Nevada Department of Education

Nevada Ready!

2018 NEVADA K-12 COMPUTER SCIENCE STANDARDS

The 2018 Nevada K-12 Computer Science Standards were written by contributors from all across the state of Nevada, under the direction of the Nevada Department of Education Office of Standards and Instructional Support.

June 26, 2018

Contents

Acknowledgements

The Nevada Department of Education Office of Standards and Instructional Support greatly appreciates the following members who shared their time and talent to facilitate the framing of these ground-breaking standards for our state. We appreciate your dedication and continued support for computer science for the students in Nevada.

K-12 Computer Science Standards Writing Team

Dr. Robert Askey
Touro University Nevada
(MS team)

Fran Bromley-Norwood
Clark County SD – Cheyenne H.S. (HS team)

Kristoffer Carroll
Regional Professional Development Pgm (MS team)

Sheilah Corcoran-Petrosky
Clark County SD – Southeast CTA (ES team)

Samuel Cornelius
Student – College of Southern Nevada (MS team)

Allen Gumm
Eureka County SD
(MS team)

Tina Holland
Washoe County SD
(ES team)

Robert Hollowood
Clark County SD – Stanton E.S. (ES team)

Sherri Kelley
Carson City SD – Carson H.S. (HS team)

Donna Levy
Clark County SD – Southwest CTA (HS team)

Amee Lombardi
Washoe CSD – Damonte Ranch H.S. (HS team)

Teri Mann
Clark County SD – Stanton E.S. (ES team)

Robert Maw
Carson City SD – (MS team)

Jaci McCune
Regional Professional Development Pgm (ES team)

Eric Mendelsohn
Industry/Parent – Code Central (ES team)

Mark Newburn
Nevada School Board of Education (HS team)

Randalyn Orong
Churchill County SD – Churchill M.S. (MS team)

Joseph Palmaffy
Charter School – Pinecrest Academy (MS team)

Paola Santos
Clark County SD – Williams E.S. (ES team)

Andreas Stefik
University Nevada Las Vegas
(HS team)

Scott Underwood
Clark County SD – Coronado H.S. (HS team)

Brenda Walker
Clark County SD – Diskin E.S. (ES team)

Irene Waltz
Carson City SD – Fritsch E.S. (ES team)

Lance West
Elko County SD – Spring Creek H.S. (HS team)

Robert Williams
Nye County School District
(MS team)

Robin Williams
Nye County SD – Pahrump Valley H.S. (HS team)

Acknowledgements

K-12 Computer Science Standards Internal Review Team

Randall Boone
Clark County

Patty Davies
Clark County

Lloyd Mann
Clark County

Kelly McFall
Clark County

Kimberly Moody
Clark County

Erin Murtaugh
Clark County

Robbie Pearce
Clark County

Michael Seal
Lyon County

Pavel Solin
Washoe County

Regional Professional Development Program Workgroup

Provided connections with other Nevada Academic Content Standards

Nathalie Brugman
Family Engagement

Shan Cannon
EL Literacy

Kris Carroll
Secondary Science

Stacy Cohen
EL Science

Kathy Dees
EL Math

Sue Dolphin
EL Math

Mendy Henry
EL Literacy

David Janssen
EL Math

Carl Jarvinen
Secondary Science

Sara Lasley
Secondary Literacy

Robyn Markovic
EL Literacy

Jaci McCune
Prof Learning Facilitator

Candice Meiries
Secondary Math

Cindy Ortiz
Secondary Math

Valerie Seals
EL Literacy

Bret Sibley
Secondary Science

Laurie Thompson
Secondary Literacy

Acknowledgements

Nevada Department of Education
700 East Fifth Street
Carson City, Nevada 89701
http://www.doe.nv.gov

Dr. Steve Canavero
State Superintendent of Public Instruction

Office of Standards and Instructional Support

Dave Brancamp Director	Andre DeLeon Science/Health/PE	Darrin Hardman ELA
Cindi Chang Computer Science	Lisa Ford ELA – Fine Arts	Mary Holsclaw Social Studies
Heather Crawford-Ferre Nevada Ready 21	Tracy Gruber Math	Amanda Pinter Administrative Assistant

Overview

Introduction

Our world is increasingly dependent upon technology, computers, and the training necessary to compete in a global digital age. *The Nevada K-12 Computer Science Standards* provide the rigor that our students need to become proficient at problem solving, computational thinking, and innovators of computational artifacts, rather than just users of them. Computer Science intersects with every subject area our students will take in school and every career path they may choose to pursue following graduation.

The *Nevada K-12 Computer Science Standards* cover the core concepts, and many sub concepts, in the field of computer science that every student should master to become computationally literate members of society. The *Nevada K-12 Computer Science Standards* also include the practice behaviors that our students will be actively engaged in as they learn, adapt, and employ these standards to produce computational artifacts. All curricular decisions that districts, schools, and teachers make regarding computer science education for every K-12 student in Nevada should be based on these standards.

Mission

Our mission is to provide engaging and rigorous computer science education for ALL Nevada students, regardless of their age, race, gender, disability, socioeconomic level, or what school they attend, and to prepare them for a wide variety of postsecondary experiences and careers in the digital age. Our students are not only consumers of technology, but creators of it.

Vision

Nevada's vision is that our students are informed citizens who can:

- critically engage in public discussion on computer science topics,
- develop as learners, users, and creators of CS knowledge and artifacts,
- better understand the role of computing in the world around them; and
- learn, perform, and express themselves in other subjects and interests.

Process

The *Nevada K-12 Computer Science Standards* are the product of a diverse group of contributors, including K-12 educators, administrators, public charter, higher education, the State Board of Education, the Regional Professional Development Program, industry, parents, and student representation. The 26 member writing team represented eight of Nevada's seventeen counties; members from all of the remaining counties were invited to participate on the internal review team. The grade-band teams consisted of nine elementary-, eight middle-, and nine high-school writers. The standards were written over the course of three days: August 23 – 25, 2017. The draft of these standards was published for a 30-day public review on the Nevada Department of Education's website on November 1, 2017. All districts were notified by public memorandum on this date. The public was invited to provide feedback until November 30, 2017, after which revisions were made to the standards document based on data collected during the review period. The final draft went before the Commission on Professional Standards, then before the Nevada Board of Education, and finally before the Legislative Commission for final ratification on June 26, 2018.

Concepts

The Nevada K-12 Computer Science Standards represent five core concepts in the field of computer science. There are multiple sub concepts listed that represent specific ideas within each core concept. We have provided a brief overview of each sub concept for further clarification, (see chart below). In addition, there are five cross-cutting topics that are interwoven within each core concept throughout the standards, but do not have stand-alone descriptions, including Abstraction, System Relationships, Human-Computer Interaction, User Inspired Software Design, Privacy and Security, and Communication and Coordination.

Concept	Sub concept	Overview
Algorithms and Programming	Algorithms	People evaluate and select algorithms based on performance, reusability, and ease of implementation. Knowledge of common algorithms improves how people develop software, secure data, and store information.
	Control	Programmers consider tradeoffs related to implementation, readability, and program performance when selecting and combining control structures.
	Modularity	Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose. These modules can be procedures within a program; combinations of data and procedures or independent, but interrelated, programs. Modules allow for better management of complex tasks.
	Program Development	Diverse teams can develop programs with broad impact through careful review and by drawing on the strengths of members in different roles. Design decisions often involve tradeoffs. The development of complex programs is aided by resources such as libraries and tools to edit and manage parts of the program. Systematic analysis is critical for identifying the effects of lingering bugs.
	Variables	Data structures are used to manage program complexity. Programmers choose data structures based on functionality, storage, and performance tradeoffs.
Computing Systems	Devices	Many everyday objects contain computational components that sense and act on the world. In early grades, students learn features and applications of common computing devices. As they progress, students learn about connected systems and how interaction between humans and devices influences design decisions.
	Hardware and Software	Computing systems use hardware and software to communicate and process information in digital form. In early grades, students learn how systems use both hardware and software to represent and process information. As they progress, students gain a deeper understanding of the interaction between hardware and software at multiple levels within computing systems.

Concept	Sub concept	Overview
	Troubleshooting	When computing systems do not work as intended, troubleshooting strategies help people solve the problem. In early grades, students learn that identifying the problem is the first step to fixing it. As they progress, students learn systematic problem-solving processes and how to develop their own troubleshooting strategies based on a deeper understanding of how computing systems work.
Data and Analysis	Collection, Visualization, and Transformation	Data is collected with both computational and non-computational tools and processes. In early grades, students learn how data about themselves and their world is collected and used. As they progress, students learn the effects of collecting data with computational and automated tools.
	Inference and Models	Data science is one example where computer science serves many fields. Computer science and science use data to make inferences, theories, or predictions based upon data collected from users or simulations. In early grades, students learn about the use of data to make simple predictions. As they progress, students learn how models and simulations can be used to examine theories and understand systems and how predictions and inferences are affected by more complex and larger data sets.
	Storage	Data can be composed of multiple data elements that relate to one another. For example, population data may contain information about age, gender, and height. People make choices about how data elements are organized and where data is stored. These choices affect cost, speed, reliability, accessibility, privacy, and integrity.
Impacts of Computing	Culture	The design and use of computing technologies and artifacts can improve, worsen, or maintain inequitable access to information and opportunities.
	Safety, Law and Ethics	Laws govern many aspects of computing, such as privacy, data, property, information, and identity. These laws can have beneficial and harmful effects, such as expediting or delaying advancements in computing and protecting or infringing upon people's rights. International differences in laws and ethics have implications for computing.
	Social Interactions	Many aspects of society, especially careers, have been affected by the degree of communication afforded by computing. The increased connectivity between people in different cultures and in different career fields has changed the nature and content of many careers.
Networks and the Internet	Cybersecurity	Transmitting information securely across networks requires appropriate protection. In early grades, students learn how to protect their personal information. As they progress, students learn increasingly complex ways to protect information sent across networks.
	Network Communication and Organization	Computing devices communicate with each other across networks to share information. In early grades, students learn that computers connect them to other people, places, and things around the world. As they progress, students gain a deeper understanding of how information is sent and received across different types of networks.

The K-12 Computer Science Framework, led by the Association for Computing Machinery, Code.org, Computer Science Teachers Association, Cyber Innovation Center, and National Math and Science Initiative in partnership with states and districts, informed the development of this work. View the framework at http://k12cs.org.

Practices

The Nevada K-12 Computer Science Standards incorporate seven practices. By Grade 12, it is expected that every computationally literate student will engage with these practice behaviors as they learn the standards and develop computational artifacts. The interrelated practices are listed in the chart below in an order that simulates the developmental process taken to produce computational artifacts.

Identifier	Practice
P1	Fostering an Inclusive Computing Culture
P1.1	Include the unique perspectives of others and reflect on one's own perspectives when designing and developing computational products.
P1.2	Address the needs of diverse end users during the design process to produce artifacts with broad accessibility and usability.
P1.3	Employ self- and peer-advocacy to address bias in interactions, product design, and development methods.
P2	Collaborating Around Computing
P2.1	Cultivate working relationships with individuals possessing diverse perspectives, skills, and personalities.
P2.2	Create team norms, expectations, and equitable workloads to increase efficiency and effectiveness.
P2.3	Solicit and incorporate feedback from, and provide constructive feedback to, team members and other stakeholders.
P2.4	Evaluate and select technological tools that can be used to collaborate on a project.
P3	Recognizing and Defining Computational Problems
P3.1	Identify complex, interdisciplinary, real-world problems that can be solved computationally.
P3.2	Decompose complex real-world problems into manageable subproblems that could integrate existing solutions or procedures.
P3.3	Evaluate whether it is appropriate and feasible to solve a problem computationally.
P4	Developing and Using Abstractions
P4.1	Extract common features from a set of interrelated processes or complex phenomena.

Identifier	Practice
P4.2	Evaluate existing technological functionalities and incorporate them into new designs.
P4.3	Create modules and develop points of interaction that can apply to multiple situations and reduce complexity.
P4.4	Model phenomena and processes and simulate systems to understand and evaluate potential outcomes.
P5	Creating Computational Artifacts
P5.1	Plan the development of a computational artifact using an iterative process that includes reflection on and modification of the plan, taking into account key features, time and resource constraints, and user expectations.
P5.2	Create a computational artifact for practical intent, personal expression, or to address a societal issue.
P5.3	Modify an existing artifact to improve or customize it.
P6	Testing and Refining Computational Artifacts
P6.1	Systematically test computational artifacts by considering all scenarios and using test cases.
P6.2	Identify and fix errors using a systematic process.
P6.3	Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability, and accessibility.
P7	Communicating About Computing
P7.1	Select, organize, and interpret large data sets from multiple sources to support a claim.
P7.2	Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose.
P7.3	Articulate ideas responsibly by observing intellectual property rights and giving appropriate attribution.

The K-12 Computer Science Framework, led by the Association for Computing Machinery, Code.org, Computer Science Teachers Association, Cyber Innovation Center, and National Math and Science Initiative in partnership with states and districts, informed the development of this work. View the framework at http://k12cs.org.

How to Read the Standards

The *Nevada K-12 Computer Science Standards* are divided into Grades K, 1, 2, 3, 4, 5, 6-8, and 9-12. There are also Advanced 9-12 higher level standards included that are optional and intended to be used by teachers for either advanced students, to incorporate into upper level courses, or for Career and Technical Education (CTE) programs.

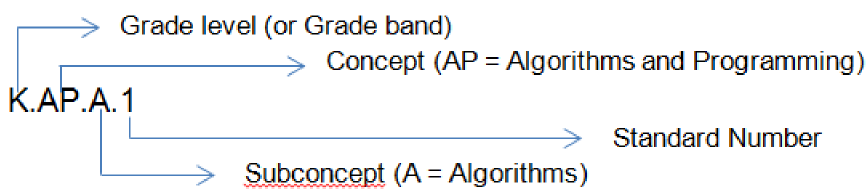
The standards are divided by the five main concepts mentioned earlier. Those main concepts include algorithms and programming, computing systems, data and analysis, impacts of computing, and networks and the Internet. Within each main concept there may be two to five sub concepts represented, such as algorithms, program development, variables, troubleshooting, or cybersecurity. Each standard ties back to a practice that every computationally literate student will engage with as they develop computational artifacts (see Figure A).

Figure A

Practice Connection	Concept Connection
Practice 4. Developing and Using Abstractions *4. Model phenomena and processes and simulate systems to understand and evaluate potential outcome* (K.AP.A.1). **Practice 6. Testing and Refining Computational Artifacts** *2. Identify and fix errors using a systematic process* (K.AP.PD.1).	**Algorithms and Programming** • AP.A: Algorithms - *People follow and create processes as part of daily life. Many of these processes can be expressed as algorithms that computers can follow* (K.AP.A.1). • AP.PD: Program Development - *People develop programs collaboratively and for a purpose, such as expressing ideas or addressing problems* (K.AP.PD.1).

The standards are coded as follows: (see Figure B).

Figure B



Algorithms and Programming

Students who demonstrate understanding can:

Indicator	Standard
K.AP.A.1	Model daily processes by creating and following sets of step-by-step instructions (algorithms) to complete tasks.
K.AP.PD.1	Identify and fix (debug) errors in a sequence of instructions (algorithms) that includes loops.

Practice Connection	Concept Connection
Practice 4. Developing and Using Abstractions *4. Model phenomena and processes and simulate systems to understand and evaluate potential outcome* (K.AP.A.1). **Practice 6. Testing and Refining Computational Artifacts** *2. Identify and fix errors using a systematic process* (K.AP.PD.1).	**Algorithms and Programming** ● AP.A: Algorithms - *People follow and create processes as part of daily life. Many of these processes can be expressed as algorithms that computers can follow* (K.AP.A.1). ● AP.PD: Program Development - *People develop programs collaboratively and for a purpose, such as expressing ideas or addressing problems* (K.AP.PD.1).

	Connections to other NVACS at this grade level
NVACS for Science	(K.AP.PD.1) – CCC: Cause and Effect, CCC: Patterns (K.AP.A.1) – SEP: Obtaining, evaluating and communicating information, CCC: Systems and System Models
NVACS for Mathematics	SMP 6, 7 & 8
NVACS for ELA	(K.AP.A.1) – RF.K.A, RI.K.7 (K.AP.PD.1) – RL.K.2, W.K.6
NVACS for Social Studies	None

Computing Systems

Students who demonstrate understanding can:

Indicator	Standard
K.CS.HS.1	Use appropriate terminology in identifying and describing the function of common physical components of computing systems (hardware). *For example: monitor, keyboard, mouse, earbuds, headphones, printer.*
K.CS.HS.2	Recognize some computing devices (e.g., computer, smartphone) can perform a variety of tasks and some computing devices are specialized (e.g., navigation system, game controller).

Practice Connection	Concept Connection
Practice 7. Communicating About Computing *2. Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose* (K.CS.HS.1 and K.CS.HS.2).	**Computing Systems** • CS.HS: Hardware and Software - *A computing system is composed of hardware and software. Hardware consists of physical components, while software provides instructions for the system. These instructions are represented in a form that a computer can understand* (K.CS.HS.1 and K.CS.HS.2).

	Connections to other NVACS at this grade level
NVACS for Science	(K.CS.HS.1) – Structure and Function
NVACS for Mathematics	SMP 2, 3, & 5
NVACS for ELA	(K.CS.HS.1) – L.K.5c, L.K.6, RI.K.2 (K.CS.HS.2) – W.K.6
NVACS for Social Studies	None

Data and Analysis

Students who demonstrate understanding can:

Indicator	Standard
K.DA.S.1	Recognize that data can be collected and stored on different computing devices over time.

Practice Connection	Concept Connection
Practice 4. Developing and Using Abstractions *2. Evaluate existing technological functionalities and incorporate them into new designs* (K.DA.S.1).	**Data and Analysis** ● DA.S: Storage - *Computers store data that can be retrieved later. Identical copies of data can be made and stored in multiple locations for a variety of reasons, such as to protect against loss* (K.DA.S.1).

	Connections to other NVACS at this grade level
NVACS for Science	None
NVACS for Mathematics	SMP 5, K.MD.B.3
NVACS for ELA	W.K.6
NVACS for Social Studies	None

Impacts of Computing

Students who demonstrate understanding can:

Indicator	Standard
K.IC.C.1	Understand how computing devices have changed people's lives.
K.IC.SI.1	Exhibit good digital citizenship using technology safely, responsibly, and ethically.

Practice Connection	Concept Connection
Practice 1. Foster an Inclusive Computing Culture 1. *Include the unique perspectives of others and reflect on one's own perspectives when designing and developing computational products* (K.IC.C.1). **Practice 2. Collaborating Around Computing** 1. *Cultivate working relationships with individuals possessing diverse perspectives, skills, and personalities* (K.IC.SI.1).	**Impacts of Computing** ● IC.C: Culture - *Computing technology has positively and negatively changed the way people live and work. Computing devices can be used for entertainment and as productivity tools, and they can affect relationships and lifestyles* (K.IC.C.1). ● IC.SI: Social Interactions - *Computing has positively and negatively changed the way people communicate. People can have access to information and each other instantly, anywhere, and at any time, but they are at the risk of cyberbullying and reduced privacy* (K.IC.SI.1).

	Connections to other NVACS at this grade-level
NVACS for Science	(K.IC.C.1) – CCC: Cause and Effect (K.IC.C.A) – K-ESS3-2 (Influence of Engineering, Technology and Science on society and the natural world. People depend on various technologies in their lives: human life would be very different without technology.)
NVACS for Mathematics	None
NVACS for ELA	(K.IC.C.1) – RI.K.3 (K.IC.SI.1) – SL.K.1
NVACS for Social Studies	(K.IC.C.1) – SS.K.17 (K.IC.SI.1) – SS.K.17

Networks and the Internet

Students who demonstrate understanding can:

Indicator	Standard
K.NI.C.1	Explain that a password helps protect the privacy of information.

Practice Connection	Concept Connection
Practice 7. Communicating About Computing *3. Articulate ideas responsibly by observing intellectual property rights and giving appropriate attribution* (K.NI.C.1).	**Networks and the Internet** ● NI.C: Cybersecurity - *Connecting devices to a network or the Internet provides great benefit. Care must be taken to use authentication measures, such as strong passwords, to protect devices and information from unauthorized access* (K.NI.C.1).

	Connections to other NVACS at this grade level
NVACS for Science	(K.NI.C.1) – CCC: Cause and Effect
NVACS for Mathematics	SMP 3 & 4
NVACS for ELA	(K.NI.C.1) – W.K.2
NVACS for Social Studies	None

Algorithms and Programming

Students who demonstrate understanding can:

Indicator	Standard
1.AP.PD.1	Describe the iterative process of program development (including terminology, steps taken, and the logic of choices).
1.AP.V.1	Model the way programs store and manipulate data by using numbers or other symbols to represent information.

Practice Connection	Concept Connection
Practice 7. Communicating About Computing *2. Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose* (1.AP.PD.1). **Practice 4. Developing and Using Abstractions** *4. Model phenomena and processes and simulate systems to understand and evaluate potential outcomes* (1.AP.V.1).	**Algorithms and Programming** ● AP.PD: Program Development - *People develop programs collaboratively and for a purpose, such as expressing ideas or addressing problems* (1.AP.PD.1). ● AP.V: Variables - *Information in the real world can be represented in computer programs. Programs store and manipulate data, such as numbers, words, colors, and images. The type of data determines the actions and attributes associated with it* (1.AP.V.1).

	Connections to other NVACS at this grade level
NVACS for Science	(1.AP.PD.1) – CCC: Patterns, SEP: Obtaining, Evaluating and Communicating Information (1.AP.V.1) – CCC: Systems and System Models, SEP: Analyzing and Interpreting Data
NVACS for Mathematics	SMP 2, 3, 4; 1.MD.C.4
NVACS for ELA	(1.AP.PD.1) – W.1.2, W.1.3 (1.AP.V.1) – SL.1.5
NVACS for Social Studies	None

Computing Systems

Students who demonstrate understanding can:

Indicator	Standard
1.CS.D.1	Select and operate appropriate device and software to perform a variety of tasks, and recognize that users have different needs and preferences for the technology they use.

Practice Connection	Concept Connection
Practice 1. Fostering an Inclusive Computing Culture *1. Include the unique perspectives of others and reflect on one's own perspectives when designing and developing computational products* (1.CS.D.1).	**Computing Systems** ● CS.D: Devices - *People use computing devices to perform a variety of tasks accurately and quickly. Computing devices interpret and follow the instructions they are given literally* (1.CS.D.1).

	Connections to other NVACS at this grade level
NVACS for Science	(1.CS.D.1) – SEP: Asking Questions and Defining Problems
NVACS for Mathematics	SMP 5
NVACS for ELA	None
NVACS for Social Studies	None

Data and Analysis

Students who demonstrate understanding can:

Indicator	Standard
1.DA.S.1	Recognize that a variety of data (e.g., music, video, images, text) can be stored in and retrieved from a computing device.

Practice Connection	Concept Connection
Practice 4. Developing and Using Abstractions *2. Evaluate existing technological functionalities and incorporate them into new designs* (K.DA.S.1).	**Data and Analysis** ● DA.S: Storage - *Computers store data that can be retrieved later. Identical copies of data can be made and stored in multiple locations for a variety of reasons, such as to protect against loss* (1.DA.S.1).

	Connections to other NVACS at this grade level
NVACS for Science	None
NVACS for Mathematics	SMP 5
NVACS for ELA	(1.DA.S.1) – W.1.6
NVACS for Social Studies	None

Impacts of Computing

Students who demonstrate understanding can:

Indicator	Standard
1.IC.SI.1	Work respectfully and responsibly with others online.

Practice Connection	Concept Connection
Practice 2. Collaborating Around Computing 1. *Cultivate working relationships with individuals possessing diverse perspectives, skills, and personalities* (1.IC.SI.1).	**Impacts of Computing** ● IC.SI: Social Interactions - *Computing has positively and negatively changed the way people communicate. People can have access to information and each other instantly, anywhere, and at any time, but they are at the risk of cyberbullying and reduced privacy* (1.IC.SI.1).

	Connections to other NVACS at this grade level
NVACS for Science	None
NVACS for Mathematics	None
NVACS for ELA	(1.IC.SI.1) – SL.1.1
NVACS for Social Studies	(1.IC.SI.1) – SS.1.18, SS.1.20, SS.1.22

Networks and the Internet

Students who demonstrate understanding can:

Indicator	Standard
1.NI.C.1	Explain why we keep personal information (e.g., name, location, phone number, home address) private.

Practice Connection	Concept Connection
Practice 7. Communicating About Computing *3. Articulate ideas responsibly by observing intellectual property rights and giving appropriate attribution* (1.NI.C.1).	**Networks and the Internet** ● NI.C: Cybersecurity - *Connecting devices to a network or the Internet provides great benefit, care must be taken to use authentication measures, such as strong passwords, to protect devices and information from unauthorized access* (1.NI.C.1).

	Connections to other NVACS at this grade level
NVACS for Science	(1.NI.C.1) – CCC: Cause and Effect
NVACS for Mathematics	SMP 3
NVACS for ELA	(1.NI.C.1) – W.1.2
NVACS for Social Studies	Connect with financial literacy for password safety and ID theft (indicator coming soon as those standards are being approved)

Algorithms & Programming

Students who demonstrate understanding can:

Indicator	Standard
2.AP.C.1	Develop programs with sequences and loops, to express ideas or address a problem.
2.AP.M.1	Break down (decompose) the steps needed to solve a problem into a precise sequence of instructions.
2.AP.PD.1	Develop plans that describe a program's sequence of events, goals, and expected outcomes.
2.AP.PD.2	Give attribution (credit) when using the ideas and creations of others while developing programs.

Practice Connection	Concept Connection
Practice 3. Recognizing and Defining Computational Problems 2. *Decompose complex real-world problems into manageable subproblems that could integrate existing solutions or procedures* (2.AP.M.1). **Practice 5. Creating Computational Artifacts** 1. *Plan the development of a computational artifact using an iterative process that includes reflection on and modification of the plan, taking into account key features, time and resource constraints, and user expectations* (2.AP.PD.1). 2. *Create a computational artifact for practical intent, personal expression, or to address a societal issue* (2.AP.C.1). **Practice 7. Communicating About Computing** 2. *Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose* (2.AP.PD.1). 3. *Articulate ideas responsibly by observing intellectual property rights and giving appropriate attribution* (2.AP.PD.2).	**Algorithms & Programming** ● AP.C: Control - *Computers follow precise sequences of instructions that automate tasks. Program execution can also be nonsequential by repeating patterns of instructions and using events to initiate instructions* (2.AP.C.1). ● AP.M: Modularity - *Complex tasks can be broken down into simpler instructions, some of which can be broken down even further. Likewise, instructions can be combined to accomplish complex tasks* (2.AP.M.1). ● AP.PD: Program Development - *People develop programs collaboratively and for a purpose, such as expressing ideas or addressing problems* (2.AP.PD.1 and 2.AP.PD.2).

	Connections to other NVACS at this grade level
NVACS for Science	(2.AP.C.1) – SEP: Developing and Using Models, SEP: Asking Questions and Defining Problems (2.AP.M.1) – SEP: Analyzing and Interpreting Data (2.AP.PD.1) – ETS 1B: Developing Possible Solutions, SEP: Obtaining, Communicating, and Evaluating Data, SEP: Planning and Carrying Out Investigations
NVACS for Mathematics	None
NVACS for ELA	(2.AP.C.1) – W.2.2, RI.2.3 (2.AP.M.1) – W.2.2 (2.AP.PD.1) – L.2.4 (2.AP.PD.2) – SL.2.2, SL.2.4
NVACS for Social Studies	None

Computing Systems

Students who demonstrate understanding can:

Indicator	Standard
2.CS.T.1	Describe basic hardware and software problems using accurate terminology.

Practice Connection	Concept Connection
Practice 6. Testing and Refining Computational Artifacts *2. Identify and fix errors using a systematic process* (2.CS.T.1). **Practice 7. Communicating About Computing** *2. Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose* (2.CS.T.1).	**Computing Systems** ● CS.T: Troubleshooting - *Computing systems might not work as expected because of hardware or software problems. Clearly describing a problem is the first step toward finding a solution* (2.CS.T.1).

	Connections to other NVACS at this grade level
NVACS for Science	None
NVACS for Mathematics	SMP 2 & 6
NVACS for ELA	(2.CS.T.1) – L.2.4e
NVACS for Social Studies	None

Data & Analysis

Students who demonstrate understanding can:

Indicator	Standard
2.DA.S.1	Store, copy, search, retrieve, modify, and delete information using a computing device and define the information stored as data.

Practice Connection	Concept Connection
Practice 4. Developing and Using Abstractions *2. Evaluate existing technological functionalities and incorporate them into new designs* (2.DA.S.1).	**Data & Analysis** ● DA.S: Storage - *Computers store data that can be retrieved later. Identical copies of data can be made and stored in multiple locations for a variety of reasons, such as to protect against loss* (2.DA.S.1).

	Connections to other NVACS at this grade level
NVACS for Science	None
NVACS for Mathematics	None
NVACS for ELA	(2.DA.S.1) – W.2.7
NVACS for Social Studies	None

Impacts of Computing

Students who demonstrate understanding can:

Indicator	Standard
2.IC.C.1	Compare how people live and work before and after the implementation or adoption of new computing technology.
2.IC.SLE.1	Identify safe and unsafe examples of online communications.

Practice Connection	Concept Connection
Practice 1. Foster an Inclusive Computing Culture 1. *Include the unique perspectives of others and reflect on one's own perspectives when designing and developing computational products* (2.IC.C.1). 3. *Employ self- and peer- advocacy to address bias in interactions, product design, and development methods* (2.IC.C.1). **Practice 2. Collaborating Around Computing** 1. *Cultivate working relationships with individuals possessing diverse perspectives, skills, and personalities* (2.IC.SLE.1). **Practice 7. Communicating About Computing** 3. *Articulate ideas responsibly by observing intellectual property rights and giving appropriate attribution* (2.IC.SLE.1).	**Impacts of Computing** ● IC.C: Culture - *Computing technology has positively and negatively changed the way people live and work. Computing devices can be used for entertainment and as productivity tools, and they can affect relationships and lifestyles* (2.IC.C.1). ● IC.SLE: Safety, Law, and Ethics - *People use computing technology in ways that can help or hurt themselves or others. Harmful behaviors, such as sharing private information and interacting with strangers, should be recognized and avoided* (2.IC.SLE.1).

	Connections to other NVACS at this grade level
NVACS for Science	(2.IC.C.1) – CCC: Cause and Effect (2.IC.SLE.1) – SEP: Obtaining, Evaluating, and Communicating Information
NVACS for Mathematics	None
NVACS for ELA	(2.IC.C.1) – RI.2.9, RL.2.9
NVACS for Social Studies	(2.IC.C.1) – SS.2.22, SS.2.23 (2.IC.SLE.1) – Connection to financial literacy (risk)

Networks and the Internet

Students who demonstrate understanding can:

Indicator	Standard
2.NI.C.1	Explain what passwords are and why we use them; use strong passwords to protect devices and information from unauthorized access.

Practice Connection	Concept Connection
Practice 7. Communicating About Computing *3. Articulate ideas responsibly by observing intellectual property rights and giving appropriate attribution* (2.NI.C.1).	**Networks and the Internet** ● NI.C: Cybersecurity - *Connecting devices to a network or the Internet provides great benefit, care must be taken to use authentication measures, such as strong passwords, to protect devices and information from unauthorized access* (2.NI.C.1).

	Connections to other NVACS at this grade level
NVACS for Science	(2.NI.C.1) – CCC: Cause and Effect
NVACS for Mathematics	SMP 2
NVACS for ELA	(2.NI.C.1) – W.2.2
NVACS for Social Studies	Connect to financial literacy (risk)

Algorithms and Programming

Students who demonstrate understanding can:

Indicator	Standard
3.AP.PD.1	Debug (identify and fix) errors in an algorithm or program that includes sequences and loops.
3.AP.PD.2	Take on varying roles (e.g., researcher, programmer, test developer, designer, recorder) with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.
3.AP.V.1	Create programs that use variables to store and modify data.

Practice Connection	Concept Connection
Practice 2. Collaborating Around Computing *2. Create team norms, expectations, and equitable workloads to increase efficiency and effectiveness* (3.AP.PD.2). **Practice 5. Creating Computational Artifacts** *2. Create a computational artifact for practical intent, personal expression, or to address a societal issue* (3.AP.V.1). **Practice 6. Testing and Refining Computational Artifacts** *1. Systematically test computational artifacts by considering all scenarios and using test cases* (3.AP.PD.1). *2. Identify and fix errors using a systematic process* (3.AP.PD.1).	**Algorithms and Programming** ● AP.PD: Program Development - *People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs* (3.AP.PD.1 and 3.AP.PD.2). ● AP.V: Variables - *Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data* (3.AP.V.1).

	Connections to other NVACS at this grade level
NVACS for Science	(3.AP.PD.1) – CCC: Cause and Effect, SEP: Asking Questions and Defining Problems, 3-5 ETS1-2: Generate and Compare Multiple Solutions (3.AP.PD.2) – SEP: Planning and Carrying Out Investigations (3.AP.V.1) – CCC: Cause and Effect
NVACS for Mathematics	SMP 3
NVACS for ELA	(3.AP.PD.1) – RI.3.8 (3.AP.PD.2) – W.3.5 (3.AP.V.1) – W.3.5
NVACS for Social Studies	None

Computing Systems

Students who demonstrate understanding can:

Indicator	Standard
3.CS.D.1	Describe how internal and external parts of computing devices function to form a system.

Practice Connection	Concept Connection
Practice 7. Communicating About Computing *2. Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose* (3.CS.D.1).	**Computing Systems** ● CS.D: Devices - *Computing devices may be connected to other devices or components to extend their capabilities, such as sensing and sending information. Connections can take many forms, such as physical or wireless. Together, devices and components form a system of interdependent parts that interact for a common purpose* (3.CS.D.1).

	Connections to other NVACS at this grade level
NVACS for Science	(3.CS.D.1) – CCC: Systems and System Models, CCC: Structure and Function
NVACS for Mathematics	SMP 2
NVACS for ELA	(3.CS.D.1) – RL.3.5, RI.3.2
NVACS for Social Studies	None

Data and Analysis

Students who demonstrate understanding can:

Indicator	Standard
3.DA.CVT.1	Organize and present collected data visually to highlight relationships and support a claim.

Practice Connection	Concept Connection
Practice 7. Communicating About Computing *1. Select, organize, and interpret large data sets from multiple sources to support a claim* (3.DA.CVT.1).	**Data and Analysis** ● DA.CVT: Collection, Visualization, and Transformation - People select digital tools for the collection of data based on what is being observed and how the data will be used. For example, a digital thermometer is used to measure temperature and a GPS sensor is used to track locations. *People select aspects and subsets of data to be transformed, organized, clustered, and categorized to provide different views and communicate insights gained from the data* (3.DA.CVT.1).

	Connections to other NVACS at this grade level
NVACS for Science	SEP: Obtaining, Evaluating, and Communicating data, Engaging in Argumentation from Evidence
NVACS for Mathematics	SMP 2 & 4, 3.MD.B.3
NVACS for ELA	W.3.8, W.3.2a, SL.3.2, SL.3.4, SL.3.5
NVACS for Social Studies	None

Impacts of Computing

Students who demonstrate understanding can:

Indicator	Standard
3.IC.C.1	Discuss computing technologies that have changed the world, and express how those technologies influence and are influenced by cultural practices.
3.IC.SLE.1	Use public domain or creative commons media, and refrain from copying or using material created by others without permission.

Practice Connection	Concept Connection
Practice 3. Recognizing and Defining Computational Problems 1. *Identify complex, interdisciplinary, real-world problems that can be solved computationally* (3.IC.C.1). **Practice 7. Communicating About Computing** 3. *Articulate ideas responsibly by observing intellectual property rights and giving appropriate attribution* (3.IC.SLE.1).	**Impacts of Computing** ● IC.C: Culture - *The development and modification of computing technology is driven by people's needs and wants and can affect groups differently. Computing technologies influence, and are influenced by, cultural practices* (3.IC.C.1). ● IC.SLE: Safety, Law, and Ethics - *Ethical complications arise from the opportunities provided by computing. The ease of sending and receiving copies of media on the Internet, such as video, photos, and music, creates the opportunity for unauthorized use, such as online piracy, and disregard of copyrights, such as lack of attribution* (3.IC.SLE.1).

	Connections to other NVACS at this grade level
NVACS for Science	(3.IC.C.1) – SEP: Obtaining, Communicating, and Evaluating Information, 3-ESS3-1: Influence of engineering, technology and science on society and the natural world) Engineers improve existing technologies or develop new ones to increase their benefits.
NVACS for Mathematics	None
NVACS for ELA	(3.IC.SLE.1) – W.3.8
NVACS for Social Studies	(3.IC.C.1) – SS.3.23 (3.IC.SLE.1) – Connect to financial literacy (risk)

Networks and the Internet

Students who demonstrate understanding can:

Indicator	Standard
3.NI.C.1	Discuss real-world cybersecurity problems and how personal information can be protected.

Practice Connection	Concept Connection
Practice 3. Recognizing and Defining Computational Problems 1. *Identify complex, interdisciplinary, real-world problems that can be solved computationally* (3.NI.C.1).	**Networks and the Internet** • NI.C: Cybersecurity - *Information can be protected using various security measures. These measures can be physical and/or digital* (3.NI.C.1).

	Connections to other NVACS at this grade level
NVACS for Science	(3.NI.CI.1) – SEP: Obtaining, Communicating, and Evaluating Information, CCC: Cause and Effect
NVACS for Mathematics	None
NVACS for ELA	None
NVACS for Social Studies	Connect to financial literacy (risk)

Algorithms and Programming

Students who demonstrate understanding can:

Indicator	Standard
4.AP.A.1	Test, compare, and refine multiple algorithms for the same task and determine which is the most appropriate.
4.AP.C.1	Develop programs that include sequences, events, loops, and conditionals.
4.AP.M.1	Explore how complex tasks can be decomposed into simple tasks and how simple tasks can be composed into complex tasks.
4.AP.PD.1	Test and debug (identify and fix) errors in a program or algorithm to ensure it runs as intended.

Practice Connection	Concept Connection
Practice 3. Recognizing and Defining Computational Problems *2. Decompose complex real-world problems into manageable subproblems that could integrate existing solutions or procedures* (4.AP.M.1). *3. Evaluate whether it is appropriate and feasible to solve a problem computationally* (4.AP.A.1). **Practice 5. Creating Computational Artifacts** *2. Create a computational artifact for practical intent, personal expression, or to address a societal issue* (4.AP.C.1). **Practice 6. Testing and Refining Computational Artifacts** *1. Systematically test computational artifacts by considering all scenarios and using test cases* (4.AP.A.1 and 4.AP.PD.1). *2. Identify and fix errors using a systematic process* (4.AP.A.1 and 4.AP.PD.1). *3. Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability, and accessibility* (4.AP.A.1).	**Algorithms and Programming** ● AP.A: Algorithms - *Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others* (4.AP.A.1). ● AP.C: Control - *Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions* (4.AP.C.1). ● AP.M: Modularity - *Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created* (4.AP.M.1). ● AP.PD: Program Development - *People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs* (4.AP.PD.1).

	Connections to other NVACS at this grade level
NVACS for Science	(4.AP.A.1) – 3-5 ETS1-2: Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem, 3-5 ETS1-3: Plan and carry out fair tests in which variables are controlled and failure points are considered to identify aspects of a model or prototype that can be improved, SEP: Analyzing and Interpreting Data (4.AP.C.1) – SEP: Developing and Using Models, CCC: Patterns (4.AP.M.1) – SEP: Developing and Using Models, CCC: Systems and System Models (4.AP.PD.1) – 3-5 ETS1-3: Plan and carry out fair tests in which variables are controlled and failure points are considered to identify aspects of a model or prototype that can be improved, SEP: Constructing Explanations and Designing Solutions
NVACS for Mathematics	SMP 2 & 6
NVACS for ELA	(4.AP.A.1) – RI.4.6, W.4.5 (4.AP.C.1) – RI.4.3, W.4.2 (4.AP.M.1) – RL.4.5, RI.4.9, W.4.7 (4.AP.PD.1) – W.4.5
NVACS for Social Studies	None

Computing Systems

Students who demonstrate understanding can:

Indicator	Standard
4.CS.HS.1	Model how computer hardware and software work together as a system to accomplish tasks.

Practice Connection	Concept Connection
Practice 4. Developing and Using Abstractions *4. Model phenomena and processes and simulate systems to understand and evaluate potential outcomes* (4.CS.HS.1).	**Computing Systems** ● CS.HS: Hardware and Software - *Hardware and software work together as a system to accomplish tasks, such as sending, receiving, processing, and storing units of information as bits. Bits serve as the basic unit of data in computing systems and can represent a variety of information* (4.CS.HS.1).

	Connections to other NVACS at this grade level
NVACS for Science	CCC: Systems and System Models, SEP: Developing and Using Models
NVACS for Mathematics	SMP 4
NVACS for ELA	W.4.6, RL.5.3
NVACS for Social Studies	None

Data and Analysis

Students who demonstrate understanding can:

Indicator	Standard
4.DA.IM.1	Use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate ideas.

Practice Connection	Concept Connection
Practice 7. Communicating About Computing *1. Select, organize, and interpret large data sets from multiple sources to support a claim* (4.DA.IM.1).	**Data & Analysis** ● DA.IM: Inference and Models - *The accuracy of inferences and predictions is related to how realistically data is represented. Many factors influence the accuracy of inferences and predictions, such as the amount and relevance of data collected* (4.DA.IM.1).

	Connections to other NVACS at this grade level
NVACS for Science	SEP: Obtaining, Evaluating, and Communicating Data, CCC: Cause and Effect, SEP: Using Mathematics and Computational Thinking
NVACS for Mathematics	SMP 2, 4.MD.B.4
NVACS for ELA	RI.4.5, RI.4.9
NVACS for Social Studies	None

Impacts of Computing

Students who demonstrate understanding can:

Indicator	Standard
4.IC.C.1	Compare and contrast how computing has changed society from the past to the present.

Practice Connection	Concept Connection
Practice 3. Recognizing and Defining Computational Problems 1. *Identify complex, interdisciplinary, real-world problems that can be solved computationally* (4.IC.C.1).	**Impacts of Computing** ● IC.C: Culture - *The development and modification of computing technology is driven by people's needs and wants and can affect groups differently. Computing technologies influence, and are influenced by, cultural practices* (4.IC.C.1).

	Connections to other NVACS at this grade level
NVACS for Science	CCC: Cause and Effect, SEP: Analyzing and Interpreting Data
NVACS for Mathematics	None
NVACS for ELA	RL.4.5
NVACS for Social Studies	Connect to financial literacy (decision making)

Networks and the Internet

Students who demonstrate understanding can:

Indicator	Standard
4.NI.NCO.1	Model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the internet, and reassembled at the destination.

Practice Connection	Concept Connection
Practice 4. Developing and Using Abstractions *4. Model phenomena and processes and simulate systems to understand and evaluate potential outcomes* (4.NI.NCO.1).	**Networks and the Internet** ● NI.NCO: Network, Communication, and Organization - *Information needs a physical or wireless path to travel to be sent and received, and some paths are better than others. Information is broken into smaller pieces, called packets, that are sent independently and reassembled at the destination. Routers and switches are used to properly send packets across paths to their destinations* (4.NI.NCO.1).

	Connections to other NVACS at this grade level
NVACS for Science	CCC: Systems and System Models, SEP: Developing and Using Models
NVACS for Mathematics	SMP 4
NVACS for ELA	None
NVACS for Social Studies	None

Algorithms and Programming

Students who demonstrate understanding can:

Indicator	Standard
5.AP.M.1	Demonstrate how to decompose a task of complexity into simple tasks and compose a simple task into tasks of complexity.
5.AP.M.2	Modify, incorporate, and test portions of an existing program into their own work, to develop something new or add more advanced features.
5.AP.PD.1	Use the iterative process to develop a program to express an idea or address a problem while considering others' perspectives and preferences.
5.AP.PD.2	Describe choices made during program development using code comments, presentations, and demonstrations.
5.AP.PD.3	Observe intellectual property rights and give appropriate attribution (credit) when creating or remixing programs.

Practice Connection	Concept Connection
Practice 1. Fostering an Inclusive Computing Culture 1. *Include the unique perspectives of others and reflect on one's own perspectives when designing and developing computational products* (5.AP.PD.1). **Practice 3. Recognizing and Defining Computational Problems** 2. *Decompose complex real-world problems into manageable subproblems that could integrate existing solutions or procedures* (5.AP.M.1). **Practice 5. Creating Computational Artifacts** 1. *Plan the development of a computational artifact using an iterative process that includes reflection on and modification of the plan, taking into account key features, time and resource constraints, and user expectations* (5.AP.PD.1). 2. *Create a computational artifact for practical intent, personal expression, or to address a societal issue* (5.AP.PD.3). 3. *Modify an existing artifact to improve or customize it* (5.AP.M.2). **Practice 7. Communicating About Computing** 2. *Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose* (5.AP.PD.2). 3. *Articulate ideas responsibly by observing intellectual property rights and giving appropriate attribution* (5.AP.PD.3).	**Algorithms and Programming** ● AP.M: Modularity - *Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created* (5.AP.M.1 and 5.AP.M.2). ● AP.PD: Program Development - *People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs* (5.AP.PD.1-3).

	Connections to other NVACS at this grade level
NVACS for Science	(5.AP.M.1) – CCC: Patterns, Systems and System Models (5.AP.M.2) – 3-5 ETS1-2: Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem, 3-5 ETS1-3: Plan and carry out fair tests in which variables are controlled and failure points are considered to identify aspects of a model or prototype that can be improved (5.AP.PD.1) – CCC: Cause and Effect, SEP: Obtaining, Communicating, and Evaluating Information (5.AP.PD.2) – SEP: Obtaining, Communicating, and Evaluating Information
NVACS for Mathematics	SMP 2 & 3
NVACS for ELA	(5.AP.M.1) – RL.5.5, RI.5.9, W.7 (5.AP.M.2) – SL.5.5, RI.5.9, SL.1.c,d, SL.5.5, RL.5.7 (5.AP.PD.1) – RL.5.6, W.5.5 (5.AP.PD.2) – SL.5.1a, W.5.6 (5.AP.PD.3) – W.5.8, W.5.9
NVACS for Social Studies	None

Computing Systems

Students who demonstrate understanding can:

Indicator	Standard
5.CS.T.1	Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies.

Practice Connection	Concept Connection
Practice 6. Testing and Refining Computational Artifacts *2. Identify and fix errors using a systematic process* (5.CS.T.1).	**Computing Systems** • CS.T: Troubleshooting - *Computing systems share similarities, such as the use of power, data, and memory. Common troubleshooting strategies, such as checking that power is available, checking that physical and wireless connections are working, and clearing out the working memory by restarting programs or devices, are effective for many systems* (5.CS.T.1).

	Connections to other NVACS at this grade level
NVACS for Science	3-5 ETS1-1: Define a simple design problem reflecting a need or a want that includes specified criteria for success and constraints on materials, time, or cost. 3-5 ETS1-2: Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem.
NVACS for Mathematics	SMP 1, 2, & 3
NVACS for ELA	W.5.5
NVACS for Social Studies	None

Data and Analysis

Students who demonstrate understanding can:

Indicator	Standard
5.DA.IM.1	Recognize how text, images, and sounds are represented as binary numbers in computing devices.

Practice Connection	Concept Connection
Practice 4. Developing and Using Abstractions *1. Extract common features from a set of interrelated processes or complex phenomena* (5.DA.IM.1).	**Data and Analysis** ● DA.IM: Inference and Models - *The accuracy of inferences and predictions is related to how realistically data is represented. Many factors influence the accuracy of inferences and predictions, such as the amount and relevance of data collected* (5.DA.IM.1).

	Connections to other NVACS at this grade level
NVACS for Science	SEP: Using Mathematics and Computational Thinking, SEP: Developing and Using Models
NVACS for Mathematics	SMP 7 & 8, 5.OA.B.3
NVACS for ELA	None
NVACS for Social Studies	None

Impacts of Computing

Students who demonstrate understanding can:

Indicator	Standard
5.IC.C.1	Brainstorm ways to improve the accessibility and usability of technology products for the diverse needs and wants of users.
5.IC.SI.1	Seek diverse perspectives for the purpose of improving computational artifacts.

Practice Connection	Concept Connection
Practice 1. Fostering an Inclusive Computing Culture 1. *Include the unique perspectives of others and reflect on one's own perspectives when designing and developing computational products* (5.IC.SI.1). 2. *Address the needs of diverse end users during the design process to produce artifacts with broad accessibility and usability* (5.IC.C.1).	**Impacts of Computing** ● IC.C: Culture - *The development and modification of computing technology is driven by people's needs and wants and can affect groups differently. Computing technologies influence, and are influenced by, cultural practices* (5.IC.C.1). ● IC.SI: Social Interactions - *Computing technology allows for local and global collaboration. By facilitating communication and innovation, computing influences many social institutions such as family, education, religion, and the economy* (5.IC.SI.1).

	Connections to other NVACS at this grade level
NVACS for Science	(5.IC.C.1) – SEP: Asking Questions and Defining Problems
NVACS for Mathematics	SMP 5
NVACS for ELA	(5.IC.C.1) – W.5.5, W.5.6 (5.IC.SI.1) – SL.5.1, SL.5.3
NVACS for Social Studies	(5.IC.C.1) – SS.5.19

Networks and the Internet

Students who demonstrate understanding can:

Indicator	Standard
5.NI.NCO.1	Explain the concept of network protocols.
5.NI.NCO.2	Identify the advantages and disadvantages of various network types (e.g., wire, WiFi, cellular data).

Practice Connection	Standard Concept Connection
Practice 4. Developing and Using Abstractions 1. *Extract common features from a set of interrelated processes or complex phenomena* (5.NI.NCO.2). 4. *Model phenomena and processes and simulate systems to understand and evaluate potential outcomes* (5.NI.NCO.1).	**Networks and the Internet** ● NI.NCO: Network, Communication, and Organization - *Information needs a physical or wireless path to travel to be sent and received, and some paths are better than others. Information is broken into smaller pieces, called packets, that are sent independently and reassembled at the destination. Routers and switches are used to properly send packets across paths to their destinations* (5.NI.NCO.1 and 5.NI.NCO.2).

	Connections to other NVACS at this grade level
NVACS for Science	(5.NI.NCO.2) – CCC: Cause and Effect
NVACS for Mathematics	None
NVACS for ELA	(5.NI.NCO.1) – RI.5.4 (5.NI.NCO.2) – W.5.1, RL.5.3
NVACS for Social Studies	None

Algorithms and Programming

By the end of Grade 8, students who demonstrate understanding can:

Indicator	Standard
6-8.AP.A.1	Use flowcharts and/or pseudocode to address complex problems as algorithms.
6-8.AP.V.1	Create clearly named variables that represent different data types and perform operations on their values.
6-8.AP.C.1	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
6-8.AP.M.1	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
6-8.AP.M.2	Create procedures with parameters to organize code and make it easier to reuse.
6-8.AP.PD.1	Design meaningful solutions for others, incorporating data from collaborative team members and the end user, to meet the end user’s needs.
6-8.AP.PD.2	Incorporate existing code, media, and libraries into original programs, and give attribution.
6-8.AP.PD.3	Systematically test and refine programs using a range of test cases.
6-8.AP.PD.4	Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.
6-8.AP.PD.5	Document programs (throughout the design, development, troubleshooting, and user experience phases) in order to make them easier to follow, test, and debug by others.

Practice Connection	Concept Connection
Practice 1. Fostering an Inclusive Computing Culture 1. *Include the unique perspectives of others and reflect on one's own perspectives when designing and developing computational products* (6-8.AP.PD.1). **Practice 2. Collaborating Around Computing** 2. *Create team norms, expectations, and equitable workloads to increase efficiency and effectiveness* (6-8.AP.PD.4). 3. *Solicit and incorporate feedback from, and provide constructive feedback to, team members and other stakeholders* (6-8.AP.PD.1). **Practice 3. Recognizing and Defining Computational Problems** 2. *Decompose complex real-world problems into manageable subproblems that could integrate existing solutions or procedures* (6-8.AP.M.1). **Practice 4. Developing and Using Abstractions** 1. *Extract common features from a set of interrelated processes or complex phenomena* (6-8.AP.A.1 and 6-8.AP.M.2). 2. *Evaluate existing technological functionalities and incorporate them into new designs* (6-8.AP.PD.2). 3. *Create modules and develop points of interaction that can apply to multiple situations and reduce complexity* (6-8.AP.M.2). 4. *Model phenomena and processes and simulate systems to understand and evaluate potential outcomes* (6-8.AP.A.1). **Practice 5. Creating Computational Artifacts** 1. *Plan the development of a computational artifact using an iterative process that includes reflection on and modification of the plan, taking into account key features, time and resource constraints, and user expectations* (6-8.AP.V.1 and 6-8.AP.C.1). 2. *Create a computational artifact for practical intent, personal expression, or to address a societal issue* (6-8.AP.V.1, 6-8.AP.C.1, and 6-8.AP.PD.2). **Practice 6. Testing and Refining Computational Artifacts** 1. *Systematically test computational artifacts by considering all scenarios and using test cases* (6-8.AP.PD.3). **Practice 7. Communicating About Computing** 2. *Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose* (6-8.AP.PD.5). 3. *Articulate ideas responsibly by observing intellectual property rights and giving appropriate attribution* (6-8.AP.PD.2).	**Algorithms and Programming** ● AP.A: Algorithms - *Algorithms affect how people interact with computers and the way computers respond. People design algorithms that are generalizable to many situations. Algorithms that are readable are easier to follow, test, and debug* (6-8.AP.A.1). ● AP.V: Variables - *Programmers create variables to store data values of selected types. A meaningful identifier is assigned to each variable to access and perform operations on the value by name. Variables enable the flexibility to represent different situations, process different sets of data, and produce varying outputs* (6-8.AP.V.1). ● AP.C: Control - *Programmers select and combine control structures, such as loops, event handlers, and conditionals, to create more complex program behavior* (6-8.AP.C.1). ● AP.M: Modularity - *Programs use procedures to organize code, hide implementation details, and make code easier to reuse. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability* (6-8.AP.M.1 and 6-8.AP.M.2). ● AP.PD: Program Development - *People design meaningful solutions for others by defining a problem's criteria and constraints, carefully considering the diverse needs and wants of the community, and testing whether criteria and constraints were met* (6-8.AP.PD.1-5).

	Connections to other NVACS at this grade level
NVACS for Science	(6-8.AP.A.1) – SEP 3, SEP 5, MS-PS3-4 (6-8.AP.V.1) – SEP 4, MS-PS3-1, MS-LS2-1, MS-LS4-1 (6-8.AP.C.1) – DCI ETS1, MS-LS2-5 (6-8.AP.M.1) – CCC 4, MS-LS2-2, MS-LS3-1 (6-8.AP.M.2) – CCC 4 (6-8.AP.PD.1) – MS-LS2-5
NVACS for Mathematics	(6-8.AP.A.1) – SMP2 (6-8.AP.V.1) – SMP6, 6.EE.A.2, 7.EE.B.4 (6-8.AP.C.1) – SMP1 (6-8.AP.M.1) – SMP7 (6-8.AP.M.2) – SMP1 (6-8.AP.PD.1) – SMP1 (6-8.AP.PD.2) – SMP7 (6-8.AP.PD.3) – 7.SP.C5, 7.SP.C7 (6-8.AP.PD.5) – SMP3
NVACS for ELA	(6-8.AP.A.1) – RI.6-12.5 (6-8.AP.V.1) – L.6-12.3 (6-8.AP.C.1) – SL.6-12.3 (6-8.AP.M.2) – W.6-12.1, W.6-12.2 (6-8.AP.PD.1) – SL.6-12.1 (6-8.AP.PD.5) – SL.6-12.4
NVACS for Social Studies	(6-8.AP.PD.1-5) – SS.WGGS.17

Computing Systems

By the end of Grade 8, students who demonstrate understanding can:

Indicator	Standard
6-8.CS.D.1	Recommend improvements to the design of computing devices based on an analysis of how users interact with the devices, noting that advantages may have disadvantages and unintended consequences.
6-8.CS.HS.1	Design and evaluate projects that combine hardware and software components to collect and exchange data.
6-8.CS.T.1	Systematically identify and fix problems with computing devices and their components.

Practice Connection	Concept Connection
Practice 3. Recognizing and Defining Computational Problems 3. *Evaluate whether it is appropriate and feasible to solve a problem computationally* (6-8.CS.D.1). **Practice 5. Creating Computational Artifacts** 1. *Plan the development of a computational artifact using an iterative process that includes reflection on and modification of the plan, taking into account key features, time and resource constraints, and user expectations* (6-8.CS.HS.1). **Practice 6. Testing and Refining Computational Artifacts** 2. *Identify and fix errors using a systematic process* (6-8.CS.T.1).	**Computing Systems** ● CS.D: Devices - *The interaction between humans and computing devices presents advantages, disadvantages, and unintended consequences. The study of human-computer interaction can improve the design of devices and extend the abilities of humans* (6-8.CS.D.1). ● CS.HS: Hardware and Software - *Hardware and software determine a computing system's capability to store and process information. The design or selection of a computing system involves multiple considerations and potential tradeoffs, such as functionality, cost, size, speed, accessibility, and aesthetics* (6-8.CS.HS.1). ● CS.T: Troubleshooting - *Comprehensive troubleshooting requires knowledge of how computing devices and components work and interact. A systematic process will identify the source of a problem whether within a device or in a larger system of connected devices* (6-8.CS.T.1).

	Connections to other NVACS at this grade level
NVACS for Science	None
NVACS for Mathematics	(6-8.CS.D.1) – SMP3 (6-8.CS.HS.1) – SMP2, SMP3 (6-8.CS.T.1) – SMP8
NVACS for ELA	(6-8.CS.T.1) – W.6-12.5
NVACS for Social Studies	None

Data and Analysis

By the end of Grade 8, students who demonstrate understanding can:

Indicator	Standard
6-8.DA.S.1	Model encoding schema used by software tools to access data, stored as bits, into forms more easily understood by people (e.g., encoding schema include binary and ASCII).
6-8.DA.CVT.1	Collect data using computational tools and transform the data to make it more meaningful and useful.
6-8.DA.IM.1	Refine computational models based on the reliability and validity of the data they generate.

Practice Connection	Concept Connection
Practice 4. Developing and Using Abstractions *1. Extract common features from a set of interrelated processes or complex phenomena* (6-8.DA.S.1). *4. Model phenomena and processes and simulate systems to understand and evaluate potential outcomes* (6-8.DA.S.1 and 6-8.DA.IM.1). **Practice 5. Creating Computational Artifacts** *3. Modify an existing artifact to improve or customize it* (6-8.DA.IM.1). **Practice 6. Testing and Refining Computational Artifacts** *3. Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability, and accessibility* (6-8.DA.CVT.1).	**Data and Analysis** ● DA.S: Storage - *Applications store data as a representation. Representations occur at multiple levels, from the arrangement of information into organized formats (such as tables in software) to the physical storage of bits. The software tools used to access information translate the low-level representation of bits into a form understandable by people* (6-8.DA.S.1). ● DA.CVT: Collection, Visualization and Transformation - *People design algorithms and tools to automate the collection of data by computers. When data collection is automated, data is sampled and converted into a form that a computer can process. For example, data from an analog sensor must be converted into a digital form. The method used to automate data collection is influenced by the availability of tools and the intended use of the data. Data can be transformed to remove errors, highlight or expose relationships, and/or make it easier for computers to process* (6-8.DA.CVT.1). ● DA.IM: Inference and Models - *Computer models can be used to simulate events, examine theories and inferences, or make predictions with either few or millions of data points. Computer models are abstractions that represent phenomena and use data and algorithms to emphasize key features and relationships within a system. As more data is automatically collected, models can be refined* (6-8.DA.IM.1).

	Connections to other NVACS at this grade level
NVACS for Science	(6-8.DA.CVT.1) – MS-LS4-6 (6-8.DA.IM.1) – MS-LS4-3, MS-LS3-1 & 3-2
NVACS for Mathematics	(6-8.DA.S.1) – SMP4 (6-8.DA.CVT.1) – SMP5, 7.SP.C (6-8.DA.IM.1) – SMP1, 7.SP.C
NVACS for ELA	(6-8.DA.CVT.1) – W.6-12.9 (6-8.DA.IM.1) – W.6-12.8
NVACS for Social Studies	None

Impacts of Computing

By the end of Grade 8, students who demonstrate understanding can:

Indicator	Standard
6-8.IC.C.1	Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.
6-8.IC.C.2	Discuss and evaluate issues of bias and accessibility in the design of existing technologies.
6-8.IC.SI.1	Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.
6-8.IC.SLE.1	Identify risks associated with sharing information digitally (e.g., phishing, identity theft, hacking).
6-8.IC.SLE.2	Evaluate how legal and ethical issues shape computing practices.

Practice Connection	Concept Connection
Practice 1. Fostering an Inclusive Computing Culture 2. *Address the needs of diverse end users during the design process to produce artifacts with broad accessibility and usability* (6-8.IC.C.2). **Practice 2. Collaborating Around Computing** 4. *Evaluate and select technological tools that can be used to collaborate on a project* (6-8.IC.SI.1). **Practice 5. Creating Computational Artifacts** 2. *Create a computational artifact for practical intent, personal expression, or to address a societal issue* (6-8.IC.SI.1). **Practice 7. Communicating About Computing** 2. *Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose* (6-8.IC.C.1 and 6-8.IC.SLE.1). 3. *Articulate ideas responsibly by observing intellectual property rights and giving appropriate attribution* (6-8.IC.SLE.2).	**Impacts of Computing** ● IC.C: Culture - *Advancements in computing technology change people's everyday activities. Society is faced with tradeoffs due to the increasing globalization and automation that computing brings* (6-8.IC.C.1 and 6-8.IC.C.2). ● IC.SI: Social Interactions - *People can organize and engage around issues and topics of interest through various communication platforms enabled by computing, such as social networks and media outlets. These interactions allow issues to be examined using multiple viewpoints from a diverse audience* (6-8.IC.SI.1). ● IC.SLE: Safety, Law and Ethics - *There are tradeoffs between allowing information to be public and keeping information private and secure. People can be tricked into revealing personal information when more public information is available about them online* (6-8.IC.SLE.1 and 6-8.IC.SLE.2).

	Connections to other NVACS at this grade level
NVACS for Science	None
NVACS for Mathematics	None
NVACS for ELA	(6-8.IC.SI.1) – SL.6-12.1
NVACS for Social Studies	None

Networks and the Internet

By the end of Grade 8, students who demonstrate understanding can:

Indicator	Standard
6-8.NI.NCO.1	Compare and contrast modeled protocols used in transmitting data across networks and the Internet.
6-8.NI.C.1	Explain how physical and digital security measures protect electronic information.
6-8.NI.C.2	Apply multiple methods of encryption to model the secure transmission of information.

Practice Connection	Concept Connection
Practice 4. Developing and Using Abstractions *4. Model phenomena and processes and simulate systems to understand and evaluate potential outcomes* (6-8.NI.NCO.1 and 6-8.NI.C.2). **Practice 7. Communicating About Computing** *2. Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose* (6-8.NI.C.1).	**Networks and the Internet** • NI.NCO: Network, Communication, and Organization - *Computers send and receive information based on a set of rules called protocols. Protocols define how messages between computers are structured and sent. Considerations of security, speed, and reliability are used to determine the best path to send and receive data* (6-8.NI.NCO.1). • NI.C: Cybersecurity - *The information sent and received across networks can be protected from unauthorized access and modification in a variety of ways, such as encryption to maintain its confidentiality and restricted access to maintain its integrity. Security measures to safeguard online information proactively address the threat of breaches to personal and private data* (6-8.NI.C.1 and 6-8.NI.C.2).

	Connections to other NVACS at this grade level
NVACS for Science	(6-8.NI.NCO.1) – MS-PS4-3, MS-ETS1-2, SEP2 (6-8.NI.C.1) – SEP6
NVACS for Mathematics	None
NVACS for ELA	None
NVACS for Social Studies	(6-8.NI.C.1-2) – Connect to financial literacy (risk)

Algorithms and Programming

By the end of Grade 12, students who demonstrate understanding can:

Indicator	Standard
9-12.AP.A.1	Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
9-12.AP.V.1	Demonstrate the use of both linked lists and arrays to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.
9-12.AP.C.1	Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.
9-12.AP.C.2	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.
9-12.AP.M.1	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
9-12.AP.M.2	Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
9-12.AP.PD.1	Systematically design and develop programs for broad audiences by incorporating feedback from users.
9-12.AP.PD.2	Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries.
9-12.AP.PD.3	Evaluate and refine computational artifacts to make them more usable by all and accessible to people with disabilities.
9-12.AP.PD.4	Design and develop computational artifacts working in team roles using collaborative tools.
9-12.AP.PD.5	Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

Practice Connection	Concept Connection
Practice 2. Collaborating Around Computing *4. Evaluate and select technological tools that can be used to collaborate on a project* (9-12.AP.PD.4). **Practice 3. Recognizing and Defining Computational Problems** *2. Decompose complex real-world problems into manageable subproblems that could integrate existing solutions or procedures* (9-12.AP.C.1 and 9-12.AP.M.1). **Practice 4. Developing and Using Abstractions** *1. Extract common features from a set of interrelated processes or complex phenomena* (9-12.AP.V.1). **Practice 5. Creating Computational Artifacts** *1. Plan the development of a computational artifact using an iterative process that includes reflection on and modification of the plan, taking into account key features, time and resource constraints, and user expectations* (9-12.AP.PD.1). *2. Create a computational artifact for practical intent, personal expression, or to address a societal issue* (9-12.AP.A.1, 9-12.AP.C.1-2, and 9-12.AP.M.2). **Practice 6. Testing and Refining Computational Artifacts** *3. Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability, and accessibility* (9-12.AP.PD.3). **Practice 7. Communicating About Computing** *2. Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose* (9-12.AP.PD.5). *3. Articulate ideas responsibly by observing intellectual property rights and giving appropriate attribution* (9-12.AP.PD.2).	**Algorithms and Programming** ● AP.A: Algorithms - *People evaluate and select algorithms based on performance, reusability, and ease of implementation. Knowledge of common algorithms improves how people develop software, secure data, and store information* (9-12.AP.A.1). ● AP.V: Variables - *Data structures are used to manage program complexity. Programmers choose data structures based on functionality, storage, and performance tradeoffs* (9-12.AP.V.1). ● AP.C: Control - *Programmers consider tradeoffs related to implementation, readability, and program performance when selecting and combining control structures* (9-12.AP.C.1 and 9-12.AP.C.2). ● AP.M: Modularity - *Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose. These modules can be procedures within a program; combinations of data and procedures; or independent, but interrelated, programs. Modules allow for better management of complex tasks* (9-12.AP.M.1 and 9-12.AP.M.2). ● AP.PD: Program Development - *Diverse teams can develop programs with a broad impact through careful review and by drawing on the strengths of members in different roles. Design decisions often involve tradeoffs. The development of complex programs is aided by resources such as libraries and tools to edit and manage parts of the program. Systematic analysis is critical for identifying the effects of lingering bugs* (9-12.AP.PD.1-5).

	Connections to other NVACS at this grade level
NVACS for Science	(9-12.AP.A.1) – HS-ETS1-2 (9-12.AP.C.1) – HS-ETS1-3, SEP7 (9-12.AP.M.1) – HS-ETS1-2 (9-12.AP.M.2) – HS-ETS1-4 (9-12.AP.PD.1) – CCC4, HS-ETS1-3 (9-12.AP.PD.3) – SEP8 (9-12.AP.PD.4) – SEP5, SEP8 (9-12.AP.PD.5) – SEP8

NVACS for Mathematics	(9-12.AP.A.1) – SMP 1, 2, 4, 7, & 8 (9-12.AP.V.1) – SMP 4, 5, & 7 (9-12.AP.C.1) – SMP 2 & 3 (9-12.AP.C.2) – SMP 4 (9-12.AP.M.1) – SMP 2, 6, & 7 (9-12.AP.PD.1) – SMP 1-4 (9-12.AP.PD.3) – SMP 3 & 6 (9-12.AP.PD.4) – SMP 4 & 5 (9-12.AP.PD.5) – SMP 1, 2, 4, & 5
NVACS for ELA	(9-12.AP.A.1) – ELA W.9-10.3 (9-12.AP.V.1) – ELA L.9-10.1b (9-12.AP.C.1) – ELA W.9-10.1 (9-12.AP.M.1) – ELA RI.9-10.3 (9-12.AP.PD.1) – ELA W.9-10.5 (9-12.AP.PD.2) – ELA RI.9-10.7 (9-12.AP.PD.3) – ELA RI.9-10.8 (9-12.AP.PD.4) – ELA W.9-10.3a (9-12.AP.PD.5) – ELA W.9-10.2a
NVACS for Social Studies	None

Computing Systems

By the end of Grade 12, students who demonstrate understanding can:

Indicator	Standard
9-12.CS.D.1	Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.
9-12.CS.HS.1	Compare levels of abstraction and interactions between application software, system software, and hardware layers.
9-12.CS.T.1	Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.

Practice Connection	Concept Connection
Practice 4. Developing and Using Abstractions 1. *Extract common features from a set of interrelated processes or complex phenomena* (9-12.CS.D.1 and 9-12.CS.HS.1). **Practice 6. Testing and Refining Computational Artifacts** 2. *Identify and fix errors using a systematic process* (9-12.CS.T.1).	**Computing Systems** • CS.D: Devices - *Computing devices are often integrated with other systems, including biological, mechanical, and social systems. These devices can share data with one another. The usability, dependability, security, and accessibility of these devices, and the systems they are integrated with, are important considerations in their design as they evolve* (9-12.CS.D.1). • CS.HS: Hardware and Software - *Levels of interaction exist between the hardware, software, and user of a computing system. The most common levels of software that a user interacts with include system software and applications. System software controls the flow of information between hardware components used for input, output, storage, and processing* (9-12.CS.HS.1). • CS.T: Troubleshooting - *Troubleshooting complex problems involves the use of multiple sources when researching, evaluating, and implementing potential solutions. Troubleshooting also relies on experience, such as when people recognize that a problem is similar to one they have seen before or adapt solutions that have worked in the past* (9-12.CS.T.1).

	Connections to other NVACS at this grade level
NVACS for Science	(9-12.CS.D.1) – SEP2, SEP8 (9-12.CS.HS.1) – SEP2, SEP8 (9-12.CS.T.1) – SEP2, SEP8
NVACS for Mathematics	(9-12.CS.CS.T.1) – SMP3
NVACS for ELA	(9-12.CS.D.1) – ELA RI.9-10.3 (9-12.CS.HS.1) – ELA RI.6.9 (9-12.CS.T.1) – ELA W.9-10.2b
NVACS for Social Studies	None

Data and Analysis

By the end of Grade 12, students who demonstrate understanding can:

Indicator	Standard
9-12.DA.S.1	Translate between different bit representations of real-world phenomena, such as characters, numbers, and images (e.g., convert hexadecimal colors to decimal percentages, ASCII/Unicode representation).
9-12.DA.S.2	Evaluate the tradeoffs in how data elements are organized and where data is stored.
9-12.DA.CVT.1	Create interactive data visualizations or alternative representations using software tools to help others better understand real-world phenomena.
9-12.DA.IM.1	Create computational models that represent the relationships among different elements of data collected from a phenomenon, process, or model.

Practice Connection	Concept Connection
Practice 3. Recognizing and Defining Computational Problems 3. *Evaluate whether it is appropriate and feasible to solve a problem computationally* (9-12.DA.S.2). **Practice 4. Developing and Using Abstractions** 1. *Extract common features from a set of interrelated processes or complex phenomena* (9-12.DA.S.1). 4. *Model phenomena and processes and simulate systems to understand and evaluate potential outcomes* (9-12.DA.CVT.1 and 9-12.DA.IM.1).	**Data and Analysis** ● DA.S: Storage - *Data can be composed of multiple data elements that relate to one another. For example, population data may contain information about age, gender, and height. People make choices about how data elements are organized and where data is stored. These choices affect cost, speed, reliability, accessibility, privacy, and integrity* (9-12.DA.S.1 and 9-12.DA.S.2). ● DA.CVT: Collection, Visualization and Transformation - *Data is constantly collected or generated through automated processes that are not always evident, raising privacy concerns. The different collection methods and tools that are used influence the amount and quality of the data that is observed and recorded. People transform, generalize, simplify, and present large data sets in different ways to influence how other people interpret and understand the underlying information. Examples include visualization, aggregation, rearrangement, and application of mathematical operations* (9-12.DA.CVT.1). ● DA.IM: Inference and Models - *The accuracy of predictions or inferences depends upon the limitations of the computer model and the data the model is built upon. The amount, quality, and diversity of data and the features chosen can affect the quality of a model and ability to understand a system. Predictions or inferences are tested to validate models.* (9-12.DA.IM.1).

	Connections to other NVACS at this grade level
NVACS for Science	(9-12.DA.S.1) – SEP2, SEP4 (9-12.DA.S.2) – HS-ETS1-3, HS-LS2-7, HS-ETS1-2 & 1-3, HS-LS4-6, HS-ETS1-2 & 1-3, SEP8 (9-12.DA.CVT.1) – SEP2, HS-LS2-4, HS-LS2-1 & 2-2, HS-LS3-3, HS-LS4-3, HS-LS4-1, HS-LS4-6 (9-12.DA.IM.1) – SEP2, SEP4, SEP5, SEP8
NVACS for Mathematics	(9-12.DA.S.1) – SMP 4-7 (9-12.DA.CVT.1) – SMP 4 & 5 (9-12.DA.IM.1) – SMP 4 & 5
NVACS for ELA	(9-12.DA.S.2) – ELA RI.9-10.8 (9-12.DA.CVT.1) – ELA W.9-10.2a (9-12.DA.IM.1) – ELA W.9-10.2a
NVACS for Social Studies	None

Impacts of Computing

By the end of Grade 12, students who demonstrate understanding can:

Indicator	Standard
9-12.IC.C.1	Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
9-12.IC.C.2	Test and refine computational artifacts to reduce bias and equity deficits.
9-12.IC.C.3	Demonstrate ways a given algorithm applies to problems across disciplines.
9-12.IC.C.4	Explain the potential impacts of artificial intelligence on society.
9-12.IC.SI.1	Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields.
9-12.IC.SLE.1	Explain the beneficial and harmful effects that intellectual property laws can have on innovation.
9-12.IC.SLE.2	Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users.
9-12.IC.SLE.3	Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.

Practice Connection	Concept Connection
Practice 1. Fostering an Inclusive Computing Culture 1. *Include the unique perspectives of others and reflect on one's own perspectives when designing and developing computational products* (9-12.IC.C.4). 2. *Address the needs of diverse end users during the design process to produce artifacts with broad accessibility and usability* (9-12.IC.C.1 and 9-12.IC.C.2). **Practice 2. Collaborating Around Computing** 4. *Evaluate and select technological tools that can be used to collaborate on a project* (9-12.IC.SI.1). **Practice 3. Recognizing and Defining Computational Problems** 1. *Identify complex, interdisciplinary, real-world problems that can be solved computationally* (9-12.IC.C.3). **Practice 7. Communicating About Computing** 2. *Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose* (9-12.IC.SLE.2). 3. *Articulate ideas responsibly by observing intellectual property rights and giving appropriate attribution* (9-12.IC.SLE.1 and 9-12.IC.SLE.3).	**Impacts of Computing** ● IC.C: Culture - *The design and use of computing technologies and artifacts can improve, worsen, or maintain inequitable access to information and opportunities* (9-12.IC.C.1-4). ● IC.SI: Social Interactions - *Many aspects of society, especially careers, have been affected by the degree of communication afforded by computing. The increased connectivity between people in different cultures and in different career fields has changed the nature and content of many careers* (9-12.IC.SI.1). ● IC.SLE: Safety, Law and Ethics - *Laws govern many aspects of computing, such as privacy, data, property, information, and identity. These laws can have beneficial and harmful effects, such as expediting or delaying advancements in computing and protecting or infringing upon people's rights. International differences in laws and ethics have implications for computing* (9-12.IC.SLE.1-3).

	Connections to other NVACS at this grade level
NVACS for Science	(9-12.IC.C.1) – ETS2.8 (9-12.IC.C.2) – SEP2, SEP3, SEP8 (9-12.IC.C.4) – SEP6, ETS2.8 (9-12.IC.SI.1) – SEP8 (9-12.IC.SLE.1) – SEP6, SEP 7, SEP8, HS-ETS1-3 (9-12.IC.SLE.3) – HS-ETS1-3, SEP7
NVACS for Mathematics	(9-12.IC.C.2) – SMP 4 (9-12.IC.C.3) – SMP 7 (9-12.IC.SI.1) – SMP 5
NVACS for ELA	(9-12.IC.C.1) – ELA RI.9-10.8 (9-12.IC.C.4) – ELA W.9-10.2 (9-12.IC.SI.1) – ELA SL.9-10.1b (9-12.IC.SLE.1) – ELA W.9-10.2 (9-12.IC.SLE.2) – ELA W.9-10.2 (9-12.IC.SLE.3) – ELA RI.9-10.8
NVACS for Social Studies	(9-12.IC.SLE.1-3) – Connect to financial literacy (risk)

Networks and the Internet

By the end of Grade 12, students who demonstrate understanding can:

Indicator	Standard
9-12.NI.NCO.1	Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.
9-12.NI.C.1	Give examples to illustrate how sensitive data can be affected by malware and other attacks.
9-12.NI.C.2	Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts.
9-12.NI.C.3	Compare various security measures, considering tradeoffs between the usability and security of a computing system.
9-12.NI.C.4	Explain tradeoffs when selecting and implementing cybersecurity recommendations.

Practice Connection	Concept Connection
Practice 3. Recognizing and Defining Computational Problems *3. Evaluate whether it is appropriate and feasible to solve a problem computationally* (9-12.NI.C.2). **Practice 4. Developing and Using Abstractions** *1. Extract common features from a set of interrelated processes or complex phenomena* (9-12.NI.NCO.1). **Practice 6. Testing and Refining Computational Artifacts** *3. Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability, and accessibility* (9-12.NI.C.3). **Practice 7. Communicating About Computing** *2. Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose* (9-12.NI.C.1 and 9-12.NI.C.4).	**Networks and the Internet** ● NI.NCO: Network, Communication, and Organization - *Network topology is determined, in part, by how many devices can be supported. Each device is assigned an address that uniquely identifies it on the network. The scalability and reliability of the Internet are enabled by the hierarchy and redundancy in networks* (9-12.NI.NCO.1). ● NI.C: Cybersecurity - *Network security depends on a combination of hardware, software, and practices that control access to data and systems. The needs of users and the sensitivity of data determine the level of security implemented* (9-12.NI.C.1-4).

	Connections to other NVACS at this grade level
NVACS for Science	(9-12.NI.NCO.1) – SEP8, HS-ETS1-1 (9-12.NI.C.1) – SEP2 (9-12.NI.C.2) – HS-ETS1-1, HS-ETS1-2 (9-12.NI.C.3) – HS-ETS1-3 (9-12.NI.C.4) – HS-ETS1-3
NVACS for Mathematics	None
NVACS for ELA	(9-12.NI.NCO.1) – ELA RI.9-10.8 (9-12.NI.C.1) – ELA RI.9-10.1 (9-12.NI.C.3) – ELA RI.6.9 (9-12.NI.C.4) – ELA W.9-10.2d
NVACS for Social Studies	(9-12.NI.C.1-4) – Connect to financial literacy (risk)

Algorithms and Programming

** These Advanced Computer Science Standards are considered to be higher level concepts that may be used by your advanced students, incorporated into upper level courses, and/or used in Career and Technical Education (CTE) programs. These are in addition to the 9-12 Computer Science Standards and are optional.*

By the end of Grade 12, students who demonstrate understanding can:

Indicator	Standard
A9-12.AP.A.1	Describe how artificial intelligence drives many software and physical systems.
A9-12.AP.A.2	Implement an artificial intelligence algorithm to play a game against a human opponent or solve a problem.
A9-12.AP.A.3	Use and adapt classic algorithms to solve computational problems.
A9-12.AP.A.4	Evaluate algorithms in terms of their efficiency, correctness, and clarity.
A9-12.AP.V.1	Compare and contrast fundamental data structures and their uses.
A9-12.AP.C.1	Illustrate the flow of execution of a recursive algorithm.
A9-12.AP.M.1	Construct solutions to problems using student-created components, such as procedures, modules and/or objects.
A9-12.AP.M.2	Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.
A9-12.AP.M.3	Demonstrate code reuse by creating programming solutions using libraries and APIs.
A9-12.AP.PD.1	Plan and develop programs for broad audiences using a software life cycle process.
A9-12.AP.PD.2	Explain security issues that might lead to compromised computer programs.
A9-12.AP.PD.3	Develop programs for multiple computing platforms.
A9-12.AP.PD.4	Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project.
A9-12.AP.PD.5	Develop and use a series of test cases to verify that a program performs according to its design specifications.
A9-12.AP.PD.6	Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality).
A9-12.AP.PD.7	Evaluate key qualities of a program through a process such as a code review.

A9-12.AP.PD.8	Compare multiple programming languages and discuss how their features make them suitable for solving different types of problems.

Practice Connection	Concept Connection
Practice 2. Collaborating Around Computing 4. *Evaluate and select technological tools that can be used to collaborate on a project* (A9-12.AP.PD.4). **Practice 3. Recognizing and Defining Computational Problems** 2. *Decompose complex real-world problems into manageable subproblems that could integrate existing solutions or procedures* (A9-12.AP.C.1). **Practice 4. Developing and Using Abstractions** 1. *Extract common features from a set of interrelated processes or complex phenomena* (A9-12.AP.M.2). 2. *Evaluate existing technological functionalities and incorporate them into new designs* (A9-12.AP.A.3-4 and A9-12.AP.V.1). **Practice 5. Creating Computational Artifacts** 1. *Plan the development of a computational artifact using an iterative process that includes reflection on and modification of the plan, taking into account key features, time and resource constraints, and user expectations* (A9-12.AP.PD.1). 2. *Create a computational artifact for practical intent, personal expression, or to address a societal issue* (A9-12.AP.M.1 and A9-12.AP.PD.3). 3. *Modify an existing artifact to improve or customize it* (A9-12.AP.A.2, A9-12.AP.M.3, and A9-12.AP.PD.6). **Practice 6. Testing and Refining Computational Artifacts** 1. *Systematically test computational artifacts by considering all scenarios and using test cases* (A9-12.AP.PD.5). 3. *Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability, and accessibility* (A9-12.AP.PD.7). **Practice 7. Communicating About Computing** 2. *Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose* (A9-12.AP.A.1, A9-12.AP.PD.2 and A9-12.AP.PD.8).	**Algorithms and Programming** ● AP.A: Algorithms - *People evaluate and select algorithms based on performance, reusability, and ease of implementation. Knowledge of common algorithms improves how people develop software, secure data, and store information* (A9-12.AP.A.1-4). ● AP.V: Variables - *Data structures are used to manage program complexity. Programmers choose data structures based on functionality, storage, and performance tradeoffs* (A9-12.AP.V.1). ● AP.C: Control - *Programmers consider tradeoffs related to implementation, readability, and program performance when selecting and combining control structures* (A9-12.AP.C.1). ● AP.M: Modularity - *Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose. These modules can be procedures within a program; combinations of data and procedures; or independent, but interrelated, programs. Modules allow for better management of complex tasks* (A9-12.AP.M.1-3). ● AP.PD: Program Development - *Diverse teams can develop programs with a broad impact through careful review and by drawing on the strengths of members in different roles. Design decisions often involve tradeoffs. The development of complex programs is aided by resources such as libraries and tools to edit and manage parts of the program. Systematic analysis is critical for identifying the effects of lingering bugs* (A9-12.AP.PD.1-8).

	Connections to other NVACS at this grade level
NVACS for Science	(A9-12.AP.A.1) – HS-ETS1-2 (A9-12.AP.C.1) – HS-ETS1-3, SEP7 (A9-12.AP.M.1) – HS-ETS1-2 (A9-12.AP.M.2) – HS-ETS1-4 (A9-12.AP.PD.1) – CCC4, HS-ETS1-3 (A9-12.AP.PD.3) – SEP8 (A9-12.AP.PD.4) – SEP5, SEP8 (A9-12.AP.PD.5) – SEP8
NVACS for Mathematics	(A9-12.AP.A.1) – SMP 1, 2, 4, 7, & 8 (A9-12.AP.V.1) – SMP 4, 5, & 7 (A9-12.AP.C.1) – SMP 2 & 3 (A9-12.AP.C.2) – SMP 4 (A9-12.AP.M.1) – SMP 2, 6, & 7 (A9-12.AP.PD.1) – SMP 1-4 (A9-12.AP.PD.3) – SMP 3 & 6 (A9-12.AP.PD.4) – SMP 4 & 5 (A9-12.AP.PD.5) – SMP 1, 2, 4, & 5
NVACS for ELA	(A9-12.AP.A.1) – W.9-11.2b (A9-12.AP.A.4) – RI.9-10.8 (A9-12.AP.V.1) – W.7.2a or RI.6.9 (A9-12.AP.M.1) – W.11-12.2a (A9-12.AP.M.2) – RI.11-12.3 (A9-12.AP.PD.1) – W.11-12.2b (A9-12.AP.PD.2) – W.11-12.2 (A9-12.AP.PD.7) – RI.9-10.8 (A9-12.AP.PD.8) – W.7.2a or RI.6.9
NVACS for Social Studies	None

Grades 9-12 Advanced* Computer Science Standards

Computing Systems

** These Advanced Computer Science Standards are considered to be higher level concepts that may be used by your advanced students, incorporated into upper level courses, and/or used in Career and Technical Education (CTE) programs. These are in addition to the 9-12 Computer Science Standards and are optional.*

By the end of Grade 12, students who demonstrate understanding can:

Indicator	Standard
A9-12.CS.HS.1	Categorize the roles of operating system software.
A9-12.CS.T.1	Illustrate ways computing systems implement logic, input, and output through hardware components.

Practice Connection	Concept Connection
Practice 7. Communicating About Computing *2. Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose* (A9-12.CS.HS.1 and A9-12.CS.T.1).	**Computing Systems** • CS.HS: Hardware and Software - *Levels of interaction exist between the hardware, software, and user of a computing system. The most common levels of software that a user interacts with include system software and applications. System software controls the flow of information between hardware components used for input, output, storage, and processing* (A9-12.CS.HS.1). • CS.T: Troubleshooting - *Troubleshooting complex problems involves the use of multiple sources when researching, evaluating, and implementing potential solutions. Troubleshooting also relies on experience, such as when people recognize that a problem is similar to one they have seen before or adapt solutions that have worked in the past* (A9-12.CS.T.1).

	Connections to other NVACS at this grade level
NVACS for Science	(A9-12.CS.D.1) – SEP2, SEP8 (A9-12.CS.HS.1) – SEP2, SEP8 (A9-12.CS.T.1) – SEP2, SEP8
NVACS for Mathematics	(A9-12.CS.CS.T.1) – SMP3
NVACS for ELA	None
NVACS for Social Studies	None

Data and Analysis

** These Advanced Computer Science Standards are considered to be higher level concepts that may be used by your advanced students, incorporated into upper level courses, and/or used in Career and Technical Education (CTE) programs. These are in addition to the 9-12 Computer Science Standards and are optional.*

By the end of Grade 12, students who demonstrate understanding can:

Indicator	Standard
A9-12.DA.CVT.1	Use data analysis tools and techniques to identify patterns in data representing complex systems.
A9-12.DA.CVT.2	Select data collection tools and techniques to generate data sets that support a claim or communicate information.
A9-12.DA.IM.1	Evaluate the ability of models and simulations to test and support the refinement of hypotheses.

Practice Connection	Concept Connection
Practice 4. Developing and Using Abstractions 1. *Extract common features from a set of interrelated processes or complex phenomena* (A9-12.DA.CVT.1). 4. *Model phenomena and processes and simulate systems to understand and evaluate potential outcomes* (A9-12.DA.IM.1). **Practice 7. Communicating About Computing** 2. *Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose* (A9-12.DA.CVT.2).	**Data and Analysis** • DA.CVT: Collection, Visualization and Transformation - *Data is constantly collected or generated through automated processes that are not always evident, raising privacy concerns. The different collection methods and tools that are used influence the amount and quality of the data that is observed and recorded. People transform, generalize, simplify, and present large data sets in different ways to influence how other people interpret and understand the underlying information. Examples include visualization, aggregation, rearrangement, and application of mathematical operations* (A9-12.DA.CVT.1-2). • DA.IM: Inference and Models - *The accuracy of predictions or inferences depends upon the limitations of the computer model and the data the model is built upon. The amount, quality, and diversity of data and the features chosen can affect the quality of a model and ability to understand a system. Predictions or inferences are tested to validate models* (A9-12.DA.IM.1).

	Connections to other NVACS at this grade level
NVACS for Science	(A9-12.DA.S.1) – SEP2, SEP4 (A9-12.DA.S.2) – HS-ETS1-3, HS-LS2-7, HS-ETS1-2 & 1-3, HS-LS4-6, HS-ETS1-2 & 1-3, SEP8 (A9-12.DA.CVT.1) – SEP2, HS-LS2-4, HS-LS2-1 & 2-2, HS-LS3-3, HS-LS4-3, HS-LS4-1, HS-LS4-6 (A9-12.DA.IM.1) – SEP2, SEP4, SEP5, SEP8
NVACS for Mathematics	(A9-12.DA.S.1) – SMP 4-7 (A9-12.DA.CVT.1) – SMP 4 & 5 (A9-12.DA.IM.1) – SMP 4 & 5
NVACS for ELA	(A9-12.DA.CVT.1) – RI.9-10.5 (A9-12.DA.CVT.2) – W.8.1b (A9-12.DA.IM.1) – RI.6.8
NVACS for Social Studies	None

Impacts of Computing

** These Advanced Computer Science Standards are considered to be higher level concepts that may be used by your advanced students, incorporated into upper level courses, and/or used in Career and Technical Education (CTE) programs. These are in addition to the 9-12 Computer Science Standards and are optional.*

By the end of Grade 12, students who demonstrate understanding can:

Indicator	Standard
A9-12.IC.C.1	Evaluate computational artifacts to maximize their beneficial effects and minimize harmful effects on society.
A9-12.IC.C.2	Evaluate the impact of equity, access, and influence on the distribution of computing resources in a global society.
A9-12.IC.C.3	Predict how computational innovations that have revolutionized aspects of our culture might evolve.
A9-12.IC.SLE.1	Debate laws and regulations that impact the development and use of software.

Practice Connection	Concept Connection
Practice 1. Fostering an Inclusive Computing Culture *2. Address the needs of diverse end users during the design process to produce artifacts with broad accessibility and usability* (A9-12.IC.C.1 and A9-12.IC.C.2). **Practice 3. Recognizing and Defining Computational Problems** *3. Evaluate whether it is appropriate and feasible to solve a problem computationally* (A9-12.IC.SLE.1). **Practice 6. Testing and Refining Computational Artifacts** *1. Systematically test computational artifacts by considering all scenarios and using test cases* (A9-12.IC.C.1). **Practice 7. Communicating About Computing** *2. Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose* (A9-12.IC.C.3). *3. Articulate ideas responsibly by observing intellectual property rights and giving appropriate attribution* (A9-12.IC.SLE.1).	**Impacts of Computing** • IC.C: Culture - *The design and use of computing technologies and artifacts can improve, worsen, or maintain inequitable access to information and opportunities* (A9-12.IC.C.1-3). • IC.SLE: Safety, Law and Ethics - *Laws govern many aspects of computing, such as privacy, data, property, information, and identity. These laws can have beneficial and harmful effects, such as expediting or delaying advancements in computing and protecting or infringing upon people's rights. International differences in laws and ethics have implications for computing* (A9-12.IC.SLE.1).

	Connections to other NVACS at this grade level
NVACS for Science	(A9-12.IC.C.1) – ETS2.8 (A9-12.IC.C.2) – SEP2, SEP3, SEP8 (A9-12.IC.C.4) – SEP6, ETS2.8 (A9-12.IC.SI.1) – SEP8 (A9-12.IC.SLE.1) – SEP6, SEP 7, SEP8, HS-ETS1-3 (A9-12.IC.SLE.3) – HS-ETS1-3, SEP7
NVACS for Mathematics	(A9-12.IC.C.2) – SMP 4 (A9-12.IC.C.3) – SMP 7 (A9-12.IC.SI.1) – SMP 5
NVACS for ELA	(A9-12.IC.C.1) – RI.6.8 (A9-12.IC.C.2) – RI.6.8 (A9-12.IC.SLE.1) – SL.9-10.1
NVACS for Social Studies	(A9-12.IC.SLE.1-3) – Connect to financial literacy (risk)

Networks and the Internet

** These Advanced Computer Science Standards are considered to be higher level concepts that may be used by your advanced students, incorporated into upper level courses, and/or used in Career and Technical Education (CTE) programs. These are in addition to the 9-12 Computer Science Standards and are optional.*

By the end of Grade 12, students who demonstrate understanding can:

Indicator	Standard
A9-12.NI.NCO.1	Describe the issues that impact network functionality (e.g., bandwidth, load, delay, topology).
A9-12.NI.C.1	Compare ways software developers protect devices and information from unauthorized access.

Practice Connection	Concept Connection
Practice 7. Communicating About Computing *2. Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose* (A9-12.NI.NCO.1 and A9-12.NI.C.1).	**Networks and the Internet** • NI.NCO: Network, Communication, and Organization - *Network topology is determined, in part, by how many devices can be supported. Each device is assigned an address that uniquely identifies it on the network. The scalability and reliability of the Internet are enabled by the hierarchy and redundancy in networks* (A9-12.NI.NCO.1). • NI.C: Cybersecurity - *Network security depends on a combination of hardware, software, and practices that control access to data and systems. The needs of users and the sensitivity of data determine the level of security implemented* (A9-12.NI.C.1).

	Connections to other NVACS at this grade level
NVACS for Science	(A9-12.NI.NCO.1) – SEP8, HS-ETS1-1 (A9-12.NI.C.1) – SEP2 (A9-12.NI.C.2) – HS-ETS1-1, HS-ETS1-2 (A9-12.NI.C.3) – HS-ETS1-3 (A9-12.NI.C.4) – HS-ETS1-3
NVACS for Mathematics	None
NVACS for ELA	(A9-12.NI.NCO.1) – W.6.2 (A9-12.NI.C.1) – RI.6.9
NVACS for Social Studies	(A9-12.NI.C.1-4) – Connect to financial literacy (risk)

Glossary

This glossary includes terms used within the standards and are provided to help individuals understand the content. These definitions may or may not be given to the students as written.

abstraction—[1] (process) the process of reducing complexity by focusing on the main idea; hiding details irrelevant to the question at hand and bringing together related and useful details; reduces complexity and allows one to focus on the problem; [2] (product) a new representation of a thing, a system, or a problem that helpfully reframes a problem by hiding details irrelevant to the question at hand

accessibility—the design of products, devices, services, or environments for people who experience disabilities and generally accepted by professional groups, such as the Web Content Accessibility Guidelines (WCAG) 2.0 and Accessible Rich Internet Applications (ARIA) standards

algorithm—a step-by-step process to complete a task

analog—the defining characteristic of data that is represented in a continuous, physical way; whereas digital data is a set of individual symbols, analog data is stored in physical media, such as the surface grooves on a vinyl record, the magnetic tape of a VCR cassette, or other non-digital media

app—a type of application software designed to run on a mobile device, such as a smartphone or tablet computer; also known as a mobile application

artifact—anything created by a human (see computational artifact)

audience—expected end users of a computational artifact or system

authentication—the verification of the identity of a person or process

automate—to link disparate systems and software so that they become self-regulating

automation—the process of automating

Boolean—a type of data or expression with two possible values: true and false

bug—an error in a software program that may cause a program to unexpectedly quit or behave in an unintended manner

code—any set of instructions expressed in a programming language

comment—a programmer-readable annotation in the code of a computer program added to make the code easier to understand and generally ignored by machines

complexity—the minimum amount of resources, such as memory, time, or messages, needed to solve a problem or execute an algorithm

component—an element of a larger group; usually provides a particular service or group of related services

computational—relating to computers or computing methods

computational artifact—anything created by a human using a computational thinking process and a computing device including but not limited to, a program, image, audio, video, presentation, or web page file

computational thinking—the human ability to formulate problems so that their solutions can be represented as computational steps or algorithms to be executed by a computer

computer—a machine or device that performs processes, calculations, and operations based on instructions provided by a software or hardware program

Glossary

computer science—the study of computers and algorithmic processes, including their principles, their hardware and software designs, their implementation, and their impact on society

computing—any goal-oriented activity requiring, benefiting from, or creating algorithmic processes

computing device—a physical device that uses hardware and software to receive, process, and output information, such as computers, mobile phones, and computer chips inside appliances

computing system—a collection of one or more computers or computing devices, together with their hardware and software, integrated for the purpose of accomplishing shared tasks (although a computing system can be limited to a single computer or computing device, it more commonly refers to a collection of multiple connected computers, computing devices, and hardware)

conditional—a feature of a programming language that performs different computations or actions depending on whether a programmer-specified Boolean condition evaluates to true or false and could refer to a conditional statement, conditional expression, or conditional construct

configuration—[1] (process) defining the options that are provided when installing or modifying hardware and software or the process of creating the configuration; [2] (product) the specific hardware and software details that tell exactly what the system is made up of, especially in terms of devices attached, capacity, or capability

connection—a physical or wireless attachment between multiple computing systems, computers, or computing devices

connectivity—a program's or device's ability to link with other programs and devices

control—the general power to direct the course of actions; in programming, the use of elements of programming code to direct which actions take place and the order in which they take place

control structure—a programming (code) structure that implements control, such as conditionals and loops

crowdsourcing—gathering services, ideas, or content from a large group of people, especially from the online community at the local level (e.g. classroom or school) or global level (e.g. age-appropriate online communities, like Scratch and Minecraft)

culture—a human institution manifested in the learned behavior of people, including their specific belief systems, language(s), social relations, technologies, institutions, organizations, and systems for using and developing resources

cultural practices—the displays and behaviors of a culture

cybersecurity—the protection against access to, or alteration of, computing resources through the use of technology, processes, and training

data—information that is collected and used for reference or analysis; digital or nondigital information, including numbers, text, show of hands, images, sounds, or video

data structure—a particular way to store and organize data within a computer program to suit a specific purpose so that it can be accessed and worked with in appropriate ways

Glossary

data type—a classification of data that is distinguished by its attributes and the types of operations that can be performed on it, such as integer, string, Boolean (true or false), and floating-point

debugging—the process of finding and correcting errors (bugs) in programs

decompose—to break down into components

decomposition—breaking down a problem or system into components

device —A unit of physical hardware that provides one or more computing functions within a computing system that can provide input to the computer, accept output, or both

digital—a characteristic of electronic technology that uses discrete values, generally 0 and 1, to generate, store, and process data

digital citizenship—the norms of appropriate, responsible behavior with regard to the use of technology

efficiency—a measure of the amount of resources an algorithm uses to find an answer, usually expressed in terms of the theoretical computations, the memory used, the number of messages passed, the number of disk accesses, etc.

encapsulation—the technique of combining data and the procedures that act on it to create a type

encryption—the conversion of electronic data into another form, called ciphertext, which cannot be easily understood by anyone except authorized parties

end user—a person for whom a hardware or software product is designed (as distinguished from the developers)

event—any identifiable occurrence that has significance for system hardware or software; user-generated events include keystrokes and mouse clicks; system-generated events include program loading and errors

event handler—a procedure that specifies what should happen when a specific event occurs

execute—to carry out (or “run”) an instruction or set of instructions (e.g., program, app)

execution—the process of executing an instruction or set of instructions

hardware—the physical components that make up a computing system, computer, or computing device

hierarchy—an organizational structure in which items are ranked according to levels of importance

human-computer interaction (HCI) —the study of how people interact with computers and to what extent computing systems are or are not developed for successful interaction with human beings

identifier—the user-defined, unique name of a program element (such as a variable or procedure) in code; an identifier name should indicate the meaning and usage of the element being named

implementation—the process of expressing the design of a solution in a programming language (code) that can be made to run on a computing device

inference—a conclusion reached on the basis of evidence and reasoning

input—the signals or instructions sent to a computer

integrity—the overall completeness, accuracy, and consistency of data

Glossary

internet—the global collection of computer networks and their connections, all using shared protocols to communicate

iterative—involving the repeating of a process with the aim of approaching a desired goal, target, or result

linked list—a list in which each item contains both data and a pointer to one or both neighboring items, thus eliminating the need for the data items to be ordered in memory

loop—a programming structure that repeats a sequence of instructions as long as a specific condition is true

memory—temporary storage used by computing devices

model—a representation of some part of a problem or a system (this definition differs from that used in science)

modularity—the characteristic of a software/web application that has been divided (decomposed) into smaller modules that might have several procedures that are called from inside its main procedure and existing procedures could be reused by recombining them in a new application

module—a software component or part of a program that contains one or more procedures; one or more independently developed modules make up a program

network—a group of computing devices (personal computers, phones, servers, switches, routers, etc.) connected by cables or wireless media for the exchange of information and resources

operation—an action, resulting from a single instruction, that changes the state of data

output—a place where power or information leaves a system

packet—the unit of data sent over a network

parameter—a special kind of variable used in a procedure to refer to one of the pieces of data received as input by the procedure

phenomenon—(*pl* phenomena*)* a fact, occurrence, or circumstance observed or observable

piracy—the illegal copying, distribution, or use of software

procedure—an independent code module that fulfills some concrete task and is referenced within a larger body of program code; the fundamental role of offering a single point of reference for some small goal or task that the developer or programmer can trigger by invoking the procedure itself (in these standards, procedure is used as a general term that may refer to an actual procedure or a method, function, or module of any other name by which modules are known in other programming languages)

process—a series of actions or steps taken to achieve a particular outcome

program—(n) a set of instructions that the computer executes to achieve a particular objective; (v) to produce a program by programming

programming—the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them

protocol—the special set of rules used by endpoints in a telecommunication connection when they communicate, specifying interactions between the communicating entities

prototype—an early approximation of a final product or information system, often built for demonstration purposes

Glossary

pseudocode—computing instructions written in symbolic code which must be translated into a program language before they can be executed

redundancy—a system design in which a component is duplicated, so if it fails, there will be a backup

reliability—An attribute of any system that consistently produces the same results, preferably meeting or exceeding its requirements

remix—the process of creating something new from something old, originally a process that involved music; creating a new version of a program by recombining and modifying parts of existing programs, and often adding new pieces, to form new solutions

router—a device or software that determines the path that data packets travel from source to destination

scalability—the capability of a network to handle a growing amount of work or its potential to be enlarged to accommodate that growth

security—see cybersecurity

simulate—to imitate the operation of a real-world process or system

simulation—imitation of the operation of a real-world process or system

software—programs that run on a computing system, computer, or other computing device

storage—a mechanism that enables a computer to retain data, either temporarily or permanently; a place, usually a device, into which data can be entered, in which the data can be held, and from which the data can be retrieved at a later time; a process through which digital data is saved within a data storage device by means of computing technology

string—a sequence of letters, numbers, and/or other symbols. A string might represent, for example, a name, address, or song title (functions commonly associated with strings are length, concatenation, and substring)

structure—a general term used to discuss the concept of encapsulation without specifying a particular programing methodology

switch—a high-speed device that receives incoming data packets and redirects them to their destination on a local area network (LAN)

system—a collection of elements or components that work together for a common purpose (see computing system)

test case—a set of conditions or variables under which a tester will determine whether the system being tested satisfies requirements or works correctly

topology—the physical and logical configuration of a network; the arrangement of a network, including its nodes and connecting links; logical topology is the way devices appear connected to the user; physical topology is the way they are actually interconnected with wires and cables

troubleshooting—a systematic approach to problem solving that is often used to find and resolve a problem, error, or fault within software or a computing system

user—see end user

variable—a symbolic name that is used to keep track of a value that can change while a program is running, including numbers, text, whole sentences (strings), or logical values (true or false); a data type associated with a data storage location; a value is normally changed during the course of program execution (this definition differs from that used in math)

References

Definitions were either directly taken from the sources listed below, or were excerpted or adapted for the purposes of these standards.

College Board. (2016). *AP Computer Science Principles course and exam description.* New York, NY: College Board. Retrieved from College Board

Computer Science Teachers Association (2017). *CSTA K-12 Computer Science Standards, Revised 2017*. Retrieved from CS Teachers.

Computing At School. (2013). *Computing in the national curriculum: A guide for primary teachers*. Belford, UK: Newnorth Print. Retrieved from Computing At School

Howe, Dennis. (n.d.). The Free On-line Dictionary of Computing. Retrieved from The Free Online Dictionary of Computing

Kafai, Y., & Burke, Q. (2014). *Connected code: Why children need to learn programming*. Cambridge, MA: MIT Press.

Lee, I. (2016). *Reclaiming the roots of CT. CSTA Voice: The Voice of K-12 Computer Science Education and its Educators, 12(1), 3-4.* Retrieved from CS Teachers

Massachusetts Department of Elementary and Secondary Education. (2016, June). *2016 Massachusetts digital literacy and computer science (DLCS) curriculum framework.* Maiden, A: Author. Retrieved from Massachusettes Department of Elementary and Secondary Education

National Council for the Social Studies. (2013). *The college, career, and civic life (C3) framework or social studies state standards: Guidance for enhancing the rigor of K-12 civics, economics, geography, and history.* Silver Spring, MD: Author. Retrieved from National Council for the Social Studies

Oxford dictionaries. (n.d.). Retrieved from Oxford Dictionaries

PCmag.com encyclopedia. (n.d.). Retrieved from PC Mag Encyclopedia

Pieterse, V., & Black, P.E. (Eds.). (n.d.). *Dictionary of algorithms and data structures.* Retrieved from Dictionary of Algorithms and Data Structures

Ross, B. (2016, May 10). *What is automation and how can it improve customer service? Information Age.* Retrieved from Information Age

Software testing fundamentals. (n.d.). Retrieved from Software Testing Fundamentals

South Carolina Department of Education. (May, 2017). *South Carolina Computer Science and Digital Literacy Standards.* Retrieved from South Carolina Department of Education

TechTarget network. (n.d.). Retrieved from Tech Target

Tech terms computer dictionary. (n.d.). Retrieved from Tech Terms

Techopedia technology dictionary. (n.d.). Retrieved from Techopedia

The free dictionary. (n.d.). Retrieved from The Free Dictionary

Tucker, A., McCowan, D., Deek, F., Stephenson, C., Jones, J., & Verno, A. (2016). *A model curriculum for K-12 computer science: Report of the ACM K-12 task force curriculum committee* (2nd ed.). New York, NY: Association for Computing Machinery.

Webopedia. (n.d.). Retrieved from Webopedia

Wikipedia: The free encyclopedia. (n.d.). Retrieved from Wikipedia