND ANALYSIS

Data Collection and Storage

DA.DCS.6.a Identify and use an appropriate digital data collection tool to compile information.

DA.DCS.6.b Select and utilize appropriate file formats to organize collected data.

DA.DCS.6.c Utilize a file structure to logically organize data to support individual and collaborative work.

Visualization and Communication

DA.VC.6.a Identify and label patterns in models or representations to infer connections between data sets.

DA.VC.6.b Create a spreadsheet utilizing formulas, functions and graphs to represent and analyze data.

Inference and Modeling

DA.IM.6.a Identify and utilize data sets to support or refute a hypothesis.

ALGORITHMIC THINKING AND PROGRAMMING

Algorithms

ATP.A.6.a Compare and refine multiple algorithms for the same task to determine which is the most efficient.

Variables and Data Representation

ATP.VDR.6.a Identify unknown values that need to be represented by a variable within a multi-step process.

ATP.VDR.6.b Create variables and use them within a multi-step process.

Control Structures

ATP.CS.6.a Identify and trace decisions and loops that exist in a multi-step process within a program.

Modularity

ATP.M.6.a Decompose problems into parts to facilitate the design, implementation and review of programs.

Program Development

ATP.PD.6.a Write code that utilizes algorithms, variables and control structures to solve problems or as a creative expression.

ATP.PD.6.b Test and trace to debug and refine code.

IMPACTS OF COMPUTING

Culture

IC.Cu.6.a Identify the change that current technologies have on people's everyday activities to understand the impact within a society.

IC.Cu.6.b Identify issues of bias and accessibility in the design of existing technologies to address equality and equity in society.

IC.Cu.6.c Identify and explore careers related to the field of computer science.

Social Interactions

IC.SI.6.a Analyze and present beneficial and harmful effects of electronic communications to understand their impacts on interpersonal, global, economic, political, business and cultural interactions.

Safety, Law and Ethics

IC.SLE.6.a Describe tradeoffs between allowing information to be public and keeping information private and secure to inform decision making.

IC.SLE.6.b Identify the social and economic implications of privacy in the context of safety, law or ethics to understand how privacy impacts these areas.

IC.SLE.6.c Evaluate the development of new technologies in communication, entertainment and business to understand the impact.

IC.SLE.6.d Provide appropriate credit when using resources or artifacts that are not our own.

IC.SLE.6.e Differentiate between the appropriate and inappropriate content on the internet and identify unethical and illegal online behavior.

Grade 7

COMPUTING SYSTEMS

Devices

CS.D.7.a Develop and implement a process to evaluate existing computing devices capabilities based on personal interaction with the device.

Hardware and Software

CS.HS.7.a Evaluate hardware and software combinations used to accomplish a task.

Troubleshooting

CS.T.7.a Use a systematic process to identify and evaluate the source of a routine computing problem. Select the best solution to solve the computing problem and communicate the solution to others.

NETWORKS AND THE INTERNET

Networking

NI.N.7.a Explain the role of hardware components and diagram the infrastructure of networks and the internet (including cloud servers).

NI.N.7.b Explain the protocols (i.e., rules) and why they are used to transmit data across networks and the internet.

Cybersecurity

NI.C.7.a Identify and apply introductory methods of encryption to model the secure transmission of information.

NI.C.7.b Describe the types of malware to show how malware affects information.

NI.C.7.c Identify cybersecurity concerns and measures needed to protect electronic information.

DATA AND ANALYSIS

Data Collection and Storage

DA.DCS.7.a Compare and contrast digital data collection tools to make them more useful and reliable.

DA.DCS.7.b Evaluate various file formats to understand data storage capabilities.

DA.DCS.7.c Create a logical file structure to organize data to support individual and collaborative work.

Visualization and Communication

DA.VC.7.a Communicate relations between data sets to interpret results.

DA.VC.7.b Create a spreadsheet utilizing formulas, functions and graphs to represent and analyze data.

Inference and Modeling

DA.IM.7.a Create and analyze models and simulations to accurately hypothesize a real-world situation.

ALGORITHMIC THINKING AND PROGRAMMING

Algorithms

ATP.A.7.a Select and modify pseudocode for a multi-step process to solve a problem.

Variables and Data Representation

ATP.VDR.7.a Use test cases to trace variable values to determine the result.

Control Structures

ATP.CS.7.a Use and apply decisions and loops in a program to solve a problem.

Modularity

ATP.M.7.a Decompose problems into parts to facilitate the design, implementation and review of increasingly complex programs.

Program Development

ATP.PD.7.a Write code that utilizes algorithms, variables and control structures to solve problems or as a creative expression.

ATP.PD.7.b Test, trace and debug to refine code.

ATP.PD.7.c Identify procedures that utilize parameters.

IMPACTS OF COMPUTING

Culture

IC.Cu.7.a Compare current technologies from the present to the past to evaluate the effect on people's everyday activities.

IC.Cu.7.b Evaluate various technologies to identify issues of bias and accessibility.

IC.Cu.7.c Identify and explore careers related to the field of computer science.

IC.Cu.7.d Explain how computing impacts innovation in other fields.

Social Interactions

IC.SI.7.a Analyze and present beneficial and harmful effects of electronic communications to understand their impacts on interpersonal, global, economic, political, business and cultural interactions.

Safety, Law and Ethics

IC.SLE.7.a Describe tradeoffs between allowing information to be public and keeping information private and secure to inform decision making.

IC.SLE.7.b Identify the social and economic implications of privacy in the context of safety, law or ethics to understand how privacy impacts these areas.

IC.SLE.7.c Evaluate the development of new technologies in communication, entertainment and business to understand the impact.

IC.SLE.7.d Provide appropriate credit when using resources or artifacts that are not our own.

IC.SLE.7.e Explain the connection between the longevity of data on the internet, personal online identity and personal privacy.

Grade 8

COMPUTING SYSTEMS

Devices

CS.D.8.a Evaluate the advantages and limitations of existing computing devices to recommend design improvements based on analysis of how users interact with the device.

Hardware and Software

CS.HS.8.a Design projects that combine hardware and software components that could complete a task.

Troubleshooting

CS.T.8.a Use a systematic process to identify and evaluate the source of a routine computing problem. Select the best solution to solve the computing problem and communicate the solution to others.

NETWORKS AND THE INTERNET

Networking

NI.N.8.a Model the role of hardware components to diagram the infrastructure of networks and the internet (including cloud servers).

NI.N.8.b Model protocols (i.e., rules) and explain why they are used to transmit data across networks and the internet.

NI.N.8.c Explain how a system responds when information is lost to understand the effect it has on the transferred information.

Cybersecurity

NI.C.8.a Explain how physical and digital security measures are used to protect electronic information.

NI.C.8.b Compare and contrast the effects of different types of malware to determine strategies for how to protect devices.

DATA AND ANALYSIS

Data Collection and Storage

DA.DCS.8.a Interpret digital data collection tools to manage information effectively.

DA.DCS.8.b Identify data storage systems to define how data is stored and accessed.

DA.DCS.8.c Create a logical file structure to organize data in different storage systems to support individual and collaborative work.

Visualization and Communication

DA.VC.8.a Evaluate data to construct a model or representation.

DA.VC.8.b Create a spreadsheet utilizing formulas, functions and graphs to represent and analyze data.

Inference and Modeling

DA.IM.8.a Create and analyze models and simulations to accurately hypothesize a real-world situation.

ALGORITHMIC THINKING AND PROGRAMMING

Algorithms

ATP.A.8.a Create multiple pseudocode to solve a multi-step process and justify the most efficient solution.

Variables and Data Representation

ATP.VDR.8.a Analyze test cases and determine the range of valid solutions.

ATP.VDR.8.b Use a data structure to represent a collection.

Control Structures

ATP.CS.8.a Use and apply decisions and loops in a program to solve a problem.

Modularity

ATP.M.8.a Decompose problems and subproblems into parts to facilitate the design, implementation and review of complex programs.

Program Development

ATP.PD.8.a Write code that utilizes algorithms, variables and control structures to solve problems or as a creative expression.

ATP.PD.8.b Systematically test and refine programs using a range of test cases.

ATP.PD.8.c Use procedures that utilize parameters to pass values.

IMPACTS OF COMPUTING

Culture

IC.Cu.8.a Compare current technologies and how they affect the current economy.

IC.Cu.8.b Propose potential guidelines/standards/criteria to positively impact bias and accessibility in the design of future technologies.

IC.Cu.8.c Identify and explore careers related to the field of computer science.

IC.Cu.8.d Explain how computing impacts innovation in other fields.

Social Interactions

IC.SI.8.a Evaluate the impacts of electronic communication on personal relationships to be able to evaluate differences between face-to-face and electronic communication.

Safety, Law and Ethics

IC.SLE.8.a Explain user privacy concerns related to the collection and generation of data that may not be evident through automated processes.

IC.SLE.8.b Describe the social and economic implications of privacy in the context of safety, law or ethics to be global digital citizens.

IC.SLE.8.c Identify ethical and legal security measures used to protect electronic information.

IC.SLE.8.d Provide appropriate credit when using resources or artifacts that are not our own.

Grades 9 - 12—Foundational Level

COMPUTING SYSTEMS

Devices

CS.D.9-12.F.a Identify different multifunctional computing devices and connection technologies, both virtual and physical, to describe their purpose.

CS.D.9-12.F.b Develop and apply criteria to evaluate computing systems for a given purpose.

CS.D.9-12.F.c Create an artifact to demonstrate the roles and interactions of computing systems embedded in everyday objects.

Hardware and Software

CS.HS.9-12.F.a Compare and contrast interactions between application software, system software and hardware.

Troubleshooting

CS.T.9-12.F.a Apply a systemic process to identify problems and take steps to correct them within an integrated computing system.

CS.T.9-12.F.b Analyze an IT device to determine either what repairs are needed or how to build it.

NETWORKS AND THE INTERNET

Networking

NI.N.9-12.F.a Evaluate and select networking devices to establish scalable communications.

NI.N.9-12.F.b Evaluate and select networking protocols to establish network communication.

NI.N.9-12.F.c Understand scalability and reliability of networks to describe the relationships and effects of how the different types of networks work together.

Cybersecurity

NI.C.9-12.F.a Examine and employ principles of cybersecurity.

NI.C.9-12.F.b Identify physical, social and digital security risks to address possible attacks.

DATA AND ANALYSIS

Data Collection and Storage

DA.DCS.9-12.F.a Analyze patterns in a real-world data store through hypothesis, testing and use of data tools to gain insight and knowledge.

DA.DCS.9-12.F.b Investigate data storage systems to compare and contrast how data is stored and accessed.

Visualization and Communication

DA.VC.9-12.F.a Analyze the benefits and limitations of data visualization or multisensory artifacts and tools to communicate which is most appropriate to solve a real-world problem.

Inference and Modeling

DA.IM.9-12.F.a Evaluate a model by creating a hypothesis, testing it and refining it to discover connections and trends in the data.

ALGORITHMIC THINKING AND PROGRAMMING

Algorithms

ATP.A.9-12.F.a Define and use appropriate problem solving strategies and visual artifacts to create and refine a solution to a real-world problem.

ATP.A.9-12.F.b Define and implement an algorithm by decomposing problem requirements from a problem statement to solve a problem.

Variables and Data Representation

ATP.VDR.9-12.F.a Identify types of variables and data and utilize them to create a computer program that stores data in appropriate ways.

Control Structures

ATP.CS.9-12.F.a Define control structures and Boolean logic and use them to solve real-world scenarios.

ATP.CS.9-12.F.b Use appropriate syntax to create and use a method.

ATP.CS.9-12.F.c Use data scoping to isolate data.

Modularity

ATP.M.9-12.F.a Break down a solution into procedures using systematic analysis and design.

Equivalent to: ATP.A.9-12.F.b Define and implement an algorithm by decomposing problem requirements from a problem statement to solve a problem.

ATP.M.9-12.F.b Create computational artifacts by systematically organizing, manipulating and/or processing data.

Addressed in:

ATP.VDR.9-12.A.a Utilize different data storage structures to store larger and more complex data than variables can contain.

ATP.VDR.9-12.A.b Identify the appropriate data structures or variables to use to design a solution to a complex problem.

Program Development

ATP.PD.9-12.F.a Investigate software development methodologies to select the appropriate one for a project to complete as a team.

ATP.PD.9-12.F.b Compare test methodologies to evaluate why each is used and to determine their benefits and costs.

ATP.PD.9-12.F.c Correctly use consistent naming conventions, version control and comments to demonstrate why these are important for future use, maintenance and reuse of code.

IMPACTS OF COMPUTING

Culture

IC.Cu.9-12.F.a Analyze new technology to predict realistic impacts on society.

IC.Cu.9-12.F.b Explore other professions to understand how computing has and will impact them positively and negatively.

Social Interactions

IC.SI.9-12.F.a Evaluate tools to increase connectivity of people in different cultures and career fields.

IC.SI.9-12.F.b Analyze the collection and generation of data through automated processes to explain the privacy concerns that are not always evident to users.

Safety, Law and Ethics

IC.SLE.9-12.F.a Interpret and analyze breaches in privacy and security to investigate the legal and ethical impact.

IC.SLE.9-12.F.b Analyze the concepts of usability and security to explain typical tradeoffs between them.

IC.SLE.9-12.F.c Analyze the collection and generation of data through automated processes to explain the legal concerns that are not always evident to users.

IC.SLE.9-12.F.d Explain the beneficial and harmful effects of intellectual property laws to determine the impacts on innovation.

Grades 9 - 12—Advanced Level

COMPUTING SYSTEMS

Devices

CS.D.9-12.A.a Evaluate the function of various devices to formulate a human interaction solution.

CS.D.9-12.A.b Integrate multifunctional computing devices to solve a problem.

CS.D.9-12.A.c Identify the functionality of various categories of hardware components and the communication between them, and use that information to build a system virtually or physically for a specific task.

Hardware and Software

CS.HS.9-12.A.a Categorize types of operating systems and how they will be used.

Troubleshooting

CS.T.9-12.A.a Evaluate and revise a systematic process to identify the source of a problem and the steps to correct it within individual and connected devices.

NETWORKS AND THE INTERNET

Networking

NI.N.9-12.A.a Construct a networking devices map solution for a real-world scenario to establish communication between distant devices.

NI.N.9-12.A.b Develop a solution to a real-world scenario using networking protocols to establish network communication.

NI.N.9-12.A.c Improve scalability and reliability of networks to describe the relationships and effects of how the different types of networks work together.

Cybersecurity

NI.C.9-12.A.a Identify cybersecurity ethics and law.

NI.C.9-12.A.b Implement a devised solution to counter a security threat.

DATA AND ANALYSIS

Data Collection and Storage

DA.DCS.9-12.A.a Create multidimensional data collections that can be utilized through various methods to solve complex data problems.

DA.DCS.9-12.A.b Investigate data storage and collection tools to analyze tradeoffs and limitations.

Visualization and Communication

DA.VC.9-12.A.a Create visualization or multisensory artifacts to communicate insights and knowledge gained from complex data analysis that answers real-world questions.

Inference and Modeling

DA.IM.9-12.A.a Create a model that simulates a complex system and uses extracted data to hypothesize, test and refine the model to discover connections or trends.

ALGORITHMIC THINKING AND PROGRAMMING

Algorithms

ATP.A.9-12.A.a Define and explain recursive algorithms to understand how and when to apply them.

ATP.A.9-12.A.b Use recursion to effectively solve problems.

ATP.A.9-12.A.c Define and explain sorting and searching algorithms to understand how and when to apply them.

ATP.A.9-12.A.d Use sorting and searching to analyze and organize data.

Variables and Data Representation

ATP.VDR.9-12.A.a Utilize different data storage structures to store larger and more complex data than variables can contain.

ATP.VDR.9-12.A.b Identify the appropriate data structures or variables to use to design a solution to a complex problem.

Control Structures

ATP.CS.9-12.A.a Write programs that use library methods and control structures and methods to solve a problem.

ATP.CS.9-12.A.b Refactor a program to be smaller and more efficient.

Modularity

ATP.M.9-12.A.a Construct solutions to problems using student-created components (e.g., procedures, modules, objects).

Equivalent to: ATP.CS.9-12.F.b Use appropriate syntax to create and use a method.

ATP.M.9-12.A.b Design or redesign a solution to a large-scale computational problem by identifying generalizable patterns.

Equivalent to: ATP.PD.9-12.A.a Fully implement the most appropriate software methodology to complete a team programming project.

ATP.M.9-12.A.c Create programming solutions by reusing existing code (e.g., libraries, Application Programming Interface (APIs), code repositories).

Equivalent to: ATP.CS.9-12.A.a Write programs that use library functions, methods and control structures to solve a problem.

Program Development

ATP.PD.9-12.A.a Fully implement the most appropriate software methodology to complete a team programming project.

IMPACTS OF COMPUTING

Culture

IC.Cu.9-12.A.a Evaluate an alternative solution where a current tool does not exist due to limited resources.

IC.Cu.9-12.A.b Analyze the equity, access and influence of the distribution of computing resources to see their global impact.

IC.Cu.9-12.A.c Design a study to predict how computers will revolutionize an aspect of our culture.

Safety, Law and Ethics

IC.SLE.9-12.A.a Create a scenario to demonstrate typical tradeoffs between usability and security and recommend security measures based on these or other tradeoffs.

IC.SLE.9-12.A.b Investigate intellectual property laws, including copyright, trademarks and patents, to identify some of the practical, business and ethical impacts.

Acknowledgements

ADVISORY COMMITTEE MEMBERS

Grady Burrows
BioEnterprise

Chelsey Cook-Kohn
John Marshall School of IT

Gary Greenberg
CET Connect/ThinkTV

Gloria Rogiers
Columbus State Community College

Nigamanth Sridhar
Cleveland State University

Tim Sisson
Cleveland Metropolitan SD (CMSD)

Chris Anderson
LexisNexis

Kevin Boehner
Ohio Chamber of Commerce

Lisa Chambers
Tech Corps

Tad Douce
TRECA Education

Chris Ferruso
NFIB/Ohio

Kelly Gaier-Evans
Battelle Education

Stephanie Hoeppner
Computer Science Teachers Association (CSTA)

Diane Kahle
Ohio Council for Teachers of Mathematics

Dan McMullen
Greater Cleveland Partnership/OHTec

Kenneth Moore
Wright State University

WRITING TEAM MEMBERS

Laura Ciszweski
Auburn City Schools,
Auburn Career Center

Darrin Spondike
Auburn City Schools,
Auburn Career Center

Robert Wyant
Austintown Local,
Austintown Fitch High School

Roberta Banfield
Cleveland Catholic Diocese,
Corpus Christi Academy

Patrick Connor
Cleveland Catholic Diocese,
St Ignatius High School

Sheryl Edwards
Celveland Metropolitan School District,
Campus International

Briana Guevara
Celveland Metropolitan School District,
John Marshal School of information
Technologyx

Jennifer Humphrey
Celveland Metropolitan School District,
Campus International

Pat Murakami
Dayton Regional STEM School

Shanti Coaston
East Cleveland City Schools,
Shaw High School

Tishina Brown-Harris
Eastern Local School District,
Eastern Elementary

Lashawna Grimes
Emmanuel Christian Academy

Christopher Kaminski
Fairview Park City Schools,
Fairview High School

Leah Burke
Westlake City Schools,
Bassett Elementary

Renee Snyder
Teaching & Learning Collaborative

John Fulton
Tech Elevator, Inc.

Julie Hagaman
Tri-County North School District

Lynne Pachnowski
University of Akron

Karen Plaster
University of Akron

Diane Kahle
Upper Arlington City

Lisa Murray
Vermilion Local Schools

Jenifer Conard
Warren County Career Center

Barbara Kenney
West Clermont Local

Judith Gawelek
Westerville City School District

Leah Burke
Westlake City Schools

Dave Burkhart
New Lexington City

Pamela Cook
Notre Dame College

Martin Fish
Mason City School District

Jessica Frenchik
Strongsville City

Melissa Goodall
Miami Valley Career Tech

Gary Herman
Putnam County ESC

Marvin "Mel" Hoffert
Sycamore Community

Bonnie Kovacic
Perry Local

Martin Chenzira
Pickerington Local School District

Carlo Miraldi
Great Oaks Career Campuses

Valerie Moga
Hudson City

Shellu Mumaw
Ottoville Local School

Erin Pekar
Maple Heights

Krystina Pratt
New Albany Plain Local Schools

Rebecca Stanek
Stow-Munroe Falls City

Leslie Slaven
Lakota Local Schools

James Stanton
Maumee City Schools

Deborah Wiley
North Canton City Schools/Kent State Stark Univ.

Jonathon Wilson
New Richmond Exempted Village School District

References

Bureau of Labor Statistics. (2015). Labor force statistics from the Current Population Survey [Data file]. Retrieved from http://www.bls.gov/cps

Change the Equation. (2016, August 9). New data: Bridging the computer science access gap [Blog post]. Retrieved from http://changetheequation.org/blog/new-data-bridging-computer-science-access-gap-0 (no longer available)

College Board. (2016). AP program participation and performance data 2015 [Data file]. Retrieved from https://research.collegeboard.org/programs/ap/data/archived/ap-2015

Denner, J., Martinez, J., Thiry, H., & Adams, J. (2015). Computer science and fairness: Integrating a social justice perspective into an after school program. *Science Education and Civic Engagement: An International Journal*, *6*(2): 41–54.

Goode, J. (2008, March). Increasing diversity in K–12 computer science: Strategies from the field. *ACM SIGCSE Bulletin*, *40*(1), 362–366.

Hansen, A., Hansen, E., Dwyer, H., Harlow, D., & Franklin, D. (2016). Differentiating for diversity: Using universal design for learning in computer science education. *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (pp. 376–381).

K–12 Computer Science Framework. (2016). Retrieved from http://www.k12cs.org

Kumar, A. N. (2012, July). A study of stereotype threat in computer science. *Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education* (pp. 273–278).

Margolis, J., Ryoo, J., Sandoval, C., Lee, C., Goode, J., & Chapman, G. (2012). Beyond access: Broadening participation in high school computer science. *ACM Inroads*, *3*(4), 72–78.

Margolis, J., Goode, J., Chapman, G., & Ryoo, J. J. (2014). That classroom 'magic.' *Communications of the ACM*, *57*(7), 31–33.

Morgan, R., & Klaric, J. (2007). AP students in college: An analysis of five-year academic careers. Research report no. 2007-4. Retrieved from https://eric.ed.gov/?id=ED561034

Quorum [Computer software]. (2019). Retrieved from https://quorumlanguage.com/

Snodgrass, M. R., Israel, M., & Reese, G. (2016). Instructional supports for students with disabilities in K–5 computing: Findings from a cross-case analysis. *Computers & Education*, *100*, 1–17.

Strawhacker, A. L., & Bers, M. U. (2014, August). *ScratchJr: Computer programming in early childhood education as a pathway to academic readiness and success*. Poster presented at DR K–12 PI Meeting, Washington, DC.