Ohio’s Model Curriculum for Computer Science

Grade 9-12 Foundational

ADOPTED SEPT. 2022

Ohio | Department of Education

Strand	Computing Systems
Topic	**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 and existing limitations.

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

CS.D.9-12.F.d Evaluate alternative computing architectures for emerging technologies including cluster and quantum computing.

Expectations for Learning

LEARNING PROGRESSION

By the end of grade 8:
Students will be able to identify improvements to possibly make a computing device for better interaction with users.

In the 9-12 Foundational Level:
Students will extract technological information about computing systems and their limitations to determine effective solutions for given 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.

IMPORTANT CONCEPTS

- Different devices interact with each other.
- Everyday devices have computers in them
- Criteria can be used to evaluate computing systems.
- There are multiple methods to evaluate computing systems.
- Computing systems can consist of both virtual and physical components.

KEY SKILL/PROCEDURES

- Communicate/explain how different devices interact to determine the proper device for usage.
- Identify the different devices and their purpose to evaluate the proper device for usage.
- Create criteria to effectively evaluate a computing system.
- Apply a method of evaluation to verify functionality.

Strand	Computing Systems
Topic	**Devices**

Content Elaborations

CLARIFICATIONS

Examples of simple devices could be:

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

Examples of complex devices could be:

- tablet on a refrigerator,
- an alarm system and
- a multi-functional printer.

Examples of virtual devices could be:

- a cloud server,
- a remote server,
- a software driver emulating hardware and
- a virtual hard drive.

Examples of connecting technologies could be:

- wired networks and components,
- routers and switches,
- wireless network technologies and
- Bluetooth.

CONTENT FOCUS

Understand that everyday devices have computers

Understand how different devices communicate with each other

Understand the difference between virtual and physical

COMPUTER SCIENCE PRACTICES

Practice 4. Developing and Using Abstractions

1. Extract common features from a set of interrelated processes or complex phenomena

Strand	Computing Systems
Topic	**Hardware and Software**

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

Expectations for Learning

LEARNING PROGRESSION

By the end of grade 8:
Students will make decisions on the use of hardware and software combinations to effectively complete a task.

In the 9-12 Foundational Level:
Students 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.

IMPORTANT CONCEPTS

- Hardware must have software to function.
- Software consists of operating systems, firmware and applications.

KEY SKILL/PROCEDURES

- Evaluate the purpose of operating systems, firmware and applications to enable the hardware to operate as designed.

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,
- IoT and
- simple/complex (garage door opener/multi-function printer).

Firmware controls specific hardware (graphics card).

Applications provide a service for the end-user.

CONTENT FOCUS

Focus on understanding what operating systems are and how they work.

Strand	Computing Systems
Topic	**Hardware and Software**
	COMPUTER SCIENCE PRACTICES *Practice 2.* Collaborating Around Computing 3. Solicit and incorporate feedback from and provide constructive feedback to, team members and other stakeholders. 4. Evaluate and select technological tools that can be used to collaborate on a project.

Strand	Computing Systems
Topic	**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.

Expectations for Learning

LEARNING PROGRESSION

By the end of grade 8:
Students will be able to identify a problem, make a list of possible solutions to a problem, determine the best solution and also communicate with others to help them solve the problem.

In 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.

IMPORTANT CONCEPTS

- Systematic processes can be used to solve problems.
- Not all computer parts are compatible.

KEY SKILL/PROCEDURES

- Apply an existing systematic process to correct problems.
- Identify common problems with devices.
- Explain the compatibility of different components.

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, the user tests system.
6. Create documentation.

CONTENT FOCUS

Focus on using systematic processes to solve problems.

Understand how to analyze computer parts to identify problems.

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.F.a Evaluate and select networking devices to establish scalable communications.

NI.N.9-12.F.b Evaluate and select networking protocols for classical, clustered and quantum computing 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.

Expectations for Learning

LEARNING PROGRESSION

By the end of grade 8:
Students will be able to develop a model representing networking hardware to understand the flow of information. Students will be able to generally identify a wide range of protocols and how they are utilized to transfer data across the internet. Students will understand that information that is lost via transmission will be retransmitted.

In the 9-12 Foundational Level:
Students will be 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.

IMPORTANT CONCEPTS

- Network devices have connectivity.
- Protocols are used for network communications.
- Network topologies have limitations.
- Different network topologies can be interconnected.
- Devices have both a physical and logical address.

KEY SKILL/PROCEDURES

- Demonstrate how to connect to a network device.
- Identify the tools to trace data through a network.
- Explain how to trace data through a network.
- Compare and contrast network topologies.
- Compare and contrast network devices.
- Identify the network a device is connected to.
- Define the common network protocols.

Strand	Networks and the Internet
Topic	**Networking**
	Content Elaborations **CLARIFICATIONS** Examples of network devices could be: • hubs • switches • routers • firewalls • virtual servers This should include the network differences between classical, clustered and quantum computing. This should include all versions of network addressing. The scalability and reliability of networks can be evaluated by describing the relationship between routers, switches, servers, topology and addressing. **CONTENT FOCUS** IP addresses and how they relate to devices and the network. **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.

Strand	Networks and the Internet
Topic	**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 from both existing and emergent technologies, including cluster computing and quantum key distribution.

Expectations for Learning

LEARNING PROGRESSION

By the end of grade 8:
Students understand physical measures to protect devices and software measures to protect electronic information. Students understand general practices used to identify malware infections. Students can describe how malware can affect information.

In the 9-12 Foundational Level:
Students will be able to understand the cybersecurity impact on business and economy. Students will be able to employ principles of cybersecurity and emergent technologies including cluster computing and quantum key distribution.

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.

IMPORTANT CONCEPTS

- All endpoints must be protected.
- Physical security is the protection from physical actions and events that could cause loss or damage.
- Humans are the weakest point in cybersecurity.
- As emerging technologies evolve, so do the threats of cybersecurity attacks.

KEY SKILL/PROCEDURES

- Identify the goals, objectives and purposes of cybersecurity.
- Identify common risks, alerts and warning signs.
- Describe the concepts of malware attacks.
- Identify types of controls to eliminate malware from occurring.
- Demonstrate understanding of authentication, authorization and verification.

Content Elaborations

CLARIFICATIONS

Know a response plan to understand why it is needed.

Know the cybersecurity cycle to be able to create a response plan.

Apply authentication, authorization and verification to be able to control network usage.

Know common types of cybersecurity attacks to recognize and implement countermeasures.

Strand	Networks and the Internet
Topic	**Cybersecurity**
	Physical security is necessary to limit access to important hardware, making it more difficult to add malicious code to computers and network devices. Examples of physical security could include how and when to limit computer access to certain individuals, how to secure computers, and where to store backups and external hard drives. Quantum Key Distribution (QKD) is a quantum cryptography protocol for the secure sharing of information. It produces a shared secure key for encryption and decryption of messages. **CONTENT FOCUS** • Understand the cybersecurity cycle. • Create Response plans (i.e., plan & prepare, identify, containment, investigation, remediation, follow-up). • Understand and discuss common types of cybersecurity attacks. **COMPUTER SCIENCE PRACTICES** *Practice 3.* Recognizing and Defining Computational Problems 2. Decompose complex real-world problems into manageable subproblems that could integrate existing solutions or procedures. *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 3. Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability and accessibility.

Strand	Networks and the Internet
Topic	**Cybersecurity**

NI.C.9-12.F.c Compare and contrast examples of various threat actors, such as nation-states, cyber terrorist groups, organized crime or hacktivists.

NI.C.9-12.F.d Explore and utilize examples of encryption methods, e.g., Vigenère, Bacon's cipher, and Enigma.

Expectations for Learning

LEARNING PROGRESSION

In previous grades, students began to employ best practices for threat mitigation. By the end of the foundational level in grades 9-12, students will compare threat actors and threat vectors that are used. In previous grades, students should recognize different encryption methods. In future grades, students will be able to utilize various encryption techniques to protect data.

IMPORTANT CONCEPTS

- Threat actors are those parties who plan and execute threats.
- Threat vectors are ways threat actors enter a network or system.
- Encryption is a concept that allows information to help protect files, networks and systems.

KEY SKILLS/PROCEDURES

- Discuss threat vectors and the threat actors behind attacks.
- Discuss how encryption helps to secure files, networks and systems.

Content Elaborations

CLARIFICATIONS

Threat actors could be nation-states, cyber terrorists, hacktivists or organized crime.

CONTENT FOCUS

By the end of grade 12, students should understand what data they should share and when it should be shared. They should exhibit good digital citizenry as it applies to cyber security hygiene, awareness and safe browsing. They also should know how to connect safely to online services using a Virtual Private Networking (VPN).

Students also should understand that information and data can be protected using encryption methods and other processes. This includes the idea of the CIA triad, a concept that focuses on the balance between the confidentiality, integrity and availability of data under the protection of your information security program.

COMPUTER SCIENCE PRACTICES

Practice 3. Recognizing and Defining Computational Problems

1. Identify complex, interdisciplinary, real-world problems that can be solved computationally.

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

NI.IOT.9-12.F.a Design 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.

NI.IOT.9-12.F.b Explore and plan career pathways related to IoT to identify careers associated with the computer science field.

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 can gather information to be able to design simple smart applications and an accompanying lifecycle.

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.

IMPORTANT CONCEPTS

- Smart devices with sensors are a part of the IoT life cycle.
- 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, and determine 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.

Strand	Data and Analysis
Topic	**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.

Expectations for Learning

LEARNING PROGRESSION

By the end of grade 8:
Students will consistently be able to choose and use an appropriate data tool for any given set of data. Students should be able to explain and utilize different types of file formats and explain the capacity limits of each file. Students will understand that files are stored in a root directory on the hard drive or in the cloud with folders and subfolders and be able to appropriately share files.

In the 9-12 Foundational Level:
Students will use scientific practices to select data analysis tools 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.

In the 9-12 Advanced Level:
Students will apply basic techniques for locating, collecting, manipulating and interpreting small-scale multidimensional data sets (e.g., creating and distributing 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).

IMPORTANT CONCEPTS

- Scientific practices are used for data analysis.
- Data collection tools can be used to find accurate data.
- Data analysis tools are used to interpret and investigate the data.
- There are different ways to represent data in a computer system that have varying costs and benefits.
- Data can be stored locally or on the network.

KEY SKILL/PROCEDURES

- Use data analysis tools to organize data and identify trends, patterns and outliers.
- Use scientific practices to answer questions about real-world challenges; gather, analyze and interpret data to develop models and explanations.
- Identify an appropriate data collection tool for a given situation.
- Demonstrate understanding of how data is stored on different computer systems.

Strand	Data and Analysis
Topic	Data Collection and Storage

Content Elaborations

CLARIFICATIONS
Scientific practices are a methodology that focuses on investigation, evaluation and developing explanations of the natural world.

Data collection could include data found from:

- spreadsheets,
- qualitative surveys,
- quantitative surveys, and
- experimental data.

Data analysis tools could include:

- filters,
- pivot tables,
- similarity matching,
- regression, and
- classifications.

Ways of representing data could include:

- ASCII,
- UTF,
- decimal,
- binary, and
- hexadecimal.

Ways of encoding data could include:

- XML,
- databases,
- spreadsheets, and
- flat files.

CONTENT FOCUS
Focus on extracting data into a data storage tool like a spreadsheet, database or XML file and using tools to find useful information from the data

Focus as well on how data and data structures can be stored on different systems.

Strand	Data and Analysis
Topic	**Data Collection and Storage**
	COMPUTER SCIENCE PRACTICES *Practice 3.* Recognizing and Defining Computational Problems 1. Identify complex, interdisciplinary, real-world problems that can be solved computationally. *Practice 4.* Developing and Using Abstractions 1. Extract common features from a set of interrelated processes or complex phenomena. *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.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.

Expectations for Learning

LEARNING PROGRESSION

By the end of grade 8:
Students will be able to take raw data and make a model from that data. Students then will be able to analyze the data in their model to determine patterns. Students will be able to utilize functions and formulas and will be able to create graphs from a spreadsheet that represent collections of data.

In the 9-12 Foundational Level:
Students will evaluate the ability of multisensory data artifacts to identify key features and relations by using scientific practices.

In the 9-12 Advanced Level:
Students will create data visualization artifacts using multiple data sources in order to analyze and answer real-world questions.

IMPORTANT CONCEPTS

- Multisensory artifacts, such as visualizations, are used to analyze data.
- Different multisensory artifacts have varied benefits and limitations.
- Multisensory artifacts can be used to look at real-world data and suggest solutions from the data.
- Different data analysis tools, including virtual tools, have varied advantages and disadvantages.

KEY SKILL/PROCEDURES

- Explain data analyzation artifacts.
- Analyze multisensory data artifacts to interpret their information.
- Communicate complex information gathered from analysis.

Content Elaborations

CLARIFICATIONS

A multisensory artifact could be:

- data visualizations,
- audio analysis,
- tactile analysis, and
- olfactory analysis.

Strand	Data and Analysis
Topic	**Visualization and Communications**
	A data visualization artifact could be: • a web page, • a chart, or • a graph. A web page would utilize HTML and CSS to create a data visualization. **CONTENT FOCUS** Focus on creating graphs or charts to analyze a data set that is appropriate for the target audience. Present the created graphs or charts to relay the relevant information to the target audience. **COMPUTER SCIENCE PRACTICES** *Practice 1.* Fostering an Inclusive Computing Culture 2. Address the needs of diverse end-users during the design process to produce artifacts with broad accessibility and usability. *Practice 4.* Developing and Using Abstractions 4. Model phenomena and processes and simulate systems to understand and evaluate potential outcomes. *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.F.a Evaluate a model by creating a hypothesis, testing it and refining it to discover connections and trends in the data.

Expectations for Learning

LEARNING PROGRESSION

By the end of grade 8:
Students will be able to evaluate a model associated with a concept to extract meaning.

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 to construct a model, create a hypothesis, test it and refine it to discover connections and trends between the multiple sets of data.

IMPORTANT CONCEPTS

- Scientific practices should be used to analyze data in a model.
- Data can be both valid and invalid.
- There are many different types of data models that can be utilized.
- There is a proper notation to use in a simple model.

KEY SKILL/PROCEDURES

- Create a simple model from given data.
- Evaluate data for validity.
- Demonstrate an understanding of fundamental data design techniques.
- Identify common model notations.
- Evaluate the appropriateness of common data model types.

Content Elaborations

CLARIFICATIONS

Below, are steps to model data:

- Identify entity types.
- Identify attributes.
- Apply naming conventions.
- Identify relationships.
- Assign keys.

Examples of data model types include hierarchical and relational.

Strand	Data and Analysis
Topic	**Inference and Modeling**
	Analyzation is to deconstruct an argument; examine it critically to bring out essential elements. Below, are Common Model notations: • Multiplicities (e.g., zero to one, one only, zero or more, one or more, specific range) • Attributes (e.g., names, primary keys/unique identifier, foreign key) • Associations (e.g., labels) **CONTENT FOCUS** The focus should be on creating a model from a set of data. **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.

Strand	Algorithmic Thinking and Programming
Topic	**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.

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

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

Expectations for Learning

LEARNING PROGRESSION

By the end of grade 8:
Students have a basic understanding of what goes into an algorithm and know how to write pseudocode. Students are able to analyze various solutions to determine the most efficient. Students are able to identify the parts of a program's pseudocode (input, output, decisions).

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

In the 9-12 Advanced Level:
Students will progress to learn about complex list algorithms such as sorting and searching. They will also understand and explain the concept of recursion, where a method calls itself until an exit condition is met."

IMPORTANT CONCEPTS

- Algorithms are the steps to solve a problem.
- There are strategies that simplify problem-solving.
- Problems can be broken down into smaller parts, such as inputs, outputs, decisions, loops and calculations.
- The flow of information through a program can be represented visually.
- Steps of a computer program can be represented by a diagram.

KEY SKILL/PROCEDURES

- Demonstrate how to break a problem into steps.
- Create a process flow diagram, data flow diagram and pseudocode for an algorithm.
- Explain the steps of a problem.
- Implement an algorithm in an appropriate programming language.
- Optimize an algorithm by removing redundant or repeated code and using alternative control structures.
- Use control structures to create an iterative algorithm.
- Use control structures to create a recursive algorithm.

Strand	Algorithmic Thinking and Programming
Topic	**Algorithms**
	Content Elaborations **CLARIFICATIONS** Algorithms are the steps to solve a problem. A complex problem is one in which the solution requires multiple (i.e., three or more) methods, objects and/or data structures. Problems can be broken down into smaller parts. The flow of information through a program can be represented visually. Steps of a computer program can be represented by a diagram. Present programming languages to students without extra functionality added by libraries, modules, etc. For example, using JavaScript without additional libraries like jQuery or React. A problem-solving strategy has the following steps: • Identify the problem. • Analyze the problem. • Identify decision criteria. • Develop multiple solutions. • Choose the optimal solution. **CONTENT FOCUS** The most important concept is the breaking down of problems into smaller pieces. You should also focus on at least one way of visualizing the algorithm. You should also focus on one programming language to implement the algorithm. **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.

Strand	Algorithmic Thinking and Programming
Topic	**Algorithms**
	Practice 5. Creating Computational Artifacts 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. *Practi*ce 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	**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.

Expectations for Learning

LEARNING PROGRESSION

By the end of grade 8:
Students will understand what it means to vary (opposed to being constant) and that this abstract value is expressed with an alphanumeric representation. Students will be able to determine the range of input values for variables. Students will be able to create, use and trace variables. Students will be able to identify a collection of information that can be represented by a single identifier.

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

In the 9-12 Advanced Level:
Student will move on to learn how data can be organized using different structures. 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.

IMPORTANT CONCEPTS

- Variables are used to store data in programs.
- Different types of variables store different types of data.
- There are memory and performance implications for different types of variables.
- Variables exist within a scope that is the block of code in which they are defined.
- Variables can be used to pass data and return data to and from methods.
- When primitive variables are passed to and from methods, the data is copied from one variable into another.

KEY SKILL/PROCEDURES

- Explain the different sizes and restrictions of different data types.
- Use correct data types for the storage of information.
- Use parameters and return types to pass information into and out of methods.

Strand	Algorithmic Thinking and Programming
Topic	**Variables and Data Representation**
	Content Elaborations **CLARIFICATIONS** A constant literal is not a variable as it cannot be changed. Identify primitive data types (i.e., integer, Boolean, floating-point). Information should be stored in the smallest possible data type. It should also be stored in the correct data type (e.g., characters should be stored in a character data type). Programmers create variables to store data values of selected types. A meaningful identifier is assigned to each variable to access and perform operations on the value by name. Variables enable the flexibility to represent different situations, process different sets of data, and produce varying outputs. **CONTENT FOCUS** Integer data includes integer, long integer, unsigned integer, character and byte. Floating-point includes float and double. A string is not primitive as it is a collection of characters. Using the correct data type allows efficiency in memory management by using just enough memory storage without overflow or underflow. **COMPUTER SCIENCE PRACTICES** *Practice 4.* Developing and Using Abstractions 1. Extract common features from a set of interrelated processes or complex phenomena.

Strand	Algorithmic Thinking and Programming
Topic	**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.

Expectations for Learning

LEARNING PROGRESSION

By the end of grade 8:
Students will understand variables and how to use a simple conditional and loop. Students will understand and incorporate proper processes, loops and conditionals in programs to solve problems.

In the 9-12 Foundational Level:
Students will look at Boolean statements to test a condition. They will then use this to create if/then/else statements in code. They will also code the definition of a method and then use it to solve a problem. Students will then expand their understanding to include methods that have parameters and return values. Students will also see how control structures can be nested to form more readable complex control structures.

In the 9-12 Advanced Level:
Students will move on to 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.

IMPORTANT CONCEPTS

- An IF Statement conditionally executes a section of code based on the outcome of a Boolean statement.
- A Boolean statement compares one or more items using comparison and logical operators and returns true or false.
- Code can be executed repeatedly using loops.
- Methods can be used to organize and allow the reuse of code.
- Scope determines when and where a variable can be accessed.
- Variables can be used to pass data and return data through methods.
- Decimal data could be too precise to be used in Boolean statements and cannot be used in switch selection.

Strand	Algorithmic Thinking and Programming
Topic	Control Structures
	KEY SKILL/PROCEDURES • Write an IF statement to compare one or more items and execute code depending on whether it is true or false. • Apply iteration (loops) to solve a programming problem. • Use logical operators in compound conditional statements. • Code the definition of a method. • Call a method to solve a problem. • Use parameters and return types to pass information into and out of methods. • Nest same and different control structures. **Content Elaborations** **CLARIFICATIONS** Examples of control structures could be: • if-else, • switch, • loops, • method definitions, or • calling methods. A switch selects a section of code to execute based upon the value of a variable. Coding a definition of a method involves writing out the all the code necessary to use it in a program. Comparison operators evaluate the relation between two objects and return TRUE or FALSE. Logical operators evaluate the relation between multiple statements and return TRUE or FALSE. NOT reverses the Boolean outcome. For example, if a Boolean outcome results in TRUE, a NOT would result in comparing it to FALSE. If a Boolean outcome is FALSE, a NOT would result in comparing it to TRUE. Programmers consider tradeoffs related to implementation, readability and program performance when selecting and combining control structures. Programs use procedures to organize code, hide implementation details and make code easier to reuse. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability.

Strand	Algorithmic Thinking and Programming
Topic	**Control Structures**
	CONTENT FOCUS Below, are key control structures that should be addressed (syntax and practice): • If-else • For and while loops • Method declaration • Calling methods • Comparison and logical operators Understanding program flow (defining and calling methods) is very difficult for new learners. Spend a lot of time on simple method declarations and method calls. **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. *Practice 5.* Creating Computational Artifacts 2. Create a computational artifact for practical intent, personal expression or to address a societal issue.

Strand	Algorithmic Thinking and Programming	
Topic	**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. (Oklahoma L1.AP.M.02) 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.	(See ATP.A.9-12.F.b for model curriculum information)	

Strand	Algorithmic Thinking and Programming
Topic	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.

Expectations for Learning

LEARNING PROGRESSION

By the end of grade 8:
Students will be able to design and create their own code-based program using transitional programming languages. Students will be able to identify run-time and logical errors and modify and test code to make the program robust. Students will be experienced using pre-built procedures that have one or more parameters.

In the 9-12 Foundational Level:
Students will define different software development lifecycle methodologies, including phases, roles and workflows. Students will also see how unit, system and integration tests differ, and what they are used for. They will also see how common conventions, such as naming and commenting conventions, will improve code fixes and readability. Students will also see 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 then 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."

IMPORTANT CONCEPTS

- A software methodology is a series of phases to manage the design, development, testing and support of a software project.
- Different software methodologies have different roles and organization of team members to complete a project.
- Testing methodologies specify when and how coding is tested.
- Consistent naming conventions, version control and comments allow for easier maintenance and reuse of code.

Strand	Algorithmic Thinking and Programming
Topic	Program Development

KEY SKILL/PROCEDURES

- Define different software methodologies, including phases, roles and workflows.
- Utilize unit test, system test and integration test to validate an application.
- Utilize print statements, logging statements and debuggers to identify and correct issues.
- Define and follow a naming and commenting convention.
- Use source control to maintain versions of code.
- Use appropriate standard industry language when talking about project management.

Content Elaborations

CLARIFICATIONS

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

Version control software is software that maintains revisions and changes to code. Examples would include GIT, TFS and SVN.

People design meaningful solutions for others by defining problems, criteria and constraints, carefully considering the diverse needs and wants of the community and testing whether criteria and constraints were met.

CONTENT FOCUS

Key software methodologies are the Waterfall Model and the Agile Development Model. The steps in both differences in approach should be discussed in this phase.

GIT is the most common version control structure used by colleges and students should know how to use it to save, see and merge changes to files.

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.
2. Identify and fix errors using a systematic process.

Strand	Artificial Intelligence
Topic	**Machine Learning**

AI.ML.9-12.F.a Illustrate what happens during each of the steps required when using machine learning to construct a classifier or predictor.	**Expectations for Learning**
	LEARNING PROGRESSION
	By the end of grade 8:
	Students explain the difference between training and using a reasoning model to identify how a machine learns.
	
	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 will 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. Artificial Intelligence (AI), is the science and engineering of making intelligent machines and computer programs.
	
	IMPORTANT CONCEPTS
	• Decide on a problem to solve.
	• Discover where to obtain the training data.
	• Choose a feature set.
	• Discover how to label the data.
	• Run the learning algorithm.
	• Use cross-validation set to decide when training should stop.
	• Use a test set to measure performance.
	
	KEY SKILLS/PROCEDURES
	• Outline the steps of machine learning and elaborate on what is happening.
	  ○ How the machine learns the information.
	  ○ How the information is being stored.
	  ○ How the machine changes or modifies its existing algorithm.

Strand	Artificial Intelligence
Topic	Machine Learning

KEY SKILL/PROCEDURES

- Define different software methodologies, including phases, roles and workflows.
- Utilize unit test, system test and integration test to validate an application.
- Utilize print statements, logging statements and debuggers to identify and correct issues.
- Define and follow a naming and commenting convention.
- Use source control to maintain versions of code.
- Use appropriate standard industry language when talking about project management.

Content Elaborations

CLARIFICATIONS

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

Version control software is software that maintains revisions and changes to code. Examples would include GIT, TFS and SVN.

People design meaningful solutions for others by defining problems, criteria and constraints, carefully considering the diverse needs and wants of the community and testing whether criteria and constraints were met.

CONTENT FOCUS

Key software methodologies are the Waterfall Model and the Agile Development Model. The steps in both differences in approach should be discussed in this phase.

GIT is the most common version control structure used by colleges and students should know how to use it to save, see and merge changes to files.

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.

2. Identify and fix errors using a systematic process.

Strand	Artificial Intelligence
Topic	**Machine Learning**
	Content Elaborations **CLARIFICATIONS** The cross-validation set is used to avoid overfitting. The test set consists of examples that were not used during training or for cross-validation, so it provides an unbiased prediction of the reasoner's performance on new inputs. **CONTENT FOCUS** • The origins of the data set. • How the data is being used. • How the machine will make decisions or modify the existing algorithm as it learns more information. **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.F.b Use either a supervised or unsupervised learning algorithm to train a model on real-world data, then evaluate the results.	**Expectations for Learning** **LEARNING PROGRESSION** By the end of grade 8, students will train a classification or prediction model using machine learning on a tabular data set to evaluate the quality of the data set. In the grades 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 grades 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** • The supervised learning model is trained on a training set to produce the correct labels for labeled data. Students should evaluate the results by measuring the percent of items in a test set that are labeled correctly. • The unsupervised learning model is trained to assign each input to a cluster of similar inputs. The clusters are determined by the learning algorithm because there are no labels attached to the training data. Students should evaluate the results by examining the clusters to see if they capture useful distinctions in the data set.

Strand	Artificial Intelligence
Topic	Machine Learning

KEY SKILLS/PROCEDURES

Supervised Learning Algorithm

Example: Sort animal images into categories and teach a machine how to sort the images. Then, expose the machine to new images and quantify the results based on how many correct labels were applied to the new images.

Unsupervised Learning Algorithm

Example: Teach the machine to sort images into clusters based on useful distinctions, such as two legs, long tails, etc. Do not assign the images to distinct categories. Quantify the results by evaluating the contents of each cluster.

Content Elaborations

CLARIFICATIONS

Supervised and unsupervised learning algorithms find patterns in data. In supervised learning, the "pattern" is the relationship between feature values and class labels. In unsupervised learning, the pattern is the way data is grouped into clusters. Real-world data sets are now widely available on the web. In earlier grade bands, students might test their trained models on a few new data points, but in this grade band, students are asked to quantitatively measure the performance of a trained model on a nontrivial test set.

CONTENT FOCUS

Using a search engine, conduct a search for "machine learning tools." Choose one that is free and compatible with the 1:1 devices used in the classroom. Model how the tool is used and have the students try to train the tool in various manners.

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.

Career Connections

CAREER PLANNING

Students research Artificial Intelligence career pathways using the Dynamic Career Pathways tool in OhioMeansJobs K-12 and identify qualifications (education, certification and experience) and career progression (entry, intermediate, advanced and expert level progression). Students examine how training data with models and discovering trends and relationships in a data set are used in these careers.

Strand	Artificial Intelligence
Topic	**Natural Interaction**

AI.NI.9-12.F.a 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.

Expectations for Learning

LEARNING PROGRESSION

By the end of grade 8:
Students will create a program, individually and collaboratively, that implements a language processing algorithm to create a functional chatbot.

In the 9-12 Foundational Level:
Students will construct context-free grammar to parse simple languages, use language processing tools to construct a chatbot and 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

- Context-free grammars (CFGs) are a set of recursive rules used to generate patterns of strings and to describe context-free languages. CFGs can describe all regular languages and more, but they cannot describe all possible languages.
- Sentiment analysis (or opinion mining) is a natural language processing (NLP) technique used to determine whether data is positive, negative or neutral.

KEY SKILLS/PROCEDURES

- Describe Context-Free Grammar (CFG) and how the weighted and probabilistic extensions handle the ambiguity of natural language.
- Write an algorithm to create a chatbot.
- Incorporate a sentiment analysis tool into an algorithm to allow the machine to determine whether the data is positive, negative or neutral to generate its response.

Content Elaborations

CLARIFICATIONS

Context-free grammars (CFGs) are a set of recursive rules used to generate patterns of strings and to describe context-free languages. CFGs can describe all regular languages and more, but they cannot describe all possible languages. Sentiment analysis (or opinion mining) is a natural language processing (NLP) technique used to determine whether data is positive, negative or neutral.

Strand	Artificial Intelligence	
Topic	**Natural Interaction**	
		CONTENT FOCUS Create a Chatbot that will respond to inputs based on the sentiment of inputed data. After students have gained experience testing (P6.2), debugging and revising (P6.1), they should begin to evaluate and refine their computational artifacts. **COMPUTER SCIENCE PRACTICES** *Practice 6.3* Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability and accessibility.
AI.NI.9-12.F.b Demonstrate how sentence parsers handle ambiguity.		**Expectations for Learning** **LEARNING PROGRESSION** By the end of grade 8: Students will create a program, individually and collaboratively, that implements a language processing algorithm to create a functional chatbot. In the 9-12 Foundational Level: Students will demonstrate how sentence parsers handle ambiguity. **IMPORTANT CONCEPTS** • Parsing is an NLP task to assign syntactic structures to sentences according to grammar. **KEY SKILLS/PROCEDURES** • Differentiate between how humans and Artificial Intelligence (AI) learn language ambiguities. **Content Elaborations** **CLARIFICATIONS** This was mentioned under AI Perceptions with the differences between written and spoken language. Students should begin to evaluate and refine their computational artifacts after they have gained experience testing, debugging and revising. **CONTENT FOCUS** Machine learning can overcome language ambiguities. **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	**Natural Interaction**

AI.NI.9-12.F.c Describe how artificial intelligence drives many software and physical systems.

Expectations for Learning

LEARNING PROGRESSION

By the end of grade 8:
Students 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 describe how artificial intelligence drives many software and physical systems.

IMPORTANT CONCEPTS

- Artificial Intelligence (AI) is a component of smart homes, smartwatches and smartphones.

KEY SKILLS/PROCEDURES

- Brainstorm areas where AI exists in students' everyday encounters.
- Students will choose one area and present to the class how AI is used in that area.

Content Elaborations

CLARIFICATIONS

Examples include digital ad delivery, self-driving cars and credit card fraud detection.

CONTENT FOCUS

AI is everywhere.

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	**Perception**

AI.P.9-12.F.a Explain how radar, lidar, GPS and accelerometer data are represented.

Expectations for Learning

LEARNING PROGRESSION

By the end of grade 8:
Students explain how sounds and images are represented digitally in a computer to explain how sensor data is stored in a computer.

In the 9-12 Foundational Level:
Students will explain how radar, lidar, GPS and accelerometer data are represented.

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

- Radar and lidar do depth imaging; each pixel is a depth value.
- GPS triangulates position using satellite signals and gives a location as longitude and latitude.
- Accelerometers measure acceleration in three-orthogonal dimensions.

KEY SKILLS/PROCEDURES

- Define radar, lidar, GPS and accelerometers.
- Determine if this data is a string, numeric as integer or decimal or boolean.
- Choose one sensor device to research how the data is stored on the computer.
- Students should report findings to the whole class.
- Students should know how to explain the complex concepts in simple language and should add to their report based on questions from the class.

Content Elaborations

CLARIFICATIONS

Radar and lidar measure distance as the time for a reflected signal to return to the transceiver. GPS determines position by triangulating precisely timed signals from three or more satellites. Accelerometers use orthogonally oriented strain gauges to measure acceleration in three dimensions.

<table>
<tr><th>Strand</th><th colspan="2">Artificial Intelligence</th></tr>
<tr><th>Topic</th><th colspan="2">Perception</th></tr>
<tr><td></td><td colspan="2">CONTENT FOCUS
Teachers should focus on deconstruction activities that help students analyze and interpret how sensors operate and how the computer stores data. The Artificial Intelligence (AI) components of the sensors will make decisions based on the information.

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.</td></tr>
<tr><td>AI.P.9-12.F.b Describe the limitations and advantages of various types of computer sensors.</td><td colspan="2"><h2>Expectations for Learning</h2>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 the 9-12 Foundational Level:
Students will describe the limitations and advantages of various types of computer sensors.

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
Sensors are devices that measure physical phenomena such as light, sound, temperature or pressure.
KEY SKILLS/PROCEDURES
Brainstorm types of computer sensors and the type of data they collect.
Create a table, Venn diagram or another graphic organizer to illustrate the advantages and disadvantages of each type of sensor.
Troubleshoot one sensor to provide a possible solution to a disadvantage.
<h2>Content Elaborations</h2>CLARIFICATIONS
Cameras have limited resolution, dynamic range and spectral sensitivity. Microphones have limited sensitivity and frequency response. Signals may be degraded by noise, such as a microphone in a noisy environment. Some sensors can detect things that people cannot like infrared or ultraviolet imagery or ultrasonic sounds.</td></tr>
</table>

Strand	Artificial Intelligence
Topic	**Perception**
	CONTENT FOCUS The focus is on 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	**Representation & Reasoning**

AI.RR.9-12.F.a Categorize real-world problems as classification, prediction, sequential decision problems, combination search, heuristic search, adversarial search, logical deduction or statistical inference.

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 categorize problems like classification, prediction, combinatorial search or sequential decision problems.

In the 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.

IMPORTANT CONCEPTS

- Reasoning problems can be categorized based on:
  - Types of inputs supplied,
  - Types of outputs to be produced, and
  - Characteristics of the search space, if applicable.

KEY SKILLS/PROCEDURES

- Define combination search, heuristic search, adversarial search, logical deduction and statistical inference as it relates to computer science.
- Give students examples of different real-world problems to categorize.
- Students will justify their reasoning based on their knowledge of the categories.

Content Elaborations

CLARIFICATIONS

A heuristic search is needed when the state space is too large to examine all possible states. It uses a rule of thumb to limit the search by focusing on the most promising states. In an adversarial search used in a game-playing scenario, the algorithm alternates between finding the best move for the player, the best response for the opponent and which would be the worst move from the player's perspective. An adversarial search may require heuristics if the game is complex, such as chess or go.

The reasoner can apply inference rules to reach a logical deduction by starting with a set of facts to derive new facts. The logical deduction can be done using formal logic such as propositional or predicate logic, or ad hoc inference rules used with semantic networks or the IF-THEN rules found in expert systems. Statistical inference involves reasoning with probabilities.

Strand	Artificial Intelligence
Topic	**Representation & Reasoning**
	CONTENT FOCUS Reasoning problems can be categorized based on the types of inputs supplied, the types of outputs to be produced and the characteristics of the search space, if applicable. **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.RR.9-12.F.c Describe the differences between types of search algorithms.	**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 the 9-12 Foundational Level: Students will describe the differences between types of search algorithms. 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** • There are multiple algorithms for generating a search tree, each with its own advantages. **KEY SKILLS/PROCEDURES** • Categorize problems. • Predict outcomes. • Define and use 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.

Strand	Artificial Intelligence	
Topic	Representation & Reasoning	
		CONTENT FOCUS There are multiple algorithms for generating a search tree, each with its own advantages. Students should begin to evaluate and refine their computational artifacts after they have gained experience testing, debugging and revising. **COMPUTER SCIENCE PRACTICES** *Practices 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.F.a Critically explore the positive and negative impacts of an AI system.	**Expectations for Learning** **LEARNING PROGRESSION** By the end of grade 8: Students will identify and explain how the composition of training data affects the outcome of a supervised artificial intelligence system. In the 9-12 Foundational Level: Students will critically explore the positive and negative impacts of an Artificial Intelligence (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** • Weak AI vs strong AI. • Robots replacing humans in the workforce. • Overcoming the innate bias of creating AI. • Ethics debates based on "just because we can, should we?" **KEY SKILLS/PROCEDURES** • Investigate and propose solutions to ethical and societal AI issues in a variety of settings like public safety, finance, social media marketing or government use. • Discuss issues of bias and accessibility in the design of existing technologies. • Using a web tool that trains a machine learning model without coding, investigate examples of bias and identify solutions. • Identify and analyze examples of legal policies related to AI, including why and how they were or are being developed. • Analyze real-world AI scenarios to determine the ethical and legal implications.

Strand	Artificial Intelligence
Topic	Societal Impacts
	Content Elaborations **CLARIFICATIONS** Understand the positives aspects of AI including: • Less time and • Solves problems that humans cannot. Understand the negative aspects of AI including: • Ethics of using biased data sets, • Loss of jobs and • Price of the technology. **CONTENT FOCUS** Students can have an ethical debate about AI. It is important to discuss the impacts that AI is having on society and develop criteria for the ethical design and deployment of AI-based systems **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	Impacts of Computing
Topic	**Culture**

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

IC.Cu.9-12.F.b Identify how existing and emerging computing architecture has and will impact other professions, both positively and negatively.

Expectations for Learning

LEARNING PROGRESSION

By the end of grade 8:
Students will understand the impact of computing on our country and the global economies. Students have identified technologies that have bias and accessibility issues for many areas of the world; persons with disabilities and those of a lower social economic status.

In the 9-12 Foundational Level:
Students will analyze new technologies to be able to predict and recognize their impact on society, economy and culture.

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.

IMPORTANT CONCEPTS

- Technology can have positive and negative effects on society.
- Technology can often have unintended effects on society.
- New technologies should be studied in ways that measure their impact on society.
- Computers have had a major impact on many professions, not only computer programming jobs.
- Computer science skills are important to many different professions.

KEY SKILL/PROCEDURES

- Analyze technology's impact on society.
- Describe the different impacts technology can have on society.
- Explain the changes that computers have brought to other professions.
- Define what skills are included in computer science.
- Investigate what professions are now using computer science skills.

Content Elaborations

CLARIFICATIONS

Examples of the impacts that technology has had on society often address mentality, working different hours, no longer being connected to land lines and different jobs.

Strand	Impacts of Computing
Topic	**Culture**
	Changes that computer have brought to professions would be effects such as fewer machinists and more robotics in manufacturing. **CONTENT FOCUS** Focus on how technology has changed society. Focus especially on the unintentional and unexpected consequences of technological innovation. **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. *Practice 2.* Collaborating Around Computing 4. Evaluate and select technological tools that can be used to collaborate on a project.

Strand	Impacts of Computing
Topic	**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.

Expectations for Learning

LEARNING PROGRESSION

By the end of 8th grade:
Students will have evaluated the pros and cons of face-to-face interactions versus electronic interactions (e.g., emails, phones, social media).

In the 9-12 Foundational Level:
Students will create and use detailed criteria to evaluate the appropriateness of different computer communication tools to solve a given problem. Students will also begin to recognize and investigate the amount of information generated by a computer's automated processes and the resulting privacy concerns.

Beyond the 9-12 Foundational Level:
Students will continue to investigate digital social interactions at the collegiate level.

IMPORTANT CONCEPTS

- Technology has made communication much cheaper and easier.
- Criteria can be created and used to determine the best solution to a problem.
- There are many tools that can be used to increase communication, and each has different costs and benefits.
- Information is valuable.
- Many people do not understand when and how their information is collected and used.
- Giving away private information can make it easier for your identity to be stolen.

KEY SKILL/PROCEDURES

- Identify criteria that would be useful in evaluating computer communication tools.
- Evaluate and use computer communication tools for appropriateness to solve a given problem.
- Compare and contrast different communication tools.
- Understand the value of information, especially personally identifiable information.
- Describe how automated processes can capture data.
- Explain the privacy concerns of automatic data collection.

Strand	Impacts of Computing
Topic	**Social Interactions**

Content Elaborations

CLARIFICATIONS

Examples of computer communication tools include email, social networking sites, IM, RF and Bluetooth.

Personally, identifiable information is any information that can tie a piece of data back to the person who originally owned it. Examples would include SSN, last name, first name, favorite food, birth city and state, ID number and credit card information.

Criteria for evaluating communication tools could include reliability, access to software/hardware, cost and bandwidth requirements.

CONTENT FOCUS

Focus on how technology has affected communication and the difficulty in using communication tools without giving up some privacy information. Do not worry too much about the mechanisms of automated collection. Focus on the impact of automated collection.

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

3. Articulate ideas responsibly by observing intellectual property rights and giving appropriate attribution.

Strand	Impacts of Computing
Topic	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 in classical and emerging technologies.

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.

Expectations for Learning

LEARNING PROGRESSION
By the end of grade 8:

Students understand and can explain the difference between public and personal information, the necessity of not sharing personal information and the implications of misinformation.

In the 9-12 Foundational Level:

Students will investigate 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 will focus 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 copyright law. This will also include creating, solving and implementing solutions to a variety of real-world situations.

IMPORTANT CONCEPTS

- There are legal ramifications when individuals or groups break security or privacy laws.
- There are many practices involving privacy and security that, while technically legal, are not ethical.
- It is difficult to increase security in a system without decreasing usability.
- It is difficult to increase usability in a system without decreasing security.
- Automated collection of data can have legal implications depending on how it is done.
- Users have legal rights that they often are not aware of.
- Intellectual property laws impact how data can be used and what attribution is required.
- There are multiple types of licenses that can be applied to information and processes.
- Intellectual property laws can both protect creators and stifle innovation.
- Software "Terms of Use" and privacy policies can affect your legal rights.
- Existing law and legislation may not be current due to emerging technologies' rapid advancement.
- Challenges identified with classical technologies can guide how we prepare for future technologies.

Strand	Impacts of Computing
Topic	Safety, Law and Ethics

KEY SKILL/PROCEDURES

- Describe what constitutes a breach of security or a breach of privacy.
- Analyze security and privacy breaches.
- Summarize the legal and ethical impact of security breaches.
- Explain the difference between law and ethics.
- Analyze the tradeoffs between security and usability.
- Describe the legal protections people have under current laws.
- Compare how different companies use automatic processes of data collection.
- Define intellectual property.
- Explain the beneficial and harmful effects of intellectual property laws.
- Describe the different types of licenses that exist to protect content creators.
- Explain what a Creative Commons license is and how it is used.

Content Elaborations

CLARIFICATIONS

Analyzing security and privacy breaches could include looking at how they happened and why. Additionally, you could examine what the company could have done to minimize or avoid the breach.

Students should understand software "Terms of Use" and privacy policies.

Your legal rights include the rights to your personal data and your intellectual property rights.

CONTENT FOCUS

Focus on tradeoffs, both security versus usability and free software versus commercial software. You do not need to go deeply into legal understanding. A basic understanding should be sufficient to discuss the impact of the laws.

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.