Ohio's Standards and Model Curriculum

Computer Science Grades 9-12

DECEMBER 2018

Ohio | Department of Education

Computer Science Model Curriculum for Grades 9-12

FOUNDATIONAL LEVEL

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.

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

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 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
- a microwave

Examples of complex devices could be:

- tablet on a refrigerator
- an alarm system
- a multi-functional printer

Examples of virtual devices could be:

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

Examples of connecting technologies could be:

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

CONTENT FOCUS

Understand that everyday devices have computers

Understand how different devices communicate with each other

Understand the difference between virtual and physical

Strand	Computing Systems
Topic	**Devices**
	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
- simple/complex (garage door opener/multi-function printer)

Firmware controls specific hardware (graphics card).

Applications provide a service for the end user.

Strand	Computing Systems
Topic	**Hardware and Software**
	CONTENT FOCUS Focus on understanding what operating systems are and how they work. **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. User tests system
6. Documentation

Strand	Computing Systems
Topic	Troubleshooting
	CONTENT FOCUS Focus on using systematic processes to solve problems. Understand how to analyze computer parts to identify problems. **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 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 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.

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 are able to 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.

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

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

Strand	**Networks and the Internet**
Topic	**Cybersecurity**
	Know common types of cybersecurity attacks to recognize and implement counter measures Physical security is necessary in order 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. **CONTENT FOCUS** Understanding the cybersecurity cycle Response plans (i.e., plan & prepare, identify, containment, investigation, remediation, follow-up) 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, taking into account 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	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

Strand	Data and Analysis
Topic	**Data Collection and Storage**

- Demonstrate understanding of how data is stored on different computer systems

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
- experimental data

Data analysis tools could include:

- filters
- pivot tables
- similarity matching
- regression
- classifications

Ways of representing data could include:

- ASCII
- UTF
- Decimal
- Binary
- Hexadecimal

Ways of encoding data could include:

- XML
- databases
- spreadsheets
- flat files

Strand	Data and Analysis
Topic	**Data Collection and Storage**
	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. **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

Strand	Data and Analysis
Topic	Visualization and Communications

Content Elaborations

CLARIFICATIONS

A multisensory artifact could be:

- data visualizations
- audio analysis
- tactile analysis
- olfactory analysis

A data visualization artifact could be:

- a web page
- a chart
- 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 are 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 in order 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 in order 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 or invalid. • There are many different types of data models that can be utilized. • There is 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 common data model types

Strand	Data and Analysis
Topic	Inference and Modeling

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.

Analyzation is to deconstruct an argument; to examine critically, so as to bring out the 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, 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.

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.	**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

Strand	Algorithmic Thinking and Programming
Topic	Algorithms

- Optimize an algorithm by removing redundant or repeated code and using alternative control structures

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.

Flow of information through a program can be represented visually.

Steps of a computer program can be represented by a diagram.

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. *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	**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 have an understanding of 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

Strand	Algorithmic Thinking and Programming
Topic	Variables and Data Representation

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

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 have an understanding of 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
- 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.

Strand	Algorithmic Thinking and Programming
Topic	**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. **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 in order 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 work flows. 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 specifies 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 work flows
- 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 Agile Development model. The steps in both and 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.

Strand	**Algorithmic Thinking and Programming**
Topic	**Program Development**
	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, taking into account 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	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 Explore other professions to understand how computing has and will impact them positively and negatively.

Expectations for Learning

LEARNING PROGRESSION

By the end of grade 8:
Students will have an understanding of 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

Strand	Impacts of Computing
Topic	Culture

Content Elaborations

CLARIFICATIONS

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

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.

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 are able to 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

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.

Strand	**Impacts of Computing**
Topic	**Safety, Law and Ethics**
	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.

ADVANCED LEVEL

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

Strand	Computing Systems
Topic	Devices

Content Elaborations

CLARIFICATIONS

Examples of simple devices could be –

- a garage door opener
- a toaster
- a microwave

Examples of complex devices could be –

- a tablet on a refrigerator
- alarm system
- 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 are able to 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
- IoTsimple/complex (garage door opener/multi-function printer)

CONTENT FOCUS

Focus on constructing a system for a given purpose.

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, taking into account key features, time and resource constraints, and user expectations. 2. Create a computational artifact for practical intent, personal expression, or to address a societal issue. 3. Modify an existing artifact to improve or customize it. *Practice 6*. Testing and Refining Computational Artifacts 6.3. Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability, and accessibility.

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. User tests system. 6. 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.

Strand	**Networks and the Internet**
Topic	**Networking**
	CONTENT FOCUS IPV4 and IPV6 addressing Variable length subnetting and CIDR notation Basic networking protocols **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 are able to understand the cybersecurity impact on business and 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)

Strand	Networks and the Internet
Topic	Cybersecurity
	CONTENT FOCUS Understanding virus protection Focus on ethics and laws regarding 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.

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

Strand	Data and Analysis
Topic	Data Collection and Storage

Content Elaborations

CLARIFICATIONS

When collecting data, be sure to address 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.

Data analysis tools could include:

- Spreadsheets
- Charts
- Graphs
- 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:
Students will, given a real-world relatable problem, 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 pattern 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

Strand	Data and Analysis
Topic	Visualization and Communications

Content Elaborations

CLARIFICATIONS

Complex data analysis could include:

- relational database queries
- data regression
- spreadsheets
- inferences
- predictions

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

Denormalize to improve performance

Strand	Data and Analysis
Topic	Inference and Modeling
	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. 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 recursion to effectively solve problems.

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

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

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.

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 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

Content Elaborations

CLARIFICATIONS

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

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

Strand	Algorithmic Thinking and Programming
Topic	**Algorithms**
	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.

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 of the code necessary to use it in a program.

An example of implementations 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 pass by value or pass by reference. In many languages, this may be automatically determined based on data type.

CONTENT FOCUS

Strings are basically 1D arrays. A typical data table is an example of a 2D array.

A list is similar to 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

Strand	Algorithmic Thinking and Programming
Topic	Control Structures

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.

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)
- 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 work flows. 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
- 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, taking into account 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	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 equity, access and influence of the distribution of computing resources to see their global impact.

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

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

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 wifi.

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 Investigate intellectual property laws, including copyright, trademarks and patents, to identify some of the 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

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.