Computer Science Standards of Learning for Virginia Public Schools

June 2024

Board of Education
Commonwealth of Virginia

Computer Science Standards of Learning for Virginia Public Schools

Superintendent of Public Instruction
Lisa Coons, Ed.D.

Deputy Superintendent of Teaching and Learning
Em Cooper

Assistant Superintendent of Instruction
Michelle Wallace, Ed.D.

Office of Educational Technology and Classroom Innovation
Calypso Gilstrap, Associate Director
Keisha Tennessee, Computer Science Coordinator

Table of Contents

Computer Science K-12 *Standards of Learning* Guiding Principles

Introduction

Virginia's Computer Science standards aim to raise our aspirations for computational instruction to enable students to engage and thrive in a digital world. Beginning in the earliest grades and continuing through 12th grade, students must develop a foundation of computer science knowledge and learn new approaches to problem solving that harness the power of computational thinking to become both users and creators of computing technology.

Computer Science is one of Virginia's largest and fastest growing industries, and computer programmers are needed within every field, including healthcare, transportation, and banking. Irrespective of a Virginian's post-secondary occupation, all students will benefit from learning computer science concepts and practices that support better understanding of the technologies underpinning the world around them, improve their logical reasoning and problem-solving skills, and increase their creativity and collaboration across all educational disciplines.

As students engage with sophisticated technologies in a quickly evolving landscape, Virginia's Computer Science standards couple age-appropriate digital instruction with digital stewardship practices that inform safe interactions with people and content online, the protection of private digital information, screen-time management, and awareness of socio-emotional health issues that can arise in digital settings.

The standards are organized into the following content strands: Algorithms and Programming, Computing Systems, Cybersecurity, Data and Analysis, Impacts of Computing, and Networks and the Internet. The Standards of Learning, within each strand, progress in complexity throughout the grade levels and into high school course content.

In elementary grades, students will be introduced to computer systems, develop their understanding of how multi-step solutions are executed within computer programs, and develop awareness of safe and responsible device practices. In middle school, the instructional focus extends those concepts into activities that introduce computational thinking and information-processing skills. In both the middle school and high school, computer science courses are designed to develop a deeper understanding and application of conceptual knowledge, algorithm reasoning and design, and preparation for post-secondary opportunities.

Foundational Principles

Computer Literacy is foundational to learning and post-secondary success as technology becomes increasingly incorporated into all aspects of everyday life. Computer Literacy provides critical knowledge and skills for all subject areas including mathematics, science, history, English, and fine arts. By applying computer science as a tool for learning and expression in a variety of

disciplines and interests, students will actively and proficiently participate in a world that is increasingly influenced by digital technology.

Computer Science fosters problem solving skills that are essential to all educational disciplines and post-secondary employment opportunities. Understanding how multi-step solutions are executed within computer programs allows students the opportunity to use metacognitive strategies with tasks they are performing as they work and study in any topic area. Computer Science should become an essential part of Virginia K-12 education, accessible by all, rather than a vocational part of education only for those headed to technology-based employment.

Computer Science instruction must maintain the pace of technology evolution to prepare students for the workforce. Computer science is a core technology component for students to have the ability to adapt to the future evolution of work. The workforce of the future will increasingly require that all adults effectively work in digital environments and utilize technology both ethically and responsibly. As a result, we must prioritize preparing all students with integral computer science learning opportunities throughout their academic career to ensure they are prepared for a post-secondary success in a digital world that includes computer-based problem solving, artificial intelligence and communication rooted in the use of digital tools.

Students should gain specific digital and computational concepts to harness the power of computer science and derivative applications, such as machine learning, online programming, virtual reality, and Artificial Intelligence (AI), to embrace innovation and chart the future of individuals, business, and government responsibly.

Other foundational principles for these Computer Science learning standards include:

- **Computational Thinking** leverages logical reasoning and creativity to develop problem definition, component deconstruction and reconstruction, pattern identification, and systems interactions skills that develop resilient learners.
- **Responsible Digital Collaboration** can expedite identification of solutions in a team setting, allowing diverse perspectives, and conflicting ideas to be incorporated and reconciled more efficiently.
- **Authentic Computing** is paramount. Students must develop original work products, rather than relying on artificial intelligence (AI) , to produce meaningful and authentic mastery of concepts across educational disciplines.
- **Artificial Intelligence (AI) Must be Used Ethically.** As the availability of artificial intelligence continues to increase, it is important to instruct students on effective and appropriate AI use.

Kindergarten: Computer Science

In Kindergarten, students explore the basics of computing and technology, gaining an understanding of fundamental components. They learn how information is transmitted electronically through devices and the Internet. Both independently and collaboratively, students develop the ability to identify and troubleshoot issues with computing devices, emphasize responsible behaviors, and know the importance of safeguarding online information. The standards also introduce computational thinking as a foundational problem-solving approach in computer science, encouraging students to recognize patterns, categorize items, create visual representations, and design algorithms. The iterative design process is used to plan, implement, and test algorithms in a sequential order. Additionally, students examine the societal impact of computing technologies in various settings, including homes, schools, and future careers.

Algorithms and Programming (AP)

K.AP.1 **The student will apply computational thinking to identify patterns and sort items into categories based on an attribute.**

a. Identify attributes of a set of objects.
b. Compare two objects and list attributes they have in common.
c. Sort and classify concrete objects based on one attribute.

K.AP.2 **The student will plan and implement algorithms that include sequential order.**

a. Identify tasks that are completed using sequential step-by-step instructions.
b. Recall and state thoughts, ideas, and stories in the form of sequential steps.
c. Create a design document to illustrate thoughts, ideas, and stories in a sequential manner.
d. Create and test a sequential algorithm emphasizing beginning, middle, and end.
e. Create and test a sequential algorithm to accomplish a predetermined task.

Computing Systems (CSY)

K.CSY.1 **The student will identify the purpose and components of a computing device.**

a. Identify and locate parts of a computing device, including sensors.
b. Describe the function of common components in a computing device.
c. Identify different types of computing devices with screens.
d. Identify tasks and activities at home and at school that use screens.
e. Demonstrate proper use of computing devices.

K.CSY.2 **The student will identify when a computing device might not work as expected.**

a. Identify a problem with a computing device when it is not working as expected.
b. Explain what to do when a computing device is not working as expected.

Cybersecurity (CYB)

K.CYB.1 The student will demonstrate safe and responsible use of computing technologies.

a. Identify safe and responsible uses of computing technologies based on the school rules and acceptable use policy (AUP).
b. Demonstrate safe and responsible uses of computing technologies.
c. Communicate the process(es) for reporting behaviors that do not comply with school rules or acceptable use policy.

K.CYB.2 The student will describe the importance of protecting personal information online.

a. Describe what personal information should be shared and not shared.
b. Determine to whom personal information should be shared.
c. Identify safe practices to keep personal information private.

Data and Analysis (DA)

K.DA.1 The student will gather and record data with or without a computing device.

a. Discuss the importance of data.
b. Identify numeric and non-numeric data.
c. Record data and communicate possible patterns.

K.DA.2 The student will create representations of data to make predictions and draw conclusions.

a. Create tables, object graphs, picture graphs, and/or models.
b. Describe the information from a given data visualization.
c. Use data to answer questions, make predictions, and draw conclusions.

Impacts of Computing (IC)

K.IC.1 The student will identify and discuss how computing technologies impact people's lives.

a. List computing technologies found in the classroom, home, and the community.
b. Identify how computing technologies are used in daily tasks.
c. Discuss healthy habits for using computing technologies.

K.IC.2 The student will discuss and describe how different careers use computing technologies.

a. Describe how computing technologies are used in various careers.
b. Discuss the advantages of using computing technologies in different careers.

c. Identify local jobs that utilize computer technologies.

Networks and the Internet (NI)

K.NI.1 The student will describe how people can communicate with others by connected computing devices and the Internet.

a. Identify ways to communicate with others using connected computing devices and the Internet.
b. Describe the benefits of communicating with others using connected computing devices and the Internet.

First Grade: Computer Science

In First Grade, students learn how computing components work together to form systems for collecting and sharing data/information. Students continue to develop problem-solving skills that apply computational thinking and can be applied with addressing hardware and software issues. Students refine computational thinking practices through activities such as planning, document creation, and construction of programs that consist of events and sequences. Students will further their knowledge on the importance of simple safeguards to protect private information and apply responsible behaviors and proper use of computing devices.

Algorithms and Programming (AP)

1.AP.1 The student will apply computational thinking by sorting items into categories based on multiple attributes and create patterns.

a. Describe attributes of a set of objects.
b. List the attributes a set of objects have in common.
c. Sort and classify concrete objects based on multiple attributes.
d. Create repeating and increasing patterns.

1.AP.2 The student will plan and implement algorithms that include the use of sequence and an event based on a predetermined task.

a. Plan and create a design document that illustrates thoughts, ideas, and stories in a sequential manner.
b. Construct step-by-step instructions that include decision-making and repetition.
c. Identify and explain the role of events that are used in an algorithm.
d. Test algorithms that are sequential and contain an event.

1.AP.3 The student will use the iterative design process to construct, test, and debug algorithms that include sequencing and an event.

a. Discuss and describe the concept of debugging.
b. Analyze and explain the results of an algorithm.
c. Revise and improve an algorithm to produce desired outcomes.

Computing Systems (CSY)

1.CSY.1 The student will describe how computing components work together to create a computing system.

a. Identify and define hardware, software, and computing systems.
b. Identify common components of computing systems in different types of computing devices.
c. Describe how hardware and software work together to form a computing system.

1.CSY.2 The student will use accurate terminology to describe when a computing system might not work as expected.

a. Identify and describe a problem with a device or computing system when it does not work as expected.
b. Propose a solution to simple hardware or software issues.

Cybersecurity (CYB)

1.CYB.1 The student will demonstrate safe and responsible use of computing technologies.

a. Describe safe and responsible uses of computing technologies based on the school rules and acceptable use policy (AUP).
b. Demonstrate safe and responsible behaviors when using computing technologies and online communication.
c. Discuss the process for reporting inappropriate technology use at school or home.
d. Classify appropriate and inappropriate uses of technology at school or at home.
e. Explain the consequences of inappropriate uses of computing technologies.

1.CYB.2 The student will discuss the importance of using a password to protect private information.

a. Describe the purpose of usernames and passwords.
b. Discuss how passwords are private information and are used to protect the privacy of information.

Data and Analysis (DA)

1.DA.1 The student will explore how data can be stored and retrieved from various computing devices.

a. Identify data formats used for various purposes, including audio, images, text, and video.
b. Explore and identify computing devices that collect, store, and/or display data.

1.DA.2 The student will create representations of data to make predictions and draw conclusions.

a. Collect and organize data with or without a computing device.
b. Create tables, object graphs, picture graphs, and models using abstraction.
c. Identify patterns and describe trends in data visualizations of various formats.
d. Use data to answer questions, draw conclusions, and make predictions.

Impacts of Computing (IC)

1.IC.1 The student will describe how computing technologies impact daily tasks and communication.

a. Determine when tasks should be completed with or without computing devices.
b. Describe how computing devices are used in communication.
c. Describe healthy habits for using computing technologies.

1.IC.2 The student will describe tasks and activities that use screens and categorize them based on their use.

a. Identify daily routines and activities that can be completed with or without screens.
b. Classify the different uses of screen time as learning, entertainment, or communication.

1.IC.3 The student will compare and contrast ways people complete tasks with and without computing technologies.

a. Identify tasks that can be completed with and without computing technologies.
b. Discuss advantages and disadvantages of using and not using computing technologies.
c. Describe how the appropriate use of computing technologies can improve efficiency.
d. List computing technologies used in various careers.

Networks and the Internet (NI)

1.NI.1 The student will explain that computing devices and the use of the Internet allow people the ability to gather information and connect with others.

a. Describe how the Internet can be used to gather information.
b. Explain ways people communicate using computing devices and the Internet.

Second Grade: Computer Science

In Second Grade, students apply computational thinking skills as they identify patterns and create algorithms for comparing objects based on attributes. Students follow the iterative design process, planning, implementing, and evaluating algorithms with events and loops using a block-based programming language. Proficiency is demonstrated in building, testing, and debugging programs. The use of computational thinking is expanded beyond programing and applied through abstraction and manipulation of data to create models and representations. Students define and categorize input and output components, explaining how computing systems acquire input and produce output. Students utilize the Internet to gather information and for collaborative tasks, while demonstrating safe behaviors when interacting with others and using the information. While using online resources, students learn the importance of password usage for privacy.

Algorithms and Programming (AP)

2.AP.1 **The student will apply computational thinking to identify patterns, and design algorithms to compare and contrast objects based on attributes.**

a. Compare and contrast multiple ways to sort a set of objects.
b. Create a table of features to organize objects.
c. Design an algorithm to sort objects into categories based on multiple attributes.

2.AP.2 **The student will plan and implement algorithms that consists of events and loops using a block-based programming language.**

a. Plan and create a design document to guide the construction of a program using plain language or pseudocode.
b. Identify a section of repeated actions within an algorithm and replace it with a loop.
c. Construct step-by-step instructions that include events and repetition.

2.AP.3 **The student will use the iterative design process to create, test, and debug a program containing events and loops in a block-based programming tool.**

a. Define program.
b. Read and interpret a program expressed in a block-based programming language.
c. Analyze and describe the results of a program.
d. Create and test a program that uses events and loops.
e. Revise and improve a program to produce desired outcomes.

Computing Systems (CSY)

2.CSY.1 The student will describe the characteristics of computing systems including hardware, software, input, and output.

a. Describe how hardware and software work together to accomplish a task.
b. Define and categorize components as inputs and outputs.
c. Describe how a computing system receives input and provides output.
d. Discuss how computers use binary to communicate and process information.

2.CSY.2 The student will demonstrate an understanding of how to troubleshoot simple hardware and software problems that may occur during use.

a. Propose solutions to simple hardware and software issues.
b. Use appropriate steps to perform simple troubleshooting tasks.

Cybersecurity (CYB)

2.CYB.1 The student will model safe and responsible behaviors when using information and computing technologies.

a. Explain the need for safe and responsible uses of computing technologies.
b. Create a flowchart to illustrate the process for reporting inappropriate use of technology at school or at home.
c. Demonstrate and model safe and responsible behaviors when using computing technologies and online communication.

2.CYB.2 The student will explain the importance of using passwords to protect private information.

a. Identify and classify passwords as strong or weak.
b. Explain how a strong password helps protect the privacy of information.
c. Explain the risk of sharing passwords.

Data and Analysis (DA)

2.DA.1 The student will analyze data to make decisions with or without a computing device.

a. Collect and record numeric and non-numeric data and describe possible patterns.
b. Create questions that can and cannot be answered by the data.
c. Analyze data to draw conclusions and make decisions.

2.DA.2 The student will manipulate data, create representations, and evaluate data to solve a problem.

a. Create charts, graphs, and models using abstraction to represent data.
b. Analyze data visualizations to draw conclusions.
c. Propose and evaluate a solution to a problem or question based on data and/or data visualization.

Impacts of Computing (IC)

2.IC.1 The student will examine the positive and negative impacts of how using computing technologies has changed the way people live, work, and interact.

a. Identify current uses of computing/emerging technologies and discuss how they impact society.
b. Compare and contrast appropriate and inappropriate online behaviors that apply in the physical environment and the online environment.
c. Model healthy habits for using computing technologies.

2.IC.2 The student will explain the need to balance screen time and other activities.

a. Discuss appropriate times and places for screen use.
b. List and describe alternatives to screen time.

2.IC.3 The student will explain how computing technologies have an impact on the workforce.

a. Explain how computing technology is used in various careers.
b. Identify skills needed for careers that use computing technologies.
c. Discuss how computing technologies have changed the workplace.

Networks and the Internet (NI)

2.NI.1 The student will demonstrate the use of the Internet in gathering information to accomplish a task.

a. Explore ways information is organized and shared on the Internet.
b. Gather information from the Internet.
c. Summarize collected information using own words.

Third Grade: Computer Science

In Third Grade, students delve into computational thinking, creating algorithms to tackle problems with a block-based programming language and the iterative design process. They develop skills in strategizing, constructing, executing, and testing algorithms incorporating loops, events, and conditional control structures. Students will create and evaluate data representations, using computing devices to model physical objects or processes, and assessing the societal impacts of widespread computing technology use. Students gain insights into inner workings of computing devices within a system, employing accurate terminology to address issues with malfunctioning systems. They adopt secure practices for protecting private information, understanding the correlation between passwords and security risks. Additionally, students apply the data cycle, acquiring skills to collect, store, and organize data for trend evaluation and pattern identification. Students will familiarize themselves with computing technology careers, understanding societal implications, and use of information created by others with proper permissions.

Algorithms and Programming (AP)

3.AP.1 **The student will apply computational thinking to design algorithms to extend patterns, processes, or components of a problem.**

a. Identify a pattern in an algorithm, process, or a problem.
b. Decompose a problem or task into a subset of smaller problems.
c. Design an algorithm to extend either a pattern, process, or component of a problem.

3.AP.2 **The student will plan and implement algorithms that consist of events and conditional control structures using a block-based programming language.**

a. Describe the concept of a conditional control structure.
b. Create a design document to plan an algorithm using plain language, pseudocode, or diagrams.

3.AP.3 **The student will use the iterative design process to create, test, and debug programs containing events, loops, and conditional structures in a block-based programming tool.**

a. Create and test programs that consist of events, loops, and conditional structures.
b. Analyze and describe program results to assess validity of outcome.
c. Revise and improve programs to resolve errors or produce desired outcome.

Computing Systems (CSY)

3.CSY.1 The student will model how computing devices within a computing system work.

a. Describe the role of a processor in a computing system.
b. Explain the relationship between the inputs, processors, and outputs.
c. Discuss various types of input data a computer can accept and use.
d. Model how a computing system works including input and output, processors, and sensors.

3.CSY.2 The student will use accurate terminology when troubleshooting problems when a computing system is not working as expected.

a. Identify common troubleshooting strategies used to address a variety of hardware and software problems.
b. Explain and apply troubleshooting strategies related to simple hardware and software problems.

Cybersecurity (CYB)

3.CYB.1 The student will apply safe practices to protect private information.

a. Identify and distinguish personal information that should be private.
b. Describe the importance of using a strong password.
c. Create and use strong passwords to protect private information.

3.CYB.2 The student will identify the relationship between passwords and security risk.

a. Describe how authentication and authorization protect private information.
b. Identify multiple authentication methods.
c. Discuss the security risk posed by not having a strong password.

3.CYB.3 The student will define and explain cybersecurity.

a. Define cybersecurity.
b. Research and identify problems and consequences related to inappropriate use of computing devices and networks.
c. Model safe and responsible behaviors when using computing technologies and online communication.

Data and Analysis (DA)

3.DA.1 The student will gather, store, and organize data to evaluate trends and identify patterns using a computing device.

a. Formulate questions that require the collection or acquisition of data.
b. Gather, organize, sort, and store data.
c. Examine a labeled dataset to identify potential problems within the data.
d. Discuss how data discrepancies or problems impact predictions and results.
e. Draw conclusions and make predictions based on observed data.

3.DA.2 The student will create and evaluate data representations and conclusions.

a. Create charts and graphs based on data collection.
b. Analyze data to identify patterns, draw conclusions, and make predictions.

3.DA.3 The student will create models that can represent a physical object or process.

a. Create a model to represent a physical object or process.
b. Identify how computing devices are used to create models.
c. Discuss the advantages and disadvantages of using computing devices to create models.

Impacts of Computing (IC)

3.IC.1 The student will identify and examine the positive and negative impacts of the prevalence of computing technologies.

a. Identify computing technologies that have changed the world.
b. Examine and explain how computing technologies influence and are influenced by culture.
c. Identify social and ethical issues related to the use of computing technologies.

3.IC.2 The student will discuss and describe strategies to manage screen time.

a. Define and describe screen time.
b. Explain the importance of responsible screen time management.
c. Discuss how screen time choices affect one's personal health and interactions with others.

3.IC.3 The student will identify and describe computing technology careers and their impact on society.

a. Research computing technology careers.
b. Describe the impact careers in computing technology have on society.

3.IC.4 The student will demonstrate how to use information created by others with permission.

a. Discuss copyright, piracy, and plagiarism.
b. Demonstrate how to use information created by others.

Networks and the Internet (NI)

3.NI.1 The student will describe computing networks.

a. Differentiate between a network and the Internet.
b. Identify the components of a computing network.
c. Describe how a computing device connects to a network.
d. Identify ways networks are used to transmit information.

Fourth Grade: Computer Science

In Fourth Grade, students deepen their comprehension of the intricate connection between computing devices and networks. The exploration of cybersecurity becomes a focal point, guiding students in understanding the responsible use of computing technologies and the repercussions of inappropriate usage. The students investigate the mechanisms through which computers perceive the world, incorporating sensors and various inputs into their understanding. Computational practices progress as algorithms includes variables and user input. The utilization of data is broadened, with a focus on identifying appropriate data types for problem-solving and creating models of real-world situations.

Algorithms and Programming (AP)

4.AP.1 **The student will apply computational thinking to identify patterns and design algorithms to compare and contrast multiple algorithms used for the same task.**

a. Decompose an algorithm, process, or problem into a subset of smaller problems.
b. Identify multiple algorithms for the same task.
c. Describe patterns within multiple algorithms.
d. Determine which algorithm is most effective for a given task.

4.AP.2 **The student will plan and implement algorithms that consist of sequencing, loops, variables, user input, and conditional control structures using a block-based programming language.**

a. Identify user input and its role in improving a program.
b. Describe the concept of a variable.
c. Read and explain a design document to trace and predict an algorithm using plain language, pseudocode, or diagrams.
d. Create a design document to plan an algorithm using plain language, pseudocode, or diagrams.
e. Write programs that initialize, assign values to, name, and modify variables.

4.AP.3 **The student will use the iterative design process to create, test, and debug programs containing sequencing, loops, variables, user inputs, and conditional control structures in a block-based programming tool.**

a. Create and test programs that consist of sequencing, loops, variables, user inputs, and conditional control structures.
b. Create and use variables to store and process data.
c. Trace and predict the value of variables that change over the course of the program's runtime.
d. Analyze and describe program results to assess validity of outcomes.
e. Revise and improve programs to resolve errors or produce desired outcomes.

Computing Systems (CSY)

4.CSY.1 The student will model how a computing system works to accomplish a task.

a. Describe how computing systems perceive the world through sensors and other inputs.
b. Compare and contrast how humans and computers process information from inputs.
c. Explain how computing devices may be used to classify and organize input.
d. Diagram and describe a simple computing system indicating processors, inputs, and outputs.

4.CSY.2 The student will apply troubleshooting strategies when a computing system is not working as intended.

a. Identify hardware, software, and connectivity problems using accurate terminology.
b. Apply troubleshooting strategies to address hardware, software, and connectivity problems.

4.CSY.3 The student will describe the learning process of humans and computers.

a. Compare and contrast how humans and computing technologies collect, store, and process data.
b. Identify similarities and differences on how humans and computing technologies infer and extract meaning from data.
c. Define machine learning and identify machine learning approaches: supervised, unsupervised, and reinforcement learning.

Cybersecurity (CYB)

4.CYB.1 The student will examine the impacts of appropriate and inappropriate use of computing technologies.

a. Examine and explain scenarios for appropriate and inappropriate use of computing technologies.
b. Develop possible solutions involving inappropriate use of computing technologies.

4.CYB.2 The student will identify and investigate best practices to safeguard information shared online and through online platforms.

a. Classify personal, private, and public information.
b. Research and evaluate tradeoffs of sharing information.
c. Investigate and communicate best practices to limit unauthorized access to information on a computing device.
d. Demonstrate proper use and protection of personal passwords.

e. List methods used to safeguard online information.

4.CYB.3 The student will examine how information is shared online and explain the importance of cybersecurity.

a. Investigate multiple ways people share information online.
b. Determine and describe when information should be shared and to whom it should be shared.
c. Describe how personal information can be collected and shared online.
d. Explain the importance of cybersecurity.

Data and Analysis (DA)

4.DA.1 The student will identify the appropriate type of data needed to solve a problem or answer a question.

a. Analyze a problem to determine the appropriate type of data needed.
b. Evaluate the reliability of data sources.
c. Use numeric values to represent non-numeric ideas to include binary, American Standard Code for Information Interchange (ASCII), and RGB values.
d. Collect, store, clean, and organize data for analysis and to prepare visualizations.

4.DA.2 The student will create and evaluate data representations to make predictions and conclusions.

a. Formulate questions that require the collection or acquisition of data.
b. Collect data to create charts and graphs.
c. Recognize and analyze patterns and relationships within data sets.
d. Analyze visual representations to make predictions and draw conclusions.

4.DA.3 The student will create a computational model that represents attributes and behaviors associated with a concept.

a. Examine models of physical objects and processes.
b. Create a computational model that reflects the attributes and behaviors associated with a concept.
c. Explain how a computer model illustrates a given concept.

Impacts of Computing (IC)

4.IC.1 The student will identify and examine the positive and negative impacts of the prevalence of computing technologies.

a. Identify computing technologies that have changed Virginia's economy.

b. Examine and explain how computing technologies influence and are influenced by culture.
c. Identify social and ethical issues related to computing devices and networks.

4.IC.2 The student will describe the impact of screen time on relationships at home and at school.

a. Describe the impact of excessive screen time on maintaining friendships and family dynamics.
b. Explain how playing video games and the use of social media can impact relationships and personal health.

4.IC.3 The student will examine the impact of computing technologies in the workforce.

a. Research and analyze the skills needed for careers in computing technology fields.
b. Examine the impacts of diversity and inclusivity in computing technology fields.

4.IC.4 The student will describe the importance of copyrights and intellectual property rights.

a. Demonstrate an understanding of copyright and the fair use of information.
b. Explain how intellectual property can be protected.
c. Give proper attribution to the original author of digital and online content.

Networks and the Internet (NI)

4.NI.1 The student will identify the interrelationship between computing devices and a computing network.

a. Define client and server.
b. Describe how packets are used to transmit information on a network.
c. Describe factors that may affect the speed of data transmission.
d. Differentiate between networking tasks that require Internet access and tasks that do not require Internet access.
e. Model how computing devices in a network transmit and receive information.

Fifth Grade: Computer Science

In Fifth Grade, students explore cloud computing, examining security risks and adopting best practices to protect information. They research cybersecurity policies and laws, researching ways to prevent unauthorized access to data. Algorithms undergo expansion with the inclusion of multi-way branching and nested conditional control structures. Students develop programs incorporating rational and arithmetic expressions for performing calculations. Evaluation of data source reliability becomes a focus, enabling students to draw conclusions from data visualizations. Additionally, the impact of social interactions through computing technologies and the exploration of open-source licenses are addressed, along with considerations on expanding the capabilities of computing devices.

Algorithms and Programming (AP)

5.AP.1 **The student will apply computational thinking to identify patterns, make use of decomposition to break down problems or processes into sub-components, and design algorithms.**

a. Identify patterns and repeated steps in an algorithm, problem, or process.
b. Decompose a problem or process into a subset of smaller problems or groups of sequential instructions.
c. Abstract relevant information to identify essential details.
d. Design an algorithm to solve a problem.

5.AP.2 **The student will plan and implement algorithms that consist of sequencing, loops, variables, user input, and nested conditional control structures using a block-based programming language.**

a. Describe the concept of nested conditional control structure.
b. Create a design document to trace and predict an algorithm using plain language, pseudocode, or diagrams.
c. Read, write, and interpret nested conditional control structures: “if-else” and “if-else; if-else" statements.

5.AP.3 **The student will use the iterative design process to create, test, and debug programs containing sequencing, loops, variables, user inputs, nested conditional control structures, and two-way branching conditional control structures in a block-based programming tool.**

a. Use accurate terminology to describe and explain the iterative design process.
b. Create and test programs that consist of sequencing, loops, variables, user inputs, nested conditional control structures, and two-way branching conditional control structures.
c. Trace and predict outcomes of programs.
d. Analyze and describe program results to assess validity of outcomes.
e. Revise and improve programs to resolve errors or produce desired outcomes.

Computing Systems (CSY)

5.CSY.1 The student will explain how computing systems are used to collect and exchange data.

a. Identify and explain how computing systems store data representations, including images and sound.
b. Describe the role of processing speed and storage capacity when collecting and exchanging data.

5.CSY.2 The student will describe an automated decision-making process employed by a computing system.

a. Explore decision automation and how it is used.
b. List outcomes of a process based on automated decisions.

5.CSY.3 The student will evaluate and implement troubleshooting strategies when a computing system is not operational.

a. Identify and use troubleshooting protocols to resolve hardware, software, and connectivity issues.
b. Apply prior troubleshooting practices to new problems as they arise.

Cybersecurity (CYB)

5.CYB.1 The student will identify ways to limit unauthorized access on computing devices.

a. Define virus, malware, and phishing.
b. Explain how viruses and malware can put personal information at risk.
c. Describe the role of human interactions in social engineering attacks.
d. Identify ways to protect personal and private information when using a computing device and the Internet.
e. Explain the importance of updating software.

5.CYB.2 The student will explain how cybersecurity policies and laws are designed to protect individuals.

a. Explain the importance of policies and laws related to online use of computing devices and the Internet.
b. Research and discuss current cybersecurity policies and laws that protect individuals.
c. Explain legal consequences for inappropriate use of computing technologies.

Data and Analysis (DA)

5.DA.1 The student will collect data or use data sets to solve a problem or investigate a topic.

a. Identify accurate ways data can be collected.
b. Evaluate the reliability of the data source.
c. Organize data based on similarities or patterns.
d. Compare and contrast various data elements.

5.DA.2 The student will create multiple data representations to make predictions and conclusions.

a. Formulate questions that require the collection or acquisition of data.
b. Collect data to use in creating charts, graphs, and models.
c. Analyze data as evidence to draw conclusions and make predictions.
d. Propose solutions to problems or questions based on data analysis.

5.DA.3 The student will explain the significance of training data in machine learning.

a. Compare how training data is utilized in supervised, unsupervised and reinforcement learning.
b. Explain how training data is used to make classification predictions.
c. Discuss the need and significance of diverse, inclusive, and large datasets.

Impacts of Computing (IC)

5.IC.1 The student will analyze the impact of inappropriate use of computing technologies.

a. Predict consequences for inappropriate uses of computing technologies.
b. Describe how technology-related problems can be avoided or prevented.
c. Develop solutions for a scenario involving inappropriate use of computing technologies.

5.IC.2 The student will explain the potential impact of excessive screen time on academic performance.

a. Analyze data to determine the impact of screen time on academic performance.
b. Describe how academic behaviors that lead to academic success are impacted by daily screen time.
c. Differentiate usage of screen time that benefit and hinder academic performance.

5.IC.3 The student will identify the impact of computing technologies on the workforce, culture, and global society.

a. Research and analyze computing technology careers in global society.
b. Examine the impacts of diversity and inclusivity in computing technology fields globally.
c. Explore the impact of emerging technologies on workforce, culture, and global society.

5.IC.4 The student will observe and examine intellectual property rights when considering the use of open-source licenses and copyrights.

a. Distinguish between open-source licenses and copyrights.
b. Research risks associated with inappropriate use of various digital information sources.
c. Describe and use strategies to protect online digital content and resources.

5.IC.5 The student will examine the effects of social interactions due to computing technologies.

a. List and explain how advances in computing technologies impact communication and collaboration.
b. Describe how computing technologies can be designed to engage and interact with users including those with diverse needs.
c. Evaluate activities conducted in the physical and online environments.
d. Create an artifact that illustrates a solution to address the need or want of a user.

Networks and the Internet (NI)

5.NI.1 The student will identify and describe cloud computing.

a. Define cloud computing.
b. List examples of cloud computing.
c. List the advantages and disadvantages of cloud computing.
d. Identify safe practices and potential security risks when using cloud computing.

Sixth Grade: Computer Science

In the Sixth Grade, students build upon their understanding of computing systems and the network and gather and share information effectively. Students gain a more defined understanding of how data is transmitted over the Internet. Students explore real-world applications of cybersecurity, through exploring laws governing privacy and understanding user agreements. Students continue to use computational thinking and an iterative design process to create algorithms and programs. A greater emphasis on the application of data and analysis is present with an understanding of the importance of accuracy and reliability of data, and the use of appropriate computational tools to create data visualizations that are used to make predictions and draw conclusions. Students actively engage in troubleshooting and documenting hardware and software issues to foster resilience, analytical reasoning, and effective communication. Students explore potential career paths and evaluate how computer science skills can be applied in various professions.

Algorithms and Programming (AP)

6.AP.1 **The student will apply computational thinking to identify patterns, make use of decomposition to break down problems or processes into sub-components, and design algorithms.**

a. Identify patterns and repeated steps in an algorithm, problem, or process.
b. Decompose an algorithm, problem, or process into sub-components.
c. Abstract relevant information to identify essential details.
d. Design algorithms using abstraction to accomplish a task or express a computational process.

6.AP.2 **The student will plan and implement algorithms that include conditional control structures and collection of numeric data using a block-based or text-based tool.**

a. Create a decision tree diagram to illustrate the decision-making process.
b. Read and write programs that initialize Boolean, integer, and decimal number variables.
c. Read and write programs that collect numeric data from users.
d. Read and write programs that contain nested conditional control structures.
e. Predict the results of logic expressions that use Boolean operators: and, or, and not; including expressions that use relational expressions as one or more operands.

6.AP.3 **The student will use the iterative design process to create, test, and debug programs using a block-based or text-based programming language.**

a. Create and test programs that uses multiple conditional control structures.
b. Incorporate existing code, media, or libraries into original programs.
c. Trace and predict outcomes of programs.

d. Analyze and describe program results to assess validity of outcomes.
e. Analyze the outcomes of programs to identify logic and syntax errors.
f. Incorporate feedback from others to refine program.
g. Revise and improve programs to resolve errors and produce desired outcomes.

6.AP.4 The student will demonstrate proper attribution when incorporating ideas and works of others.

a. Identify and give proper attribution of information and assets from the Internet and other sources.

Computing Systems (CSY)

6.CSY.1 The student will define and explain application software and operating systems of a computing device within a computing system.

a. Define and describe the functions of an operating system and application software.
b. List advantages and limitations of application software and operating systems based on the needs of the user.

6.CSY.2 The student will identify and explain hardware, software, and connectivity problems and troubleshooting solutions.

a. Identify and explain hardware, software, and connectivity problems and solutions with accurate terminology.
b. Identify resources for troubleshooting hardware, software, and connectivity-related problems.

6.CSY.3 The student will identify and describe Artificial Intelligence (AI).

a. Define artificial intelligence and identify the characteristics of artificial intelligence.
b. Describe how AI technologies generate information or automate decision and how people interact with AI technologies.
c. Define algorithmic bias and explain its consequences on AI technologies and systems.

Cybersecurity (CYB)

6.CYB.1 The student will evaluate the risks and benefits associated with sharing personal and public resources or artifacts.

a. Identify and explain the difference between personal and public information.

b. Discuss the consequences of sharing personal and confidential information online.
c. Evaluate risks and benefits associated with sharing information online.

6.CYB.2 The student will investigate various usage agreements designed to protect individuals.

a. Identify laws governing privacy with computing devices and emerging technologies.
b. Investigate and describe common components of usage agreements.
c. Identify user and company protections in a usage agreement.

Data and Analysis (DA)

6.DA.1 The student will utilize computational tools to collect and organize data.

a. Select and use appropriate computational tools to collect data.
b. Organize data to make it easier to understand and use.
c. Clean data to remove and correct errors.
d. Analyze data sources for accuracy and reliability.

6.DA.2 The student will utilize computational tools to visualize and evaluate data.

a. Identify different types of visual representations of data.
b. Compare various visual representations and identify when each should be used.
c. Create charts, graphs, models, and simulations to visualize data.
d. Describe and synthesize information from a visual representation of data.

6.DA.3 The student will make predictions and draw conclusions from data visualizations.

a. Visualize data using appropriate graphs, charts, and data visualization techniques to enhance understanding and communicate findings effectively.
b. Use computational tools to analyze patterns within data sets and identify trends.
c. Draw conclusions and make predictions based on the analysis and interpretation of the data visualization.
d. Utilize simulations and models to formulate, refine, and test hypotheses.

6.DA.4 The student will identify ways people curate and provide training data.

a. Identify and list ways people provide data that is used as training data.
b. Discuss the role of human intervention in curating training data.
c. Identify and describe the effect training data has on the accuracy of artificial intelligence systems.

Impacts of Computing (IC)

6.IC.1 The student will assess the impact of computing technologies on local society.

a. Explain how computing impacts innovation and describe the development of new computing technologies in communication, entertainment, and business.
b. Discuss how computing technologies have influenced various industries and sectors locally.
c. Research simple and complex problems that computing systems can be used to solve.
d. Analyze the implications of emerging technologies and potential real-world impact in the local community.

6.IC.2 The student will analyze the impact of screen time on physical and mental health.

a. Analyze and describe the impact of excessive technology usage may have on one's physical health.
b. Examine the impact of blue light on sleep patterns and regulations.
c. Propose strategies that provide alternatives of technology usage to promote physical activity.
d. Discuss the potential impact the use of social media may have on self-identity and mental health.
e. Define cyberbullying and its impact on one's health and well-being.
f. Discuss the possible effects of cyberbullying.
g. Identify ways to report illegal or psychologically maladaptive online behavior.

6.IC.3 The student will explore career pathways and identify how computer science and computational thinking practices align with these pathways.

a. Investigate a career of interest and determine how computer science and computational thinking practices are used in the chosen career.

6.IC.4 The student will identify copyrighted and licensed software material.

a. Identify the role of software licenses, including open-source, and why they are used.
b. Compare and contrast the positives and negatives of various software licenses.

6.IC.5 The student will describe the impacts of computing network architecture, including the role of the Internet in society.

a. Discuss ethical issues and laws related to accessibility, censorship, privacy, access, and safety while using the Internet.
b. Explain the role broadband connectivity has in social life, culture, and global economy.

6.IC.6 The student will investigate and analyze the impact of the progression and advancement of AI technologies on industries.

a. Discuss the type of industries that may be impacted by the use and integration of Artificial Intelligence (AI).
b. Compare and contrast the evolving nature of work across diverse industries because of the progression and advancement of Artificial Intelligence.

Networks and the Internet (NI)

6.NI.1 The student will outline the advantages and disadvantages of transmitting information over the Internet, including speed, reliability, cost, and security.

a. Explain the role of the Internet in social life, culture, and the economy.
b. Explain data transfer and the impact of connectivity speed when data is going from one device to another.
c. Compare the speed and reliability of various data transmission media.
d. Describe the advantages and disadvantages of transporting information over the Internet.

Seventh Grade: Computer Science

In Seventh Grade, students delve into the complexities of physical and digital security measures, critically evaluating threats and vulnerabilities associated with Internet use. They hone their programming skills through continued construction of algorithms, sequences, loops, and functions. Students leverage computational thinking and programming skills while applying an iterative design process to create programs as a means of creative expression and innovation. Through the use of purposeful computing devices students will collect data and define patterns, make inferences, and continue to develop computational models. Through the application of computational thinking, students will resolve hardware and software issues methodically. Students will deepen their understanding of concepts like machine learning, broadening their understanding of advanced technologies and their applications. Students will examine and evaluate the impact of computing technologies within society and globally. Additionally, students will identify individual strengths that can be used within computer science careers.

Algorithms and Programming (AP)

7.AP.1 **The student will apply computational thinking to design programs to accomplish a task as a means of creative expression or scientific exploration.**

a. Identify patterns and repeated steps in an algorithm, problem, or process.
b. Decompose an algorithm, problem, or process into sub-components.
c. Abstract relevant information to identify essential details.
d. Contrast various algorithms to solve reasoning problems when accomplishing a task.

7.AP.2 **The student will plan and implement algorithms that include sequencing, loops, variables, user input, conditional control structures, and functions using a block-based or text-based programming tool.**

a. Describe the concept of functions for use in a computer program.
b. Plan an algorithm using plain language, pseudocode, or diagrams.
c. Read and write programs that collect and use numeric and text data from users.
d. Read and write programs that contain nested conditionals and nested loops.

7.AP.3 **The student will use the iterative design process to create, test, and debug programs using a block-based or text-based programming language.**

a. Create and test programs that contain multiple control structures.
b. Trace and predict outcomes of programs.
c. Analyze the outcomes of programs to identify logic and syntax errors.
d. Analyze and describe the results of a program to assess validity of outcomes.
e. Revise and improve an algorithm to resolve errors or produce desired outcomes.

7.AP.4 The student will apply proper attribution when incorporating other sources into original work.

a. Apply proper methods of attribution when using work from the Internet and other sources.
b. Incorporate information or assets from the Internet into a program with proper attribution.

Computing Systems (CSY)

7.CSY.1 The student will design projects that use computing devices to collect and exchange data.

a. Apply project management skills to distribute tasks and maintain project timeline.
b. Generate ideas combining hardware and software components that can be used to collect and exchange data.
c. Describe how hardware and software can be used together to collect and exchange data.
d. Evaluate the usability of hardware and software to collect and exchange data.
e. Select the hardware and software components for project designs by considering factors such as functionality, cost, size, speed, accessibility, and aesthetics.

7.CSY.2 The student will apply computational thinking to troubleshoot and document hardware and software-related problems.

a. Apply systematic processes to resolve hardware, software, and connectivity-related problems.
b. Compile and record successful methods used to resolve problems for common hardware, software, and connectivity-related problems.

Cybersecurity (CYB)

7.CYB.1 The student will differentiate physical and digital security measures that protect electronic information.

a. Compare and contrast physical and digital security measures.
b. Research and synthesize the tradeoffs between usability and security.
c. Identify common threats and vulnerabilities associated with Internet use and Internet-based systems.
d. Identify potential solutions to address common threats and vulnerabilities.

Data and Analysis (DA)

7.DA.1 The student will utilize computational tools to visualize and evaluate data to draw conclusions and make predictions.

a. Develop computational models that simulate real-world phenomena, considering relevant variables and relationships.
b. Refine and modify computational models based on observed data and feedback, ensuring alignment with empirical evidence.
c. Analyze patterns and trends within observed data, comparing them with the predictions made by computational models.
d. Evaluate the effectiveness and accuracy of computational models in capturing and explaining the observed data.

7.DA.2 The student will explain the process and application of computational thinking in machine learning.

a. Explain how supervised, unsupervised, and/or reinforcement learning methods utilize decomposition, pattern recognition, abstraction, and algorithms to learn from and make decisions.
b. Explore neural networks and its role in machine learning and artificial intelligence.

Impacts of Computing (IC)

7.IC.1 The student will assess the national and global impact of computing technologies.

a. Discuss specific examples of how computing technologies have influenced various national and global industries and sectors.
b. Analyze the implications of emerging technologies and potential real-world impact nationally and globally.
c. Evaluate the environmental impact of computing technologies nationally and globally.

7.IC.2 The student will describe and explain the impact of screen time on interactions with others.

a. Describe the positive and negative impact of social media on socialization.
b. Research the type of data collected on social media and online platforms that monitor social interactions.
c. Describe and explain the evolution of screen time and the impact it has had on social interactions.
d. Create a social media usage plan that demonstrates safe practices, meaningful use, and a balanced approach.

7.IC.3 The student will identify individual preferences, skillset, and experiences and determine how these relate to a chosen computer science career field.

a. Use a career interest assessment to identify and categorize preferences, skillsets, and experiences.
b. Evaluate and connect personal skillsets, interests, talents, and values to a computer science career.

7.IC.4 The student will identify and apply strategies to prevent personal and public works from being pirated and plagiarized.

a. Discuss and describe intellectual property protections.
b. Research and list safeguards used to prevent intellectual property infringement.

7.IC.5 The student will evaluate the effect of Artificial Intelligence (AI) in various professions.

a. Research AI integration in various professions and evaluate its impact on the job market and society.
b. Examine and analyze the impact on job creation and changes in employment needs based on the use of AI.
c. Evaluate and explain the benefits and drawbacks of the implementation of AI technologies in various professions.

Networks and the Internet (NI)

7.NI.1 The student will describe and explain why protocols are essential in data transmission.

a. Define packet, router, and protocol.
b. Describe the process of sending a file through a network.
c. Explain the role of Internet Protocol (IP) addresses in transmitting information.
d. Explain how packets ensure reliable communication among computing devices.
e. Model how data is transmitted over networks and the Internet.

Eighth Grade: Computer Science

In Eighth Grade, students show mastery of understanding of the role of computing systems and data transmission over the Internet. Students contrast physical and digital safeguards implemented to protect electronic information from potential threats. Moreover, students assess the social and ethical implications of computing technologies from the perspective of both the creator and the consumer of computing technologies. The design and development of computing technologies are evaluated to account for the needs and wants of end users. Students continue to build upon previous knowledge and skills and create programs that contain multiple control structures. Their computational thinking skills are honed through the development of programs utilizing diverse data types, the development of computational models, and the use of pattern recognition and abstraction to make recommendations and predictions. As computer science skills and concepts can be applied to various careers, students will build upon their awareness of their skillsets and create education and training plans that foster continued pursuits of expanding their computer skills and knowledge to foster aspirations for post-secondary opportunities within computer science.

Algorithms and Programming (AP)

8.AP.1 **The student will apply computational thinking to construct programs to accomplish a task as a means of creative expression or scientific exploration.**

a. Identify patterns and repeated steps in an algorithm, problem, or process.
b. Decompose an algorithm, problem, or process into sub-components.
c. Abstract relevant information to identify essential details.
d. Use pseudocode, decision trees, or flowcharts to illustrate complex problems as algorithms.

8.AP.2 **The student will plan and implement algorithms that include sequencing, loops, variables, user input, conditional control structures, functions, and various data types.**

a. Describe the concept of input and output of various data types for use in a computer program.
b. Plan an algorithm using plain language, pseudocode, or diagrams.
c. Write and test algorithms expressed using block-based or text-based programming languages.

8.AP.3 **The student will use the iterative design process to create, test, and debug programs using a block-based or text-based programming language.**

a. Create and test programs that contain multiple control structures.
b. Trace and predict outcomes of programs.
c. Analyze the outcomes of programs to identify logic and syntax errors.
d. Analyze and describe the results of a program to assess validity of outcomes.
e. Revise and improve an algorithm to resolve errors or produce desired outcomes.

8.AP.4 The student will incorporate work from others into programs and projects.

a. Explain the role of Creative Commons licensing for the use and modification or "remixing" of information.
b. Utilize Creative Commons assets in a programming project.
c. Use and remix code from other projects within a programming project and provide proper attribution.

Computing Systems (CSY)

8.CSY.1 The student will recommend and design improvements to computing devices based on the needs of various users.

a. Analyze existing computing devices for advantages and limitations.
b. Recommend and design improvements to computing devices based on user interactions.

8.CSY.2 The student will apply computational thinking to troubleshoot and document hardware and software-related problems.

a. Apply systematic processes to resolve hardware, software, and connectivity-related problems.
b. Design an end-user document/guide to resolve hardware, software, and connectivity-related problems.

Cybersecurity (CYB)

8.CYB.1 The student will investigate and describe ways to protect sensitive data from malware and other attacks.

a. Identify impacts of hacking, ransomware, scams, phishing, fake vulnerability scans and the ethical and legal concerns.
b. Describe how cyber-attacks can affect a computing system.
c. Compare and contrast safe and unsafe computing practices.
d. Explore how industries and emerging technologies are addressing cyber solutions.
e. Model common prevention practices for cyber-attacks.

8.CYB.2 The student will investigate and explain how physical and digital security measures can protect electronic information for businesses, governments, and organizations.

a. Investigate and explain how physical and digital security measures are used to safeguard electronic information.
b. Research the advantages and limitations of different security measures in protecting users against security threats.

c. Explore how emerging technologies may affect methods to safeguard personal and public data.

Data and Analysis (DA)

8.DA.1 The student will create computational models to simulate events or represent phenomena.

a. Compare and contrast the use of computational models and simulations to analyze patterns and replicate phenomena.
b. Design and create complex computational models that simulate dynamic systems (abstraction), incorporating multiple variables and interactions.
c. Refine computational models based on generated outcomes.

8.DA.2 The student will evaluate computational models to analyze patterns and make recommendations or predictions.

a. Define data biases within a dataset and the unintended consequences that may impact data reliability and final analysis.
b. Analyze patterns and interpret data generated by computational models and simulations, identifying meaningful patterns and relationships.
c. Utilize data visualization techniques to communicate and present findings derived from computational models and simulations.

Impacts of Computing (IC)

8.IC.1 The student will assess the social impacts and ethical considerations of computing technologies.

a. Analyze the impact of sharing data through computing technologies.
b. Critique the role the Internet plays in social life, the global economy, and culture.
c. Evaluate online and print sources for credibility and reliability.
d. Research and discuss factors that impact access and availability to computing technologies.
e. Discuss ethical issues around cybersecurity and networks: censorship, privacy, safety, and access.

8.IC.2 The student will analyze and evaluate the ramifications of screen time in one's life.

a. Analyze scenarios or case studies to assess the impact of screen time on one's physical and mental health.

b. Justify the argument that excessive screen time and video games can have significant consequences for the physical, emotional, and cognitive development of children and adolescents.

8.IC.3 The student will identify opportunities for education, training, and preparation to enter into a chosen computer science career field.

a. Identify an education and training plan for a chosen computer science career.
b. Outline the use of computer science skills required in a chosen career.
c. Develop short-and long-term goals for a chosen career.
d. Research emerging trends in a chosen career path.

Networks and the Internet (NI)

8.NI.1 The student will model and describe the role of computing devices in transmitting data in and on computing networks and the Internet.

a. Identify the roles of computing devices: routers, switches, servers, and clients communicating over a network.
b. Design a network topology of computing devices.
c. Demonstrate how data is transmitted over networks and the Internet.
d. Analyze factors that strengthen or weaken network connectivity.

Middle School Elective: Computer Science

The Computer Science Middle School Elective standards are delineated for the implementation of a 9-week, 18-week, or 36-week implementation. Throughout this elective course, students will deepen their understanding of computer science, exploring its impact and historical context along with analysis of its current developments. Students will examine the influence and impact of computing within Virginia and globally. Students will utilize programming as a tool to create programs, integrating concepts learned K-8, while exploring various programming representations and analyzing their advantages and disadvantages.

Communication, collaboration, critical thinking, and creativity are addressed through standards that are intended to build upon project management skills and the iterative design process. Cybersecurity principles remain prominent, with a continued emphasis on safeguarding information and sensitive data. Creation is a significant component within the middle school course as students have acquired significant knowledge K-8 to now be fully prepared to create computational artifacts for predetermined tasks and solve problems. Additionally, students will delve into advanced concepts for data collection and its relationship to machine learning and Artificial Intelligence.

Standards are structured according to content strands, mirroring the organization of the K-8 standards. Each content strand delineates standards for 9-week, 18-week, and 36-week courses, illustrating the sequential progression of knowledge from the differing durations. Therefore, the content covered in an 18-week course includes the 9-week standards as well. Similarly, the 36-week course encompasses all standards outlined for both the 9-week and the 18-week course.

Algorithms and Programming (AP)

9-WEEKS:

MSCSE-9.AP.1 The student will apply computational thinking to evaluate and solve a problem.

a. Decompose a problem or process into sub-components.
b. Recognize characteristics or patterns to determine commonalities.
c. Abstract relevant information to identify essential details.
d. Use pseudocode and/or flowcharts to address complex problems as algorithms.

MSCSE-9.AP.2 The student will use iterative design process to create a program.

a. Identify the goal and objectives of the program.
b. Plan for the design or prototype of the program.
c. Develop an outline for the program's functionality.
d. Engage with peers to collect feedback on relevant aspects.

MSCSE-9.AP.3 The student will plan and implement algorithms that include loops, variables, user input, compound and nested conditional control structures, and procedure definitions that accept parameters using block-based or text-based programming.

a. Read and interpret algorithms expressed using plain language, pseudocode, and block-based or text-based programming languages.
b. Create an algorithm using plain language, pseudocode, or diagrams.
c. Implement programs that accept input values, use variables, and produce output.
d. Write and test algorithms using block-based or text-based programming languages.

MSCSE-9.AP.4 The student will use an interactive approach to trace, predict, test, and debug to improve existing programs and solve problems.

a. Trace a program for accuracy.
b. Analyze and describe the results of a program for validity.
c. Revise and improve an algorithm to resolve errors or produce desired outcomes.
d. Document programs to improve the ability to trace, test, and debug.

18-WEEKS:

MSCSE-18.AP.1 The student will improve existing solutions to problems to create new programs.

a. Categorize problems as classification, prediction, combinational search, or sequential decision problems.
b. Determine when problems can be solved with programs and automation.
c. Create a variety of programs while considering the needs and preferences of diverse user groups.
d. Utilize existing code, media, and libraries into original programs, and give attribution.

MSCSE-18.AP.2 The student will systematically use multiple test cases to verify the accuracy of a program.

a. Predict and test the outcome or output of multiple test cases.
b. Verify and refine the program based on the outcome of multiple test cases.

MSCSE-18.AP.3 The student will work collaboratively in an iterative design process to solve problems, including peer review and feedback.

a. Collaboratively plan, design, and revise programs.
b. Explain design choices, including constraints, and audiences.
c. Provide constructive feedback through peer review.
d. Reflect on collaborative experiences.

MSCSE-18.AP.4 The student will analyze, solve, and document a problem-solving process for others to duplicate the solution.

a. Analyze and decompose a problem.
b. Use abstraction to determine a solution to a problem.

MSCSE-18.AP.5 The student will investigate different coding languages.

a. Identify characteristics of block-based and text-based coding languages.
b. Analyze the advantages and disadvantages of block-based and text-based coding languages.

36-WEEKS:

MSCSE-36.AP.1 The student will develop a solution to a real-world problem using programming.

a. Identify problems that can be solved with a program.
b. Justify a proposed solution to a problem.
c. Use project management tools to support collaboration.
d. Engage in peer review and incorporate evaluative feedback through the design process.
e. Incorporate feedback provided through peer review to refine prototypes.

MSCSE-36.AP.2 The student will decompose problems and subcomponents into parts to facilitate the design, implementation, and review of programs.

a. Decompose problems into subcomponents to facilitate the creation of a program.
b. Use documentation to explain the purpose of a section of code and its relationship to other parts of a program.
c. Evaluate the limitations of models, algorithms, and programs considering multiple perspectives.
d. Systematically test and refine programs.

MSCSE-36.AP.3 The student will design and create algorithms using a text-based programming language.

a. Read and write programs that combine loops and conditional control structures.
b. Read and write programs using functions and procedures.

Computing Systems (CSY)

9-WEEKS:

MSCSE-9.CSY.1 The student will explore the structure, function, and interactions and interrelationships of hardware and software.

a. Explain the structure and function of hardware and software in computing devices.
b. Describe the interactions of hardware and software in computing systems.
c. Evaluate the usability of hardware and software from different perspectives.

MSCSE-9.CSY.2 The student will implement solutions to problems with computing devices and their components.

a. Abstract the problems with computing devices and apply appropriate troubleshooting strategies.
b. Propose and implement solutions to a variety of hardware and software problems.

18-WEEKS:

MSCSE-18.CSY.1 The student will design projects that utilize hardware and software components to collect and exchange data.

a. Explain the interactions of hardware and software components when collecting and exchanging data.
b. Justify the selection of hardware and software components for project designs by considering multiple factors: functionality, cost, size, speed, accessibility, and data bias.

MSCSE-18.CSY.2 The student will assess how computers interact with humans, other computing systems, and the environment.

a. Evaluate design and user operability to assess user accessibility, hardware and software performance, and compatibility.
b. Develop and apply criteria to evaluate a computing system for a given purpose and its intended environment.

36-WEEKS:

MSCSE-36.CSY.1 The student will analyze user interaction with computing devices and propose design improvements.

a. Analyze and evaluate the design of a computing device for the end user.
b. Propose recommendations for improvements to a computing device to include accessibility, functionality, cost, size, speed, accessibility, data bias, and aesthetics.
c. Investigate additional improvements available through emerging technologies.

Cybersecurity (CYB)

9-WEEKS:

MSCSE-9.CYB.1 The student will demonstrate ways to protect sensitive data from malware and other attacks.

a. Describe how cyber-attacks can affect a computing system.
b. Compare and contrast safe and unsafe computing practices.
c. Explore how industries and emerging technologies are addressing cyber solutions.
d. Model common prevention practices for cyber-attacks.

18-WEEKS:

MSCSE-18.CYB.1 The student will analyze tradeoffs between the use of public, privilege, private, and personal information.

a. Create strategies for protecting information.
b. Describe the benefits and drawbacks of each type of information.
c. Analyze the impact of misclassified information.
d. Outline common causes and reasons for data breaches involving information.
e. Describe the role of social engineering and human error in data breaches.

36-WEEKS:

MSCSE-36.CYB.1 The student will apply multiple methods of encryption and decryption to model the secure transmission of information.

a. Identify different types of data that can be encrypted.
b. Use simple encryption and decryption strategies to encode and decode a message.

Data and Analysis (DA)

9-WEEKS:

MSCSE-9.DA.1 The student will collect data and use computational thinking practices to transform and manipulate data.

a. Identify and use computing technologies to collect data sets.
b. Evaluate data sets and visualizations for limitations, reliability, and bias.
c. Identify encoding schemes used to represent similar data.
d. Transform data utilizing decomposition, pattern recognition, and abstraction to make inferences, predictions, and decisions about real-world phenomena.
e. Explore how machine learning, Artificial Intelligence (AI), and emerging technologies automate data analysis.

18-WEEKS:

MSCSE-18.DA.1 The student will use decomposition, pattern recognition, and abstraction to create and refine models and programs based on the data they have generated.

a. Create representations and visualizations of data.
b. Assess data reliability and biases in the data collection process, data visualizations, and existing automation.
c. Use and evaluate various schemes to represent data.
d. Explore how emerging technologies including Artificial Intelligence (AI) organize data, make decisions, and formulate predictions.

36-WEEKS:

MSCSE-36.DA.1 The student will utilize software tools and computer programs to create interactive data visualizations to analyze and better understand the real-world.

a. Communicate the significance of data visualizations and computation models in the real-world.
b. Use existing data visualizations that refresh in real-time to make predictions about real-world phenomena.
c. Create interactive data visualizations of real-world phenomena.

Impacts of Computing (IC)

9-WEEKS:

MSCSE-9.IC.1 The student will examine the impact of computing technologies that affect daily life and society.

a. Compare and contrast the benefits and risks of everyday use of computing technologies and their influences on daily life and society.
b. Discuss issues of digital citizenship, ethics, bias, and accessibility in relation to computing technologies and the Internet.
c. Identify and describe significant historical figures that made an impact in the computer science field.

MSCSE-9.IC.2 The student will investigate and analyze the impact of screen time on one's mental and physical health.

a. Investigate physical health effects associated with excessive screen time to include risk of obesity and related health issues.
b. Investigate social and emotional effects associated with excessive screen time to include development of social skills and emotional intelligence.
c. Investigate cognitive effects associated with excessive screen time to include the impact on attention span, concentration, and ability to focus.
d. Analyze and differentiate between the use of technology and screen time for instructional benefits compared to recreational activities.

e. Synthesize and report findings of the screen time usage and impact on one's mental and physical health.

MSCSE-9.IC.3 The student will explore careers that utilize computer science skills and practices in Virginia and globally.

a. Research computer science and information technology career pathways.
b. Discuss careers that use computer science skills and practices.

18-WEEKS:

MSCSE-18.IC.1 The student will examine relationships between historical advancements in computer science that influence the development of computing technologies.

a. Identify correlations between historical developments in computing technologies and changes in society, including automated decision-making and emerging computing technologies.
b. Explain how computing technologies can positively and negatively impact communities in Virginia and globally.
c. Examine how risk and benefits have changed based on the development of computing technologies.

MSCSE-18.IC.2 The student will explore computer science related career pathways.

a. Identify computer science related careers that relate to the student's interest.
b. Evaluate different computer science related careers and develop a career plan.

36-WEEKS:

MSCSE-36.IC.1 The student will assess the societal, environmental, and ethical impacts, and tradeoffs of computing technologies.

a. Describe tradeoffs between allowing information to be public and keeping information private and secure.
b. Evaluate policies, laws, and regulations for computing technologies.
c. Propose strategies to mitigate risks and increase benefits of computing technologies.
d. Assess the benefits and drawbacks of using Internet technologies by considering factors such as: data storage, cost, speed, reliability, and privacy.

Networks and the Internet (NI)

9-WEEKS:

MSCSE-9.NI.1 The student will explain that the Internet is a network of computing devices connected physically and/or wirelessly.

a. Explain the role of networks and the Internet in different computing contexts: gaming, web browsing, communication, streaming, or other common use cases.

b. Identify and describe hardware and software components that make up the Internet.
c. Explain how hardware and software components can be combined to collect and exchange data.
d. Compare and contrast different methods of data transfer.

18-WEEKS:

MSCSE-18.NI.1 The student will analyze the role of computing devices in transmitting data in computing networks and the Internet.

a. Define packet, protocol, server, and router.
b. Analyze the role of computing devices in transmitting data.
c. Model how data is transmitted across networks and the Internet.

36-WEEKS:

MSCSE-36.NI.1 The student will analyze the role of computing devices and network architecture in transmitting data in computing networks and the Internet.

a. Define client, server, router, and switches.
b. Identify how computing devices collect and exchange data.
c. Compare and contrast network topologies.
d. Illustrate the arrangement of topology among elements of a network.
e. Model how information is transmitted amongst multiple computing devices within a network.

Computer Science Foundations:

The Computer Science Foundations course is an introductory, high school computer science course. Its standards provide a structured progression from the middle school elective course while also offering a distinct learning opportunity for students who did not take the middle school elective course. The course emphasizes the academic depth of computer science with concepts and skills that would prepare students for advanced and specialized courses. Computational thinking is seen through its application in complex systems and designs of computing technologies. Students view various stakeholders within their work, evident in program development, cybersecurity safeguards, and technology usage. Advanced concepts within machine learning and Artificial Intelligence are addressed. Additionally, students acquire practical skills relevant to daily and professional life, including troubleshooting network issues and project management. Moreover, the course encourages students to analyze the societal, environmental, and ethical impacts of current and emerging technologies, fostering a deeper understanding of computing's broader and direct implications in society.

Algorithms and Programming (AP)

CSF.AP.1 The student will apply computational thinking to address a computational problem.

a. Identify real-world problems that are classification and prediction problems.
b. Decompose a problem or process into sub-components.
c. Implement abstractions to improve program modularity, reusability, and readability.
d. Identify computing-based solutions to address a computational problem.

CSF.AP.2 The student will use the iterative design process to create, test, and refine programs using a text-based programming language.

a. Create programs using a text-based programming language.
b. Document programs to improve the ability to trace, test, and debug.
c. Trace the execution of an algorithm and predict its results.
d. Analyze the outcomes of programs to identify logic and syntax errors.
e. Use multiple test cases to verify and refine the program.
f. Revise and improve an algorithm to resolve errors or produce desired outcomes.
g. Use version control and incorporate user feedback to refine program.

CSF.AP.3 The student will plan and implement algorithms and programs that include loops, variables, and compound and nested control structures using a text-based programming language.

a. Read and interpret algorithms and programs expressed using plain language, pseudocode, and text-based programming languages.

b. Create design documents using plain language, pseudocode, or diagrams.
c. Read and write algorithms and programs that accept multiple input values, use variables, and produce output.
d. Read and write algorithms and programs that include predefined functions and procedures with parameters and returns.
e. Compare several implementations of the same algorithm using different control structures.

CSF.AP.4 The student will design programs that use and manipulate data.

a. Determine appropriate data structures to address program specifications.
b. Apply basic computations on numeric and non-numeric data types.
c. Read and write programs that create, store, and manipulate primitive data.
d. Read and write programs that create, store, and manipulate linear collections of primitive data types: arrays or list.
e. Read and write programs that use relational, logical, and arithmetic expressions.
f. Read and write programs that traverse and manipulate data structures.

CSF.AP.5 The student will define and describe neural network learning algorithms.

a. Define and describe neural network learning algorithms.
b. Illustrate and describe a neural network structure.
c. Identify and discuss examples of computing technologies that utilize neural networks.
d. Compare and contrast a decision tree learning algorithm and a neural network learning algorithm.

CSF.AP.6 The student will investigate different coding languages.

a. Identify and describe characteristics of block-based and text-based coding languages.
b. Analyze the advantages and disadvantages of block-based and text-based coding languages.
c. Analyze the advantages and disadvantages of various text-based coding languages.

CSF.AP.7 The student will use search algorithms and sort algorithms.

a. Define the concept and role of a search algorithm.
b. Define the concept and role of a sort algorithm.
c. Compare and contrast bubble sort, quick sort, and merge sort.
d. Compare and contrast linear search and binary search.
e. Evaluate and determine the best search or sort algorithm to use based on intended results.

CSF.AP.8 The student will work collaboratively in an iterative design process to solve problems, including peer review and feedback.

a. Identify project management frameworks and methodologies that emphasize iteration.
b. Discuss the significance of communication and methods of communication when working collaboratively.
c. Distribute roles and responsibilities and adhere to predetermined timeline and/or project scope.
d. Collaboratively plan, design, and revise programs.
e. Provide constructive feedback through peer review.
f. Use project management tools to support collaboration.
g. Justify and explain design choices, including constraints, and audiences.
h. Reflect and discuss collaborative experience with team.

Computing Systems (CSY)

CSF.CSY.1 The student will explain the use of abstraction to hide underlying implementation of computing systems embedded in everyday objects.

a. Provide real-world examples of abstraction in computing.
b. Explain the role of abstraction to simplify complex systems.
c. Identify and describe levels of abstraction between application software, system software, and hardware layers.

CSF.CSY.2 The student will illustrate how computers create visual representations.

a. Describe how computers receive visual data from various sensors and tools.
b. Describe image processing techniques to include filtering, segmentation, and feature extraction.
c. Explain how computers use pattern recognition and classify data to interpret and make decisions.
d. Discuss ethical considerations related to the use of visual data and computer vision technologies.

CSF.CSY.3 The student will illustrate how computers can use visual representations of the world to solve problems.

a. Describe the parts of a network diagram and how they are related.
b. Explain the relationship between nodes, links, and other components of graphs.
c. Explain how a computer can solve a maze, find a route on a map, and use reasoning to solve problems.

CSF.CSY.4 The student will describe and explain the methods in which computers learn through the use of machine learning.

a. Compare and contrast the learning process of humans and computers.
b. Identify mathematical models used by supervised learning to produce classifications and predictions.

Cybersecurity (CYB)

CSF.CYB.1 The student will evaluate the methods of protecting data and computing systems, considering the context of the user and other stakeholders, with the level of cybersecurity risk.

a. Describe ways data and computing systems can be threatened by malware, ransomware, social engineering, phishing, and other cyberattacks.
b. Compare strategies to protect data and computing systems from malware, ransomware, social engineering, phishing, and other cyberattacks.

CSF.CYB.2 The student will identify and describe typical targets and perpetrators of cyberattacks.

a. Identify common targets and perpetrators of cyberattacks.
b. Identify ways data is automatically collected and generated that may or may not be evident to users.
c. Describe potential vulnerabilities when using publicly available networks.
d. Assess the impact of cyber threats on systems and people with diverse backgrounds, technical knowledge, or threat profiles.

CSF.CYB.3 The student will compare various security measures, considering tradeoffs between the usability and security of a computing system.

a. Evaluate tradeoffs between usability and security.
b. Analyze scenarios to determine tradeoffs between usability and security.
c. Propose recommendations for optimizing balance between usability and security in a given computing system.

Data and Analysis (DA)

CSF.DA.1 The student will identify and explain specific examples of real-world problems that can be effectively addressed using data analysis.

a. Describe the types of data that business, industry, and government entities collect and maintain.
b. Identify privacy and consumer protection issues that impact data representation.
c. Identify real-world problems that can be addressed through data analysis.

d. Compare two real-world datasets to identify how the values of features are encoded and represented.
e. Formulate questions to decompose a problem and develop a data project plan.

CSF.DA.2 The student will evaluate data collection and storage practices, including their impact on the stakeholders involved.

a. Identify methods for collecting and storing data of different data sizes.
b. Evaluate the technical and ethical implications of collecting and storing data from the perspectives of users, programmers, companies, and communities.
c. Identify impacts of bias in data collection and storage practices.
d. Analyze the impact of data quality, quantity, diversity, and other factors on the accuracy and reliability of data visualizations.
e. Research emerging technologies that have the capability to construct reasoning from stakeholder data.

CSF.DA.3 The student will investigate data collection practices and the role of consent, transparency, and responsible data handling.

a. Evaluate the quality of training data: completeness, accuracy, consistency, and relevance.
b. Analyze and discuss the ethical implications and social and economic impact of training data choices.

CSF.DA.4 The student will differentiate between the use of training data and reasoning models.

a. Explain the use of training data and the role it has in the development of machine learning models.
b. Explain the use of reasoning models and the role it has in the development of machine learning models.
c. Identify and discuss the similarities and differences between training data and reasoning models in artificial intelligence systems.

CSF.DA.5 The student will utilize data analysis to solve computational problems that make an impact and create solutions.

a. Formulate questions that require data collection.
b. Identify appropriate data to address predetermined question.
c. Define the stages of the data cycle and the interrelationship between each stage.
d. Identify and explain constraints of a data-driven approach.
e. Create a computational artifact of the data analysis results.

Impacts of Computing (IC)

CSF.IC.1 The student will research and hypothesize the societal, environmental, and ethical impacts of present and future computing technologies.

a. Identify the societal impacts of computing technologies and the various aspects of daily life and industry.
b. Evaluate the effect of advances in information technology on the economy, environment, and ethics, including advancements of Artificial Intelligence (AI), quantum computing, and emerging technologies.
c. Examine the environmental impact of computing technologies.
d. Propose strategies to address the ethical impacts and potential challenges of computing technologies.

CSF.IC.2 The student will examine the impact of screen-related distractions on productivity.

a. Identify digital tools and applications designed to monitor or regulate screen time usage.
b. Evaluate the impact of screen time management on productivity and well-being.
c. Examine and discuss the impact of screen time and social media on academic or workplace performance.

CSF.IC.3 The student will identify historical advancements in computer science and their parallels with changes in society.

a. Examine correlations between historical developments in computing technologies and changes in society.
b. Appraise contributions of pioneers in the field of computer science.
c. Explore the impact of Moore's Law on scientific and mathematical advancements.

CSF.IC.4 The student will explore computer science careers in Virginia and globally.

a. Research and explain the preparation and job outlook for computer science careers.
b. Examine current and future computer science career pathways involving emerging technologies.

Networks and the Internet (NI)

CSF.NI.1 The student will model the transmission of data across computing networks and the Internet.

a. Identify the roles of computing devices: routers, switches, servers, and clients.
b. Explain the role of Internet protocols: Hypertext Transfer Protocol Secure (HTTPS) and Transmission Control Protocol/Internet Protocol (TCP/IP) to provide reliable and secure data.

c. Analyze and create network topology diagrams.
d. Model how computing devices communicate via networks using TCP/IP protocols.
e. Identify common problems that impact network functionality.
f. Identify solutions to resolve common network issues.

Computer Science Principles:

The Computer Science Principles course is a continuation of the Computer Science Foundations course. This course has a strong focus on real-world applications, equipping students with a deeper understanding and expertise within computer science. Technical principles of abstraction are addressed as students gain understanding of the intricate relationship between hardware, system software, and application software. Students gain insight and understanding of the utilization of the C-I-A triad and its role in establishing cybersecurity best practices and implementation techniques to fortify defenses against threats and vulnerabilities. Data science is a pivotal component within the data and analysis content strand. Students utilize the data cycle for meaningful and informative analysis. Training data and data bias is a significant area of focus as students examine the real-world implications and uses of machine learning and Artificial Intelligence.

Algorithms and Programming (AP)

CSP.AP.1 The student will apply computational thinking to address a computational problem.

a. Identify and categorize real-world problems as classification, prediction, and sequential decision.
b. Identify the process used by specialized algorithms used for perceptual tasks using sensory inputs.
c. Decompose a computational problem or process into sub-components.
d. Use abstraction to improve program modularity, reusability, and readability.
e. Create a prototype that uses algorithms to address a complex computational problem.
f. Justify selected control structure(s) used to design algorithm.

CSP.AP.2 The student will design and create programs that use and manipulate data, include primitive data types and higher-order data structures.

a. Determine appropriate data structures to implement when given a programming problem or task.
b. Create, modify, store data in, and manipulate primitive data types like numbers, strings/characters, or Boolean values.
c. Create, modify, store data in, and manipulate linear and non-linear collections containing primitive and higher order data types: arrays, lists, objectives, or key-values structures.
d. Read and write programs that include linear data structures and processes a collection of data.

CSP.AP.3 The student will use the iterative design process to plan, implement, and create programs that satisfy user and design specifications.

a. Use project management skills to work individually and in teams.

b. Design an interactive program that accepts input from a variety of sources and produce output based on input.
c. Create a design specification document.
d. Design and create programs for various computing platforms.
e. Document programs to improve the ability to trace, test, and debug.
f. Trace the execution of an algorithm and predict its results.
g. Use proper attribution to incorporate code written by others.
h. Use multiple test cases to verify and refine program.
i. Revise and improve an algorithm to resolve errors or produce desired outcomes.
j. Solicit and synthesize user feedback to test and refine the program.
k. Apply best practices in developing programs: program development cycle, code styling, documentation, and version control.

CSP.AP.4 The student will weigh the affordances and constraints of different coding representations.

a. Compare and contrast schematic representation, pictorial representation, and other coding representations.
b. Generalize programming concepts, structures, and practices across coding representations.
c. Communicate the ways a coding representation or approach shapes solutions to problems.
d. Evaluate coding languages for specific real-world applications.

Computing Systems (CSY)

CSP.CSY.1 The student will compare the structures, functions, and interactions between application software, system software, and hardware.

a. Explain the role of abstraction and computing systems for user usability.
b. Explore the interdependent relationship between hardware and software and the effect on functionality and system architecture.
c. Analyze the components of hardware and software and propose solutions to increase functionality.
d. Describe the functions of an operating system, including resource management and process execution.
e. Construct a model to show the hierarchy of hardware, system software, and application software.

Cybersecurity (CYB)

CSP.CYB.1 The student will evaluate security technologies, techniques, and practices in terms of confidentiality, integrity, and availability.

a. Explain the C-I-A (Confidentiality, Integrity, and Availability) Triad.

b. Solve a cybersecurity problem and propose security measures related to confidentiality, integrity, and availability.
c. Compare information security and physical security measures to assess potential threats and vulnerabilities.

CSP.CYB.2 The student will explain the legal and ethical ramifications of cybersecurity breaches.

a. Describe state and federal laws that relate to cybersecurity and privacy.
b. Compare and contrast ethical and unethical hacking.
c. Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.

CSP.CYB.3 The student will explain the importance of protecting personally identifiable information (PII) and social identity.

a. Examine measures to prevent the disclosure of personally identifiable information (PII).
b. Compare and contrast ways to conduct threat analysis and to protect data and computing systems from data breaches.
c. Analyze scenarios and propose computing practices to protect personal information and reduce the risk of a data breach.

Data and Analysis (DA)

CSP.DA.1 The student will utilize, develop, evaluate, defend, and refine information about a dataset using computing technologies, techniques, and practices.

a. Identify the role of relational databases in storing data and in data utilization.
b. Analyze tradeoffs inherent in distilling raw data into data representations.
c. Evaluate data reliability and scalability.
d. Identify potential bias present in data representation practices.
e. Discuss the potential effect of data bias and provide recommendations on how to mitigate data bias.

CSP.DA.2 The student will collect and use training data.

a. Collect and clean diverse data sets to improve data quality and relevance.
b. Apply preprocessing techniques: missing values, normalization, and encoding categorical variables.
c. Create subsets of training data for training, validation, and testing.
d. Investigate potential imbalances within training data that could result in a biased model.

CSP.DA.3 The student will use supervised or unsupervised learning algorithm to train a model on real-world data.

a. Explain the difference between labeled and unlabeled data.
b. Evaluate a dataset used to train an artificial intelligence system.

c. Apply mathematical operations and algorithms to manipulate and extract insights from data sets.
d. Describe how supervised or unsupervised learning algorithms find patterns and make predictions.
e. Discuss how machines learn from data sets and derive new knowledge.
f. Describe how natural language processors (NLP) analyze data and produce output.

CSP.DA.4 The student will create and refine predictive models based on patterns in data.

a. Create and refine models or computational artifacts that can be used to make predictions and communicate effectively.
b. Justify tools and data visualizations selected to create and assess the model for accuracy.

Impacts of Computing (IC)

CSP.IC.1 The student will analyze the impacts of computing technologies across global societies.

a. Assess the impact of manufacturing and energy use on communities and the environment.
b. Analyze ways in which global collaboration is supported by new technologies.
c. Identify applications of quantum computing in various fields: scientific research, nonprofit entities, government agencies, and/or business industries.

CSP.IC.2 The student will analyze the long-term impact of excessive screen time use on one's mental and physical health.

a. Research and analyze the prevalence, causes, and long-term consequences of extended screen time usage.
b. Identify indicators of excessive social media use.
c. Propose techniques and strategies to mitigate or reduce the impact of excessive screen time usage.
d. Examine and discuss the impact of screen time and social media on academic or workplace performance.

CSP.IC.3 The student will analyze and design solutions to address local and global impacts of present and future computing technologies.

a. Analyze and evaluate equity, access, and influence on the distribution of computing resources in a global society.
b. Analyze the implications of emerging computing technologies to design solutions.
c. Create computing artifacts(s) that illustrates a solution to solve a problem locally or globally.

CSP.IC.4 The student will expand career explorations with work-based learning experiences.

a. Engage in work-based learning experiences involving computer science and related pathways.
b. Create a plan to navigate career pathways that include computer science skills and practices.

CSP.IC.5 The student will identify and describe the information processing capabilities of Artificial Intelligence in computing technologies.

a. Identify ways Artificial Intelligence applications can modify their behavior to respond to different people's emotional states.
b. Describe the role of natural language processing in computing technologies.
c. Examine ethical and privacy concerns related to Artificial Intelligence and propose recommendations to address these concerns.

Networks and the Internet (NI)

CSP.NI.1 The student will explain, analyze, and model computing networks and communication over the Internet.

a. Explain abstraction enabling computing devices to communicate to one another over an Internet connection.
b. Model abstractions and protocols enabling computers to transmit, receive, and interpret-data within networks and over the Internet.
c. Explain how abstraction enables different layers of Internet technology to build on one another.
d. Describe the seven layers of the OSI model.
e. Analyze issues pertaining to networks through the sevens layers of the OSI model.

CSP.NI.2 The student will explain design principles enabling large-scale operation of the Internet to connect devices and networks all over the world.

a. Explain design principles that permit scalability and reliability of connected devices on a network.
b. Describe issues that impact network functionality, scalability, and reliability and recommend solutions
c. Create a diagram to illustrate the communication connection between two distant devices.

Computer Science Programming (NCTE):

The Computer Science Programming standards emphasize mastery of text-based programming. This is an advanced-level programming course. This is a non-career technical education course. These standards provide students the opportunity to utilize programming as a tool to construct well designed programs, with a focus on efficiency, user protections, and vulnerability mitigation. Through continued application of computational thinking and iterative design, students will address real-world problems by developing programming solutions. The standards focus on the enhancement of technical skills, including the integration of application programming interfaces, understanding Big O Notation, and proficiency in recursive algorithms.

Algorithms and Programming (AP)

PRG.AP.1 The student will apply computational thinking to manage complex programs.

a. Identify and categorize real-world problems as classification, prediction, sequential decision, logical deduction, or statistical inference problem.
b. Analyze a large-scale computational problem, identify generalizable patterns, and implement a computing-based solution.
c. Decompose large-scale computational problems into subtasks and components processes and inter-relationships.
d. Implement and evaluate abstractions based on their modularity, reusability, and readability.

PRG.AP.2 The student will plan and implement programs that consist of compound conditionals, complex iterations, and complex computations using a text-based programming language.

a. Read and interpret algorithms expressed using plain language, and pseudocode. Read and write programs that include compound conditional execution and evaluate complex Boolean conditions.
b. Read and write programs that accept input from a variety of sources and produce output based on that input.
c. Read and write programs that include pre-defined and self-defined procedures.
d. Read and write programs that include functions with/without parameters, and functions with/without return values.
e. Read and write programs that consist of modular division, random number generation, substring manipulation and processing of individual characters.
f. Integrate external code with Application Programming Interface (APIs) and library calls.

PRG.AP.3 The student will use the iterative design process to create, test, and refine programs using a text-based programming language.

a. Trace the execution of iterative and recursive algorithms, illustrating output and changes in values of named variables.

b. Develop and systematically use a series of test cases to verify that a program performs according to its design specifications, including edge cases and all branches.
c. Use code review to evaluate the correctness, readability, and usability of a program.
d. Use debugging tools and user feedback to refine programs.
e. Modify existing program to improve functionality.

PRG.AP.4 The student will create programs that demonstrate an understanding of the data structures.

a. Use linear data structures: arrays, lists, and non-linear data structures.
b. Evaluate and convert data structures when appropriate.
c. Read and write programs that store, process, and manipulate 1D and 2D collections.
d. Identify how and when to use search and sort algorithms.
e. Read and write programs that include search and sort algorithms.

PRG.AP.5 The student will create programs that demonstrate an understanding of the interactions between classes and object-oriented design.

a. Define the role of inheritance, polymorphism, and encapsulation in object-oriented programming languages.
b. Use classes with instance data and methods to satisfy a design specification.
c. Organize programs methodically using comments and other organizational structures so that others can understand, interpret, and modify the program.

PRG.AP.6 The student will explain and justify program design and development decisions.

a. Explain the software life cycle and how it applies to the iterative design process.
b. Justify and communicate decisions and design elements.

PRG.AP.7 The student will interpret, adapt, test, debug, and refine algorithms for use in a particular context and evaluate for efficiency.

a. Use Big O notation to compare the benefits and drawbacks of using different algorithms for a particular process.

Computing Systems (CSY)

PRG.CSY.1 The student will evaluate the relationship between storage, processing, and efficiency, and analyze the role in program development.

a. Create programs that utilize persistent storage for program input and output.
b. Define the role of cache memory.
c. Analyze the impact of different types of memory on program processing speed.

d. Conduct a cost-benefit analysis for different types of memory.
e. Redesign a program to improve efficiency and performance.

Cybersecurity (CYB)

PRG.CYB.1 The student will evaluate current and emerging programming security practices.

a. Create programs that safeguard against user error.
b. Create programs that implement encryption algorithms.
c. Describe how software programs can meet basic requirements for security based on best practices.
d. Describe the impact of software vulnerabilities.
e. Evaluate methods developers use to protect unauthorized access to programs.

PRG.CYB.2 The student will write or adapt a program to avoid common vulnerabilities.

a. Understand the role of input validation in programming.
b. Develop code that validates input based on defined specifications.
c. Explain common vulnerabilities in program function and their impact.
d. Understand the impact of vulnerabilities on program function and security.

Data and Analysis (DA)

PRG.DA.1 The student will evaluate the tradeoffs between a variety of data organization and storage options.

a. Identify and compare data organization methods: variables, arrays, lists, trees, and schemas.
b. Assess and compare data storage options such as databases, file systems, local storage, and cloud storage, for scalability, reliability, privacy, and cost.
c. Evaluate the impact of data organization and storage choices on program performance, efficiency, and resource utilization.

PRG.DA.2 The student will use a variety of data types and structures in representing programmatic solutions to real-world problems.

a. Research and describe real-world reasoning problems that a reasoning algorithm can be used to sort data.
b. Read data summaries and visualizations and explain/translate into nontechnical terms for various audience groups.
c. Collect, use, and manipulate data from a variety of types and structures.
d. Utilize data analysis to create programmatic solutions and draw conclusion based on the results.

PRG.DA.3 The student will identify data biases in the data collection process and describe privacy concerns surrounding data collection and processing.

a. Use the data cycle in the collection and processing of data as part of the development of a program.
b. Describe how the data collection process should be focused, relevant, and limited to the scope of the project.
c. Analyze data to identify outliers or missing variables that could result in data biases.
d. Describe privacy considerations in the collection of data.

PRG.DA.4 The student will use a programming language to develop a data visualization.

a. Identify libraries and other resources that enable the visualization of data inputs.
b. Compare and contrast the methods of creating data visualizations, including programming languages and application software.
c. Develop a data visualization using a programming language's data processing function.
d. Create visualizations for descriptive and inferential statistical analysis based on the context and intended audience.
e. Apply mathematical operations and algorithms to manipulate and extract insights from data sets.
f. Justify the design, use, and effectiveness of different forms of data visualizations.

Impacts of Computing (IC)

PRG.IC.1 The student will examine the ramifications of technical and ethical design decisions when developing applications.

a. Use a design document to explain the reasoning for the design decisions made when developing an application.
b. Research the effects of technical design decisions on overall program function.
c. Examine and explain the impacts of unintended consequences related to program design.

PRG.IC.2 The student will use data to analyze the impact of screen time on one's mental and physical health.

a. Use statistical data to analyze the relationship between excessive screen time and attention span.
b. Analyze screen time usage data and propose recommendations to promote healthy habits.
c. Examine and discuss the impact of screen time and social media on academic or workplace performance.

PRG.IC.3 The student will expand computer science career explorations with work-based learning experiences.

a. Engage in work-based learning experiences involving computer science.

Networks and the Internet (NI)

No additional standards for Networks and the Internet.