Ohio’s Model Curriculum for Computer Science

Grade 9-12 Advanced

ADOPTED SEPT. 2022

Ohio | Department of Education

Strand	Computing Systems
Topic	**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.

Expectations for Learning

LEARNING PROGRESSION

By the end of the 9-12 Foundational Level:
Students will be able to extract technological information about computing systems to determine the best practice for situations.

In the 9-12 Advanced Level:
Students will extract technological information about computing systems to determine and integrate the best practices for complex situations.

Beyond High School:
Students will gain a deeper understanding of how all devices interact in computing systems at the collegiate level.

IMPORTANT CONCEPTS

- Devices are used to solve problems.
- Different devices interact with multiple devices.
- Different devices can be connected into a computing system.
- There are multiple methods to evaluate a computing system.

KEY SKILL/PROCEDURES

- Differentiate between a simple and complex device to understand how each is used both independently and as a resource in a system.
- Utilize multifunctional computing devices to solve a problem.
- Build a computing system to solve a need.

Content Elaborations

CLARIFICATIONS

Examples of simple devices could be:

- a garage door opener,
- a toaster and
- a microwave.

Strand	Computing Systems
Topic	**Devices**
	Examples of complex devices could be: • a tablet on a refrigerator, • alarm system and • multi-functional printer. **CONTENT FOCUS** Understand how multiple devices communicate with each other. Utilize multiple devices to solve a problem **COMPUTER SCIENCE PRACTICES** *Practice 4*. Developing and Using Abstractions 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.

Strand	Computing Systems
Topic	**Hardware/Software**

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

Expectations for Learning

LEARNING PROGRESSION

By the end of the 9-12 Foundational Level:
Students can dictate the appropriate software to install onto specific hardware to work as specified.

In the 9-12 Advanced Level:
Students will construct a working system based on hardware and software criteria.

Beyond High School:
Students will continue to further explore and evaluate the interactions and relationships between systems software and hardware at the collegiate level.

IMPORTANT CONCEPTS

- Different operating systems are used for different purposes.

KEY SKILL/PROCEDURES

- Describe the hardware requirements and the purpose of the hardware to determine which operating system will be used for installation.

Content Elaborations

CLARIFICATIONS

Operating systems are software that controls the computer basic functions. Examples of operating system types are:

- server (web, e-mail, file, application, print),
- desktop/client,
- mobile and
- IoT simple/complex (garage door opener/multi-function printer).

CONTENT FOCUS

Focus on constructing a system for a given purpose to enhance its performance, reliability, usability and accessibility.

Strand	Computing Systems
Topic	**Hardware/Software**
	COMPUTER SCIENCE PRACTICES *Practice 5*. Creating Computational Artifacts 1. Plan the development of a computational artifact using an iterative process that includes reflection on and modification of the plan, considering 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 6.3. Evaluate and refine a computational artifact multiple times

Strand	Computing Systems
Topic	**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.

Expectations for Learning

LEARNING PROGRESSION

By the end of the 9-12 Foundational Level:
Students will understand the troubleshooting process to evaluate a predetermined situation.

In the 9-12 Advanced Level:
Students will use the troubleshooting process to resolve a real-world problem.

Beyond High School:
Students will continue to identify and compare more complex problems in computing systems at the collegiate level.

IMPORTANT CONCEPTS

- Some processes are better suited to troubleshoot a given problem.
- Processes can be fine-tuned to create a better solution.

KEY SKILL/PROCEDURES

- Evaluate a systematic process for suitability to a problem.
- Modify a systematic troubleshooting process.

Content Elaborations

CLARIFICATIONS

The troubleshooting process includes the following:

1. Identify a problem/Gather information,
2. Theorize possible solutions,
3. Test possible solutions. If no solutions result, return to step 2,
4. Implement the solution,
5. Do a full system test,
6. Do a user test and
7. Create documentation.

CONTENT FOCUS

Focus on identifying the most effective systematic process to solve a problem.

Strand	Computing Systems
Topic	**Troubleshooting**
	COMPUTER SCIENCE PRACTICES *Practice 3.* Recognizing and Defining Computational Problems 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 6*. Testing and Refining Computational Artifacts 1. Systematically test computational artifacts by considering all scenarios and using test cases. 2. Identify and fix errors using a systematic process. *Practice 7*. Communicating About Computing 2. Describe, justify and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose.

Strand	Networks and the Internet
Topic	**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.

Expectations for Learning

LEARNING PROGRESSION

By the end of the 9-12 Foundational Level:
Students are able to gather information to be able to create simple LANs and WANs with the understanding of how devices connect.

In the 9-12 Advanced Level:
Students will be able to develop and connect a working network model from given criteria.

Beyond High School:
Students should be able to fully understand how devices communicate and connect beyond the local LAN.

IMPORTANT CONCEPTS

- Network architecture is both logical and physical subnetting.
- Protocols are used for network communications.

KEY SKILL/PROCEDURES

- Demonstrate understanding of subnetting.
- Choose the appropriate network protocol to use for specific communication situations.
- Create a LAN.
- Expand a LAN to a WAN.

Content Elaborations

CLARIFICATIONS

Know IPV4 and IPV6 addressing.

Know variable-length subnetting and CIDR notation.

Know basic networking protocols.

Distance communication can be a remote connection/VPN connection.

CONTENT FOCUS

- IPV4 and IPV6 addressing
- Variable length subnetting and CIDR notation
- Basic networking protocols

Strand	Networks and the Internet
Topic	**Networking**
	COMPUTER SCIENCE PRACTICES *Practice 2.* Collaborating Around Computing 4. Evaluate and select technological tools that can be used to collaborate on a project. *Practice 4.* Developing and Using Abstractions 2. Decompose complex real-world problems into manageable subproblems that could integrate existing solutions or procedures. 3. Create modules and develop points of interaction that can apply to multiple situations and reduce complexity. *Practice 5.* Creating Computational Artifacts 3. Modify an existing artifact to improve or customize it.

Strand	Networks and the Internet
Topic	**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.

Expectations for Learning

LEARNING PROGRESSION

By the end of the 9-12 Foundational Level:
Students can understand the cybersecurity impact has on business and the economy.

In the 9-12 Advanced Level:
Students will be able to find the cybersecurity threat and provide a plan of action to remove the threat.

Beyond High School:
Students will be able to intern at a company and be able to help identify and mitigate cybersecurity threats in real-time.

IMPORTANT CONCEPTS

- There is a difference between ethical and unethical uses of cybersecurity.
- There are multiple techniques that can be used to counter a security threat.
- A plan of action should include multiple ways to secure a network and counter an attack.

KEY SKILL/PROCEDURES

- Remove malware threats.
- Implement authentication, authorization and verification.
- Implement a plan of action to remove a cybersecurity threat.

Content Elaborations

CLARIFICATIONS

Know what virus protection software does and how it works.

Understand a signature to be able to maintain integrity.

Know how to detect where threats are located and how to remove these safely (i.e., safe mode.)

Know how to reinstall the system from backups.

Know how social threats are used to alter reliability (e.g., changing of data.)

CONTENT FOCUS

Understanding virus protection.

Focus on ethics and laws regarding cybersecurity.

Strand	Networks and the Internet	
Topic	**Cybersecurity**	
		Implementing a solution to address threats. **COMPUTER SCIENCE PRACTICES** *Practice 2.* Collaborating Around Computing 1. Cultivate working relationships with individuals possessing diverse perspectives, skills and personalities. 4. Evaluate and select technological tools that can be used to collaborate on a project. *Practice 3.* Recognizing and Defining Computational Problems 2. Decompose complex real-world problems into manageable subproblems that could integrate existing solutions or procedures.
NI.C.9-12.A.c Compare and contrast various threat actors, such as nation-states, cyber terrorist groups, organized crime or hacktivists. **NI.C.9-12.A.d** Explore and utilize examples of encryption methods (e.g., Vigenére, Bacon's cipher and Enigma).		**Expectations for Learning** **LEARNING PROGRESSION** In the Foundational Level 9-12, students researched types of threat actors. In the Advanced Level 9-12, students should focus on a deeper level of understanding of how and why various threat actors break the law, what individuals and groups hope to gain or change by their actions and the mitigation of physical security attacks, such as social engineering, security policies and malicious actors. **IMPORTANT CONCEPTS** • Compare encryption methods with a focus on real-world examples. • Compare types of threat actors and their objectives. • Use safe browsers and cyber practices to protect personal information. • Explore and discuss real-world scenarios including social engineering and other cyber threat vectors. **KEY SKILLS/PROCEDURES** • Discuss threat vectors and the threat actors behind attacks. • Discuss the interrelationship between the CIA (confidentiality, integrity and availability) triad; a concept that focuses on the balance between the confidentiality, integrity and availability of data under the protection of information security programs.

Strand	Networks and the Internet
Topic	**Cybersecurity**
	Content Elaborations **CLARIFICATIONS** Cybersecurity assessment examples may include full- or limited-scope penetration tests, hardware analysis and source code audits. Undesired outcomes include a false sense of security that may result from improperly conducted assessments. **CONTENT FOCUS** Researching malware includes understanding the different classes of malware (e.g., potentially unwanted programs, ransomware, rootkits, trojans, viruses, worms) and the reasoning for its application by an adversary. Mitigations may include configuring file permissions, configuring host- network-based firewalls and using encryption technology for network communications. Mitigation strategies include reducing the potential vulnerabilities caused by social engineering of humans, which is an attack vector present in all systems, including hardware, software and networks. **COMPUTER SCIENCE** *Practice 3.* Recognizing and Defining Computational Problems 1. Identify complex, interdisciplinary, real-world problems that can be solved computationally. *Practice 6.* Testing and Refining Computational Artifacts. 1. Systematically test computational artifacts by considering all scenarios and using test cases.

Strand	Networks and the Internet
Topic	**Internet of Things**

NI.IOT.9-12.A.a Design and implement an IoT life cycle scenario that encompasses data gathering, transmission, reception and data analysis to demonstrate how the IoT operates and apply these skills to design products that model the process.

Expectations for Learning

LEARNING PROGRESSION

In grade 8:
Students modeled the lifecycle of information in the IoT: data gathering, transmission, reception and analysis to recreate a real-world scenario.

By the end of the 9-12 Foundational Level:
Students are able to gather information to be able to design simple smart applications and an accompanying life cycle.

In the 9-12 Advanced Level:
Students will be able to design and implement a working smart device and its life cycle or application for a smart device life cycle. Beyond high school, students should be able to fully understand how smart devices communicating with each other create an IoT.

IMPORTANT CONCEPTS

- The IoT life cycle consists of smart devices.
- A smart device may receive as well as send information.
- Students will identify and analyze the data they gather and what they transmit across the Internet, as well as what entity receives the data and what is done with the data after it is received.

KEY SKILLS/PROCEDURES

- Demonstrate understanding of smart devices.
- Demonstrate the life cycle of smart device implementations.
- Design a smart device life cycle.
- Implement a smart device life cycle (advanced).

Content Elaborations

CLARIFICATIONS

Smart devices, while an essential part of an IoT life cycle, must be paired with other devices to transmit and receive data. That data must then be analyzed to provide meaning from that data.

CONTENT FOCUS

Smart device life cycles.

Strand	Networks and the Internet
Topic	**Internet of Things**
	COMPUTER SCIENCE PRACTICES *Practice 3.* Recognizing and Defining Computational Problems 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. *Practice 4.* Developing and Using Abstractions 2. Evaluate existing technological functionalities and incorporate them into new designs. *Practice 5.* Creating Computational Artifacts 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. *Practice 6.* Testing and Refining Computational Artifacts 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. **Career Connections** **CAREER PLANNING** Have students work with professionals such as those within the district's Business Advisory Council to identify mentorship/job shadowing opportunities where they can explore how IoT relates to the world of work and how the IoT life cycle scenario is applied. Consider how this type of opportunity may lead to an internship or other work-based learning opportunity for students.

Strand	Data and Analysis
Topic	**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.

Expectations for Learning

LEARNING PROGRESSION

In the 9-12 Foundational Level:
Students will use data analysis tools and scientific practices to organize data and identify trends, patterns and outliers. Students will be able to understand the benefits of different data collection tools and pick the appropriate one to use for a problem. Students will learn how data is stored on different computer systems.

By the end of the 9-12 Advanced Level:
Students will apply basic techniques for collecting, manipulating and interpreting small-scale multidimensional data sets (e.g., collecting and organizing user surveys, accessing real-world data sets). Students will use various data collection techniques for different types of computational problems (e.g., mobile device Global Positioning System (GPS), user surveys, embedded system sensors, open data sets, social media data sets).

Beyond High School:
Students will apply techniques for collecting, manipulating and interpreting large-scale multidimensional data sets (e.g., collecting and organizing user surveys, accessing real-world data sets).

IMPORTANT CONCEPTS

- A multidimensional collection is a collection of multiple sets of the same data.
- Multidimensional collections can vary in performance depending on the data.

KEY SKILL/PROCEDURES

- Utilize various methods of data collection.
- Manipulate and interpret small-scale multidimensional data sets.
- Identify the best data collection tool for a given situation.

Content Elaborations

CLARIFICATIONS

When collecting data, be sure to address the tradeoffs and limitations of data storage.

Below, are examples of multidimensional collections:

- A library contains data about books.
- A window has many panes.
- A video game has many NPCs who have the same attributes, but different values in those attributes.

Strand	Data and Analysis
Topic	**Data Collection and Storage**
	Data analysis tools could include: • Spreadsheets, • Charts, • Graphs and • SQL. **CONTENT FOCUS** Focus on how to collect, organize and manipulate data to solve real-world problems. **COMPUTER SCIENCE PRACTICES** *Practice 7.* Communicating About Computing 1. Select, organize and interpret large data sets from multiple sources to support a claim.

Strand	Data and Analysis
Topic	**Visualization and Communications**

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.

Expectations for Learning

LEARNING PROGRESSION

By the end of the 9-12 Foundational Level:
Students will be able to evaluate the ability of data visualization artifacts to identify key features and relations by using scientific practices.

In the 9-12 Advanced Level:
Given a real-world relatable problem, students will perform complex data analysis to create a data visualization that produces an optimal solution.

Beyond High School:
Students will further their understanding of advanced visualization techniques. They will also look at advanced data analysis tools.

IMPORTANT CONCEPTS

- Data analysis is used to find patterns and trends.
- Different inputs and actions can affect behaviors.
- Real-world questions can be restated and interpreted as data questions.
- Based on the data analysis, changes can be made to improve a solution or process.

KEY SKILL/PROCEDURES

- Accurately complete complex data analysis.
- Create multisensory artifacts from data sets.
- Translate real-world questions into data analysis questions.
- Communicate the solutions found in the data set.

Content Elaborations

CLARIFICATIONS

Complex data analysis could include:

- relational database queries,
- data regression,
- spreadsheets,
- inferences and
- predictions.

Strand	Data and Analysis
Topic	**Visualization and Communications**
	Data questions include testable hypotheses based on the data. For example, "If the price of eggs goes up, do people buy fewer eggs?" **CONTENT FOCUS** Focus on how to translate real-world questions into data questions that can be answered by data analysis. Have students create and present the visualizations based on their analysis. **COMPUTER SCIENCE PRACTICES** *Practice 3.* Recognizing and Defining Computational Problems 1. Identify complex, interdisciplinary, real-world problems that can be solved computationally. *Practice 5.* Creating Computational Artifacts 2. Create a computational artifact for practical intent, personal expression, or to address a societal issue. *Practice 7.* Communicating About Computing 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.

Strand	Data and Analysis
Topic	**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.

Expectations for Learning

LEARNING PROGRESSION

In the 9-12 Foundational Level:
Students will construct a model using collected data, and create a hypothesis, test it and refine it to discover connections and trends.

In the 9-12 Advanced Level:
Students will collect, analyze and reorganize multiple sets of data in order to construct a model, and create a hypothesis, test it and refine it to discover connections and trends between the multiple sets of data.

Beyond High School:
Student will collect, analyze and reorganize multiple sets of data to construct a real-world model to find patterns and be able to do prediction analysis.

IMPORTANT CONCEPTS

- Data models can have different applications.
- Data models can be constructed from multiple data sources.

KEY SKILL/PROCEDURES

- Demonstrate an understanding of when and when not to create a data model.
- Explain and correctly use advanced common model notations.
- Create a data model from multiple data sources.

Content Elaborations

CLARIFICATIONS

Apply data model patterns

Normalize to reduce data redundancy

De-normalize to improve performance

An example of multiple sets of data would be a high school course registration where the following exists: a schedule, prerequisites, seat maximum/minimum, teacher/course availability and early registration for more senior students.

Strand	Data and Analysis
Topic	**Inference and Modeling**
	Below are common model notations. • Associations • Entity roles • Subtyping • Aggregation • Composition • OR constraint • Exclusive OR (XOR) constraint **CONTENT FOCUS** Use advanced common model notations when creating a data model from multiple sources. **COMPUTER SCIENCE PRACTICES** *Practice 6*. Testing and Refining Computational Artifacts 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 3. Articulate ideas responsibly by observing intellectual property rights and giving appropriate attribution.

Strand	Algorithmic Thinking and Programming
Topic	**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 iteration to effectively solve problems.

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

ATP.A.9-12.A.d 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.

ATP.A.9-12.A.f Compare and contrast classical, cluster and quantum computing algorithms.

Expectations for Learning

LEARNING PROGRESSION

By the end of the 9-12 Foundational Level:
Students can break a problem into steps, and then, in addition to pseudocode, represent the algorithm using process and data flow diagrams. Students can explain the steps of the problem and be able to communicate why they broke the problem down the way they did. Finally, they can implement their created algorithm in an appropriate programming language and refine it using best practices.

In the 9-12 Advanced Level:
Students will use complex list algorithms such as sorting and searching. They will also understand, explain and use the concept of recursion, where a method calls itself until an exit condition is met. Students will further explain the need for cluster computing algorithms, where the need for improved performance and availability over a single computer exists. Students will explain the need for quantum computing algorithms and cryptography: where the existence of hidden problem structures can exploit in ways that classical computers cannot.

Beyond High School:
Students will look at how to describe the efficiency of algorithms to better compare them. Additionally, they will learn how advanced algorithms are crafted for tasks such as artificial intelligence and pathing.

IMPORTANT CONCEPTS

- A recursive method must simplify a problem and have an exit condition.
- Sorting and searching analyzes and organizes the data.
- Different sorting and searching algorithms work more efficiently with different sets of data.

KEY SKILL/PROCEDURES

- Describe how recursive algorithms can be more or less effective than iterative algorithms.
- Demonstrate how to rewrite simple algorithms from iterative to recursive.
- Explain Recursive Sorting algorithms.
- Explain Simple Sorting algorithms.
- Explain simple and binary searches.

Strand	Algorithmic Thinking and Programming
Topic	Algorithms

Content Elaborations

CLARIFICATIONS

Use visualization to show the various searches, the efficiency of access times and completion times.

Demonstrate the construction and logic behind a decimal to binary recursive function.

A simple sort does not utilize recursion.

CONTENT FOCUS

Focus on using already optimized algorithms for determining which should be used. Recursion should be understood and utilized, not necessarily created from scratch.

COMPUTER SCIENCE PRACTICES

Practice 4. Developing and Using Abstractions

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

2. Create a computational artifact for practical intent, personal expression or to address a societal issue.

Practice 6. Testing and Refining Computational Artifacts

3. Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability and accessibility.

Career Connections

CAREER PLANNING

Students research Quantum Computing Career Pathways using the Dynamic Career Pathways tool in OhioMeansJobs K-12. Students research the salaries and qualifications for Quantum Computing Careers. Students also practice mock interviewing for Quantum Computing Careers using questions such as:

- How is a qubit measured?
- What is a quantum logic gate?
- What is the difference between classical computing and quantum computing?

List some of the engineering challenges a quantum computer engineer might face.

Strand	Algorithmic Thinking and Programming
Topic	**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.

Expectations for Learning

LEARNING PROGRESSION

By the end of the 9-12 Foundational Level:
Students will understand that variables have different storage requirements and restrictions and will be able to choose the correct variable type to use for a task. Students will be able to use parameters to pass variable information into methods and return values to get information out of a method. Students will understand scope or when the program is able to access or change a variable.

In the 9-12 Advanced Level:
Student will learn how data can be organized using different structures. Students will learn that this allows more complex pieces of information to be passed into and out of methods. Additionally, they will learn what the benefits and costs are for these advanced data structures. They will also see how these data structures can be used as templates to create multiple instances of the structure.

Beyond High School:
Students will continue to learn how to store larger and more complex pieces of data. They will also investigate connecting to larger data stores like databases and how to store information from a database in a local data structure.

IMPORTANT CONCEPTS

- Data structures can be used to represent multiple variables as one item.
- Advanced data types are more appropriate for certain problems.
- Multiple similar pieces of data can be stored in a structure called an array.
- Variables and data structures can be passed into a method by reference or by value.

KEY SKILL/PROCEDURES

- Code the definition of a data structure and implement multiple instances.
- Use a data structure to pass information into or out of a method.
- Describe the benefits and costs of advanced data structures.
- Demonstrate how data structures can be passed to and from methods and objects.
- Organize like data into an array.

Strand	Algorithmic Thinking and Programming
Topic	**Variables and Data Representation**
	Content Elaborations **CLARIFICATIONS** Data structures could include classes, json objects, stucts, records and xml. Coding a definition of a data structure involves writing out all the code necessary to use it in a program. An example of implementation would be objects created from a data structure. Data structures are used to manage program complexity. Programmers choose data structures based on functionality, storage and performance tradeoffs. Passing a variable by reference points to its original memory location, and changes in the method will be reflected in the passed variable. Passing a variable by value will create a temporary local variable that can be changed without changing the passed variable. Not all languages support choose to pass by value or pass by reference. In many languages, this may be automatically determined based on the data type. **CONTENT FOCUS** Strings are basically 1D arrays. A typical data table is an example of a 2D array. A list is like an array, but elements are ordered in some manner and can be added and removed, varying the size of the list. The language being studied may dictate the data structures being used. **COMPUTER SCIENCE PRACTICES** *Practice 4*. Developing and Using Abstractions 1. Extract common features from a set of interrelated processes or complex phenomena. 3. Create modules and develop points of interaction that can apply to multiple situations and reduce complexity.

Strand	Algorithmic Thinking and Programming
Topic	**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.

Expectations for Learning

LEARNING PROGRESSION

By the end of the 9-12 Foundational Level:
Students will understand how to use both binary and unary Boolean statements to test a condition. They can use this to create if/then/else statements in code. They can also code the definition of a method and then use it to solve a problem. Students will understand how to create methods that have parameters and return values. Students know how control structures can be nested to form more readable complex control structures.

In the 9-12 Advanced Level:
Students will learn how to put together methods, control structures and data structures to form a well-written program. They will use methods to organize the code into readable parts, data structures to organize data for easier processing, and control structures to direct the flow of the program. They will also learn to identify and remove redundant and inefficient code from a program.

Beyond High School:
Students will expand their knowledge of methods and control structures by seeing how multiple programming languages handle them. They will also look at how programming languages turn these control structures into machine code.

IMPORTANT CONCEPTS

- Developers can use code from other sources to complete their programs.
- Code that appears in multiple places in a program can be put into a method for code reuse.
- Programs are made up of methods, control structures and data structures working together.

KEY SKILL/PROCEDURES

- Create and call multiple methods in a program.
- Write a program to solve a problem using methods, control structures and data structures.
- Identify and remove redundant and inefficient code from a program.

Content Elaborations

CLARIFICATIONS

Refactoring a program involves rewriting modules to make them more efficient by simplifying algorithms, removing inefficient code and using optimal control structures.

Strand	Algorithmic Thinking and Programming
Topic	**Control Structures**
	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. **CONTENT FOCUS** Key library methods include: • math-related functions (square root, sin, cos) and • input and output functions such as reading and writing to and from the console and files. Optional 3rd party libraries include graphical user interface tools. **COMPUTER SCIENCE PRACTICES** *Practice 4.* Developing and Using Abstractions 2. Evaluate existing technological functionalities and incorporate them into new designs. *Practice 5.* Creating Computational Artifacts 2. Create a computational artifact for practical intent, personal expression or to address a societal issue. *Practice 6.* Testing and Refining Computational Artifacts 3. Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability and accessibility.

Strand	Algorithmic Thinking and Programming	
Topic	**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.	(**For ATP.M.9-12.A.a**: See ATP.CS.9-12.F.b for model curriculum information) (**For ATP.M.9-12.A.b**: See ATP.PD.9-12.A.a for model curriculum information) (**For ATP.M.9-12.A.c**: See ATP.CS.9-12.A.a for model curriculum information)	

Strand	Algorithmic Thinking and Programming
Topic	**Program Development**

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

Expectations for Learning

LEARNING PROGRESSION

By the end of the 9-12 Foundational Level:
Students will be able to define different software development lifecycle methodologies, including phases, roles and workflows. Students will understand how unit, system and integration tests differ and what they are used for. They will also understand how common conventions, such as naming and commenting conventions, improve code fixes and readability. Students will also understand how source control can make finding errors easier by illustrating the changes that were made to the code.

In the 9-12 Advanced Level:
Students will learn how to implement a solution to a problem using a software methodology from inception through support. This will require them to learn how to communicate as a team and work within their assigned role. Additionally, they will see how the practices they have learned so far make the project implementation simpler.

Beyond High School:
Students will further their understanding by seeing how larger groups can operate effectively. They will see how multiple team projects can work to get complex tasks done very quickly. They will also understand some of the pitfalls and difficulties large scale projects hold, including legacy code, communication and data storage.

IMPORTANT CONCEPTS

- Software development lifecycle methodologies allow a team to work more efficiently.
- Software development lifecycle methodologies outline the steps for design, development, test, rollout and support for an application, and the roles of team members.
- Two or more programmers may be working on the same program at the same time, but on different sections. Defining coding standards and updating design documentation helps integrate the different methods each programmer is implementing.

KEY SKILL/PROCEDURES

- Solve a complex problem as a team using a single software methodology.
- Gather requirements from stakeholders.
- Collaborate with team members.

Strand	Algorithmic Thinking and Programming
Topic	**Program Development**

Content Elaborations

CLARIFICATIONS

Software methodologies include Waterfall, Agile, top-down, bottom-up and RAD.

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.

Collaboration involves

- a shared communication medium,
- a common vision of the end product,
- regularly scheduled meetings and
- a clear understanding of roles.

A stakeholder is a person with an interest or concern in the project.

CONTENT FOCUS

The focus here is to use a software methodology from beginning to end, enabling students to see the methodology in action. The students should see how specific jobs within the methodology support one another.

COMPUTER SCIENCE PRACTICES

Practice 5. Creating Computational Artifacts

1. Plan the development of a computational artifact using an iterative process that includes reflection on and modification of the plan, considering key features, time and resource constraints and user expectations.

Practice 6. Testing and Refining Computational Artifacts

1. Systematically test computational artifacts by considering all scenarios and using test cases.

Strand	Artificial Intelligence
Topic	**Machine Learning**

AI.ML.9-12.A.a Evaluate a data set used to train a real AI system by considering the size of the data set, the way that the data were acquired and labeled, the storage required and the estimated time to produce the data set.

Expectations for Learning

LEARNING PROGRESSION

By the end of grade 8:
Students will illustrate how objects in an image can be segmented and labeled to construct a training set for object recognition.

In the 9-12 Foundational Level:
Students will illustrate what happens during each of the steps required when using machine learning to construct a classifier or predictor.

In the 9-12 Advanced Level:
Students evaluate a data set used to train a real Artificial Intelligence (AI) system by considering the size of the data set, the way that the data were acquired and labeled, the storage required and the estimated time to produce the data set.

IMPORTANT CONCEPTS

- A large data set is typically required to capture the diversity of a complex domain and narrow down the range of possible reasoner behaviors.
- There are multiple ways to construct, clean and verify a data set.
- There can be large costs associated with creating the data set and processing the data.
- Labeling training data is labor intensive and may require specialized expertise like spotting disease in x-rays.
- Bias can be introduced during each step of the data set creation.

KEY SKILLS/PROCEDURES

- An activity can be done with common machine learning data sets found in repositories or publicly available demographic, economic or environmental data sets.

Content Elaborations

CLARIFICATIONS

Data sets for real-world problems may involve many features and the defining characteristics of a class may involve complex relationships among these features. To narrow down the class to be learned and distinguish it from millions of other possible classes, the learning algorithm must see many examples.

Strand	Artificial Intelligence	
Topic	**Machine Learning**	
		CONTENT FOCUS • Uses of Big Data • Analysis of data sets • Bias of data sets **COMPUTER SCIENCE PRACTICES** *Practice 3.1* Identify complex, interdisciplinary, real-world problems that can be solved computationally. At any level, students should be able to identify problems that have been solved computationally.
AI.ML.9-12.A.b Using a data visualization tool, investigate imbalances in training data in terms of gender, age, ethnicity or other demographic variables that could result in a biased model.		**Expectations for Learning** **LEARNING PROGRESSION** By the end of grade 8: Students will explain how the choice of training data shapes the behavior of the classifier to identify how bias can be introduced if the training set is not properly balanced. In 9-12 Foundational Level: Students will use either a supervised or unsupervised learning algorithm to train a model on real-world data, then evaluate the results. In the 9-12 Advanced Level: Students will investigate imbalances in training data in terms of gender, age, ethnicity or other demographic variables that could result in a biased model by using a data visualization tool. **IMPORTANT CONCEPTS** • Machine learning algorithms will take advantage of any imbalances or correlations in the training set that help lower the error rate. • If the data set is not representative, correlations can be misleading. **KEY SKILLS/PROCEDURES** • Brainstorm limitations to data. • Research current events that illustrate bias in Artificial Intelligence (AI) facial recognition software. **Content Elaborations** **CLARIFICATIONS** Data exploration to help students uncover imbalances or correlations can be done using histograms in spreadsheet software, or using any number of data visualization tools.

Strand	Artificial Intelligence
Topic	**Machine Learning**
	CONTENT FOCUS **Sources of bias in data sets** Imbalances of data sets from historical data like police databases from 1900 - 2000, due to race or gender bias. For example, in the 1970s, women needed a male cosigner (father or husband) to apply for a credit card. **COMPUTER SCIENCE PRACTICES** *Practices 3.1* Identify complex, interdisciplinary, real-world problems that can be solved computationally. At any level, students should be able to identify problems that have been solved computationally.

Strand	Artificial Intelligence
Topic	**Natural Interaction**

AI.NI.9-12.A.a Identify and debate the issues of AI and consciousness.

Expectations for Learning

LEARNING PROGRESSION

By the end of grade 8:
Students will critically analyze and discuss features that make an entity intelligent, including discussing differences between human, animal and machine intelligence to identify how machine intelligence varies from natural intelligence.

In the 9-12 Foundational Level:
Students will construct context-free grammar to parse simple languages and use language processing tools to construct a chatbot. Use sentiment analysis tools to extract emotional tone from text.

In the 9-12 Advanced Level:
Students will identify and debate the issues of Artificial Intelligence (AI) and consciousness.

IMPORTANT CONCEPTS

- AI consciousness affects those who interact with it.
- Bias, as an inherent human condition, is coded into the functionality of AI.
- Can computers "think"?

KEY SKILLS/PROCEDURES

- Identify examples of issues related to bias, perception, privacy and accuracy in artificial intelligence.

Content Elaborations

CLARIFICATIONS

Published in 1950, Alan Turing's paper "Computing Machinery and Intelligence" started the debate and the Strong AI Hypothesis. Edsger Dijkstra, in a 1984 paper, compared the question of whether machines can think with questions, such as "can submarines swim?" or "can airplanes fly?". Most will agree that airplanes can, in fact, fly but submarines can't swim.

CONTENT FOCUS

- Define the word "thinking."
- Research how computers think.
- Research how reliable computer decisions are.

COMPUTER SCIENCE PRACTICES

Practice 3.1 Identify complex, interdisciplinary, real-world problems that can be solved computationally.
At any level, students should be able to identify problems that have been solved computationally.

Strand	Artificial Intelligence
Topic	**Perception**

AI.P.9-12.A.a Describe some of the technical difficulties in making computer perception systems function well for diverse groups.

Expectations for Learning

LEARNING PROGRESSION

By the end of grade 8:
Students will describe how a vision system might exhibit cultural bias if it lacked knowledge of objects not found in the culture of the people who created it to create inclusive and equitable data sets.

In grades 9-12 Foundational Level:
Describe the limitations and advantages of various types of computer sensors.

In the 9-12 Advanced Level:
Students will describe some of the technical difficulties in making computer perception systems function well for diverse groups.

IMPORTANT CONCEPTS

- Facial Recognition software is only reliable for white males.
- Digital cameras are limited in their ability to create contrast on darker objects, limiting their ability to distinguish features.

KEY SKILLS/PROCEDURES

- Discuss bias created by programmers' limitations of data sets.
- Explore technological articles on why corporations are moving away from facial recognition software.
- Debate the ethics of whether or not historical data should be used in locating high-crime areas to increase police presence.

Content Elaborations

CLARIFICATIONS

Dark or low contrast facial features are harder to recognize than bright, high contrast features. Children's speech is in a higher register and less clearly articulated than adult speech.

CONTENT FOCUS

The focus is on the cultural bias of programmers, ethics and limited data sets.

COMPUTER SCIENCE PRACTICES

Practice 5.3 Modify an existing artifact to improve or customize it. At all grade levels, students should be able to examine existing artifacts to understand what they do.

Strand	Artificial Intelligence
Topic	**Perception**

AI.P.9-12.A.b Illustrate the abstraction hierarchy for speech understanding, from waveforms to sentences, showing how knowledge at each level is used to resolve ambiguities in the levels below.

Expectations for Learning

LEARNING PROGRESSION

By the end of grade 8:
Students will illustrate how sequences of words can be recognized as phrases, even if some of the words are unclear, by looking at how the words fit together to create a text recognition program.

In the 9-12 Advanced Level:
Students will illustrate the abstraction hierarchy for speech understanding, from waveforms to sentences, showing how knowledge at each level is used to resolve ambiguities in the levels below.

IMPORTANT CONCEPTS

The spoken language hierarchy is:

- waveforms
  - articulatory gesture
    - sounds
      - morphemes
        - words
          - phrases
            - sentences

KEY SKILLS/PROCEDURES

- Demonstrate how an intelligent agent can misunderstand spoken words if you change the tone or pitch of your voice.
- Explain how written punctuation placement determines the meaning of a statement.
  - Example: "Let's eat, grandma." vs "let's eat grandma."
- Discuss how spoken language take this into account.

Content Elaborations

CLARIFICATIONS

To go from noisy, ambiguous signals to meaning requires recognizing structure and applying domain knowledge at multiple levels of abstraction. A classic example: The sentences "How to recognize speech" and "How to wreck a nice beach" are virtually identical at the waveform level.

Strand	Artificial Intelligence
Topic	**Perception**
	CONTENT FOCUS How do chatbots determine responses to oral language vs. written language? Students are familiar with their smart devices such as phones and watches. They can change tone and accent while asking these devices various questions. **COMPUTER SCIENCE PRACTICES** *Practice 3.* Recognizing and Defining Computational Problems 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: • Identify complex, interdisciplinary, real-world problems that can be solved computationally. • Decompose complex real-world problems into manageable subproblems that could integrate existing solutions or procedures. • Evaluate whether it is appropriate and feasible to solve a problem computationally.

Strand	Artificial Intelligence
Topic	**Representation & Reasoning**
AI.RR.9-12.A.a Write code to create an algorithmic search.	**Expectations for Learning** **LEARNING PROGRESSION** By the end of grade 8: Students will model the process of solving a graph search problem using breadth-first search to draw a search tree. In grades 9-12 Foundational Level: Students will list an algorithm that could be used to solve a problem for each of these types of reasoning problems: classification, prediction, sequential decision making, combinatorial search, heuristic search, adversarial search, logical deduction and statistical inference. In the 9-12 Advanced Level: Students will write code to create an algorithmic search. **IMPORTANT CONCEPTS** • There are multiple algorithms for generating a search tree, each with its own advantages. **KEY SKILLS/PROCEDURES** • Categorizing problems. • Predicting outcomes. • Defining and using the various algorithms. **Content Elaborations** **CLARIFICATIONS** Some algorithms may be easier to implement in a particular programming language, work faster, require less memory to store data and be applicable in a wider variety of situations than other algorithms. Algorithms used to search and sort data are common in a variety of software applications. Encryption algorithms are used to secure data, and compression algorithms make data storage more efficient. At this level, the analysis may involve simple calculations of steps. Analysis using sophisticated mathematical notation to classify algorithm performance, like Big-O notation, is not expected. **CONTENT FOCUS** The focus of this content is on using code to create an algorithmic search. **COMPUTER SCIENCE PRACTICES** *Practice 6.3* Evaluate and refine a computational artifact multiple times to enhance its performance, reliability,usability and accessibility.

Strand	Artificial Intelligence
Topic	**Representation & Reasoning**

AI.RR.9-12.A.b Illustrate breadth-first, depth-first and best-first search algorithms to grow a search tree.

Expectations for Learning

LEARNING PROGRESSION

By the end of grade 8:
Students will model the process of solving a graph search problem using breadth-first search to draw a search tree.

In grades 9-12 Foundational Level:
Students will categorize real-world problems like classification, prediction, sequential decision problems, combination search heuristic search, adversarial search, logical deduction or statistical inference.

In the 9-12 Advanced Level:
Students will illustrate breadth-first, depth-first and best-first search algorithms to grow a search tree.

IMPORTANT CONCEPTS

- Breadth-first search finds shallow solutions quickly if they exist, but requires lots of memory.
- Depth-first search is better than breadth-first for cases where the number of nodes at each level grows exponentially because it uses less memory.
- A Binary Tree follows one simple rule that each parent node has no more than two child nodes, whereas, a Binary Search Tree is a variant of the binary tree that follows a relative order to how the nodes should be organized in a tree.

KEY SKILLS/PROCEDURES

- Illustrate breadth-first, depth-first and best-first search algorithms to grow a search tree for tic-tac-toe.
- Illustration is done by drawing the tree. Writing code is appropriate for advanced students (AP CSA/CSP).

Content Elaborations

CLARIFICATIONS

There are multiple algorithms for generating a search tree, each with its own advantages.

CONTENT FOCUS

Students should begin to evaluate and refine their computational artifacts after they have gained experience testing, debugging and revising.

COMPUTER SCIENCE PRACTICES

Practice 6.3 Evaluate and refine a computational artifact multiple times to enhance its performance, reliability,usability and accessibility.

Strand	Artificial Intelligence
Topic	**Societal Impacts**

AI.SI.9-12.A.a Design an AI system to address social issues or explain how AI could be used to address a social issue.

Expectations for Learning

LEARNING PROGRESSION

By the end of grade 8:
Students will understand tradeoffs in the design of Artificial Intelligence (AI) systems and how decisions can have unintended consequences in the function of a system.

In grades 9-12 Foundational Level:
Students will critically explore the positive and negative impacts of an AI system.

In the 9-12 Advanced Level:
Students will design an AI system to address social issues or explain how AI could be used to address a social issue.

IMPORTANT CONCEPTS

- What Can AI Do?
- Find trends, patterns and associations
- Discover inefficiencies
- Execute plans
- Learn and become better
- Predict future outcomes based on historical trends
- Inform fact-based decisions

KEY SKILLS/PROCEDURES

- Identify and research projects from organizations focusing on social issues and design potential solutions to the problems identified.
- Create an algorithm that will solve a social issue identified in student research.

Content Elaborations

CLARIFICATIONS

Technology has helped societies all over the world to battle the most pressing issues and solve social problems. By promising faster technological advancement, artificial intelligence (AI) promises to provide answers to questions relating to the environment, security and society that we are all exploring today.

Strand	Artificial Intelligence
Topic	**Societal Impacts**
	CONTENT FOCUS • Research how to solve societal issues. • Students can explore fact-checking AI. • Consider how computers make our work easier or harder. **COMPUTER SCIENCE PRACTICES** *Practice 6.3* Evaluate and refine a computational artifact multiple times to enhance its performance, reliability,usability and accessibility. After students have gained experience testing (P6.2), debugging and revising (P6.1), they should begin to evaluate and refine their computational artifacts.

Strand	Artificial Intelligence
Topic	**Representation & Reasoning**

AI.RR.9-12.F.b For each of these types of reasoning problems (classification, prediction, sequential decision making, combinatorial search, heuristic search, adversarial search, logical deduction and statistical inference), list an algorithm that could be used to solve that problem.

Expectations for Learning

LEARNING PROGRESSION

By the end of grade 8:
Students will model the process of solving a graph search problem using breadth-first search to draw a search tree.

In grade 9-12 Foundational Level:
Students will list an algorithm that could be used to solve a problem.

In grade 9-12 Advanced Level:
Students will illustrate breadth-first, depth-first and best-first search algorithms to grow a search tree.

IMPORTANT CONCEPTS

- Artificial Intelligence (AI) includes a wide variety of reasoning algorithms for solving different types of reasoning problems. Some algorithms use symbolic representations like search trees. Others are numerical in nature, such as neural nets operating on feature vectors.

KEY SKILLS/PROCEDURES

- Categorizing problems
- Predicting outcomes
- Defining and using the various algorithms

Content Elaborations

CLARIFICATIONS

Different algorithms can be used to tie shoes or decide which path to take on the way home from school. While the end results may be similar, they may not be the same. In the example of going home, some paths could be faster, slower or more direct depending on varying factors, like time or obstacles. Algorithms can be expressed in noncomputer languages, including natural language, flowcharts and pseudocode.

CONTENT FOCUS

AI includes a wide variety of reasoning algorithms for solving different types of reasoning programs. Some use symbolic representations while others are numerical in nature.

COMPUTER SCIENCE PRACTICES

Practice 3.1 Identify complex, interdisciplinary, real-world problems that can be solved computationally. At any level, students should be able to identify problems that have been solved computationally.

Strand	Impacts of Computing
Topic	**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 global impact of the distribution of computing resources in terms of equity, access and influence.

IC.Cu.9-12.A.c Design a study of the potential impacts of classical computers, clustered computing and quantum computing in different fields.

IC.Cu.9-12.A.d Evaluate and explore how research and commercial entities are using clustered and quantum computing as alternative solutions due to the limitations of classical computers.

Expectations for Learning

LEARNING PROGRESSION

By the end of the 9-12 Foundational Level:
Students practiced analyzing new technologies and professions, using appropriate criteria to be able to predict and recognize the true measurable impact on each part of our complex society.

In the 9-12 Advanced Level:
Students will be able to evaluate, analyze, adapt and make predictions about new technologies and professions to solve real-world situations. These real-world situations will include a wide variety of economic, scarcity, social and geopolitical factors. Students will focus on individual, local and global interactions to solve these higher-level real-world situations.

Beyond High School:
Students will continue to investigate the impacts of technology and computing resources on all aspects of our society and culture at the collegiate level.

IMPORTANT CONCEPTS

- Technology is not equally used or available around the world.
- There are often free and different alternatives to modern computing tools.
- It is important to account for scarcity when evaluating tools to solve a problem.
- Economic, social and geopolitical factors have influenced the computer's impact on the world.
- Computers have had a dramatic impact on our culture.
- Identify how emerging computer technologies are impacting how we imagine the future in both current research and cultural society.

KEY SKILL/PROCEDURES

- Evaluate a current computing tool's appropriateness for a specific problem.
- Compare and contrast computing tools based on appropriateness and scarcity.
- Describe some of the difficulties in distributing computing resources around the world.
- Explain how economic, social and geopolitical factors influence computing.
- Identify ways computation has revolutionized our culture.
- Predict how computers may impact our culture in the future.

Strand	Impacts of Computing
Topic	Culture

Content Elaborations

CLARIFICATIONS

Computer distribution differs by network availability, such as whether broadband is available, as well as hardware availability such as the number of households that have computers.

An example of an alternative solution would be a person who does not have international phone service, but wants to talk with people from home, using a free phone app to make an online video call when both ends are connected to Wi-Fi.

CONTENT FOCUS

Focus on how computer distribution is different across the world, and how people then need to solve computing problems differently depending on their available resources.

COMPUTER SCIENCE PRACTICES

Practice 1. Fostering an Inclusive Computing Culture

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.

Practice 2. Collaborating Around Computing

4. Evaluate and select technological tools that can be used to collaborate on a project.

Practice 7. Communicating About Computing

1. Select, organize and interpret large data sets from multiple sources to support a claim.

Strand	Impacts of Computing
Topic	**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 Evaluate and explore how research and commercial entities use intellectual property laws including copyright, trademarks, and patents to identify practical, business and ethical impacts.

Expectations for Learning

LEARNING PROGRESSION

By the end of the 9-12 Foundational Level:
Students investigated security concerns, data breaches, privacy policies, intellectual property laws, the inverse relationship of usability versus security, a user's legal rights and the Creative Commons license. All these investigations focused on both the legal and ethical impacts on our complex society.

In the 9-12 Advanced Level:
Students will further investigate the inverse relationship of usability versus security as well as all aspects of intellectual property laws. This will also include creating, solving and implementing solutions to a variety of real-world situations.

Beyond High School:
Students will continue to investigate the safety, law and ethics of digital interactions and data at the collegiate level.

IMPORTANT CONCEPTS

- Security measures must account for usability concerns to be successful.
- Security requires tradeoffs, both of usability and of other concepts, such as speed, size.
- Intellectual property laws include copyright, patent and trademark.
- Intellectual property laws have significant impact in the computing world.
- Intellectual property laws have ethical and legal implications for businesses, people and society.

KEY SKILL/PROCEDURES

- Create a realistic scenario that requires computer security.
- Explain what security tradeoffs are and how they work.
- Defend the tradeoffs of security and other concepts the student use.
- Define copyright and explain how it is protected.
- Explain what "free," "open-source" and "commercial" software types are and the differences between them.
- Argue the positive and negative impacts of copyrights.
- Describe how to research for existing intellectual property holdings.
- Evaluate and explore Terms of Use and Privacy Policies.

Strand	Impacts of Computing
Topic	Safety, Law and Ethics
	Content Elaborations **CLARIFICATIONS** Any time personally identifiable information is being shared on a computer there should be security involved, so any scenario that the students create that includes this would be acceptable. **CONTENT FOCUS** Focus on using the understanding of security and usability tradeoffs to solve a problem. **COMPUTER SCIENCE PRACTICES** *Practice 7.* Communicating About Computing 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.