



# LARAMIE COUNTY SCHOOL DISTRICT 2

---

*Students First*

## Computer Science Curriculum

**Approved by the Laramie County School District #2  
Board of Trustees**

**To be submitted to board in December, 2022**

**Revisions approved**

## Table of Contents

Acknowledgements	3
Introduction	4
Mission	6
Course/Grade Level Purposes	6
Kindergarten	9
First Grade	14
Second Grade	21
Third Grade	27
Fourth Grade	36
Fifth Grade	43
Sixth Grade	50
Change course name	58
Change course name	65
Change course name	74
Change course name	83
Change course name	90
Change course name	98
Change course name	104
Change course name	106
Change course name	109
Appendix A Change / Keep if needed	116

## Acknowledgements

Laramie County School District #2 would like to acknowledge the following people for their dedication, support, and hard work put forth during the development of this curriculum:

### **Computer Science Subject Area Committee (SAC):**

Elementary Grades

Sara Schnell

Secondary Grades

Mary Jacobsen - PBHS

Heather Goodwine - BHS

Brittany Mitchell - BHS

Todd Sweeter - PBHS Principal

Traci Schneider – LCSD2 Curriculum Coordinator

## Introduction

The purpose of Laramie County School District #2's Computer Science Curriculum is to provide a clear, organized framework on which to build instruction in the classroom. The curriculum includes clear outcomes and components of these outcomes which further clarify the skills necessary to achieve each outcome. Each outcome also describes the depth of knowledge and level of rigor required for students to demonstrate their conceptual understanding of the knowledge and skills outlined in the curriculum.

**All standards and outcomes are not created equal.** The Computer Science curriculum has been carefully aligned to include complete coverage of the Wyoming Computer Science Standards. However, it is important to note that all standards are not of equal importance. Some standards, commonly called priority standards, are essential for students to master in order to be prepared for the next grade level or course. Others, commonly referred to as supporting standards, are taught within the context of the priority standards, but do not receive the same degree of instruction and assessment as the priority standards. They act as a scaffold to help students understand and attain the priority standards.

LCSD2 has analyzed the content of each course and prioritized the concepts of greater importance. These are called **Essential Standards** and are noted in each course curriculum for grades 7-12. In grades K-6, essential standards are described in the **Proficiency Scales**. LCSD2 teachers may access the scales in the Curriculum Library. Parents may request copies from the building administrator. District assessments are created to assess student learning on these essential standards.

Each outcome has been assigned a code number consisting of symbols for grade level or course, content area, sub content area and outcome number. In the example shown below, CS stands for Computer Science - K stands for kindergarten (grade) - AP stands for Algorithms and Programming (content area) - M stands for Modularity (sub-content) – 1 symbolizes that it is the first outcome in this sub level.

Example:

**Outcome CS.K.AP.M.01:**

**Using grade appropriate content and complexity, decompose (breakdown) the steps needed to solve a problem into a precise sequence of instructions. (1A-AP-11)**

At the end of each outcome, the code number in parentheses indicates the CSTA K-12 Computer Science Standards to which it aligns and includes the identifier, concept, and standard number. In the above example, the 1A stands for grade levels K-2, AP stands for Algorithms & Programming, and 11 stands for standard 11 under 1A level of CSTA k-12 CS Standards.

A link to the complete Wyoming Computer Science State Standards document can be found [here](#).

## Mission

LCSD2 believes Computer Science will empower students with 21st Century skills like creativity, collaboration, problem solving, and communication to become participants in a technological society. Take students from consuming technology to producing technology to face the challenges that impact the world.

### **Computer Science:**

Computer Science is the study of computing principles, design, and applications (hardware & software); the creation, access, and use of information through algorithms and problem solving, and the impact of computing on society

### **Computational Thinking:**

Computational thinking is a necessary and meaningful 21st century skill. Computational thinking is defined as the thought processes involved in formulating a problem and expressing its solutions in such a way that a computer (human or machine) can effectively carry them out. Computational thinking develops into competencies in problem solving, critical thinking, productivity, and creativity. Over time, engaging in computational thought builds a student's capacity to persevere, work efficiently, gain confidence, recognize and resolve ambiguity, generalize concepts, and communicate effectively. In order to adapt to global advancements in technology, students will need to use their computational thinking skills to formulate, articulate, and discuss solutions in a meaningful manner.

## Course/Grade Level Purposes

### **Kindergarten**

Students in kindergarten will be able to protect their information. Students will follow step-by-step instructions to accomplish tasks and problem-solve. Students will work respectfully and responsibly with others in an online environment.

### **First Grade**

Students in first grade will be able to protect their information. Students will follow step-by-step instructions and create programs with help to accomplish tasks. Students will work respectfully and responsibly with others in an online environment.

### **Second Grade**

Students in second grade will be able to protect and safeguard their information. Students will follow and write step-by-step instructions and create programs to accomplish tasks. Students will work respectfully and responsibly with others in an online environment.

### **Third Grade**

Students will be able to model and discuss internal and external computing systems, as well as troubleshoot problems that may occur. Students will be able explore and discuss real-world problems and processes related to networks and the internet. In addition, students will also be able to collect and analyze data to support inferences and models. Building on their previous understanding of algorithms and programming, students will work collaboratively and independently to create and modify by increasing complex programs for a variety of purposes. Students will be able to explain the cultural, social, and ethical impacts of computing.

#### **Fourth Grade**

Students will be able to model and discuss internal and external computing systems, as well as troubleshoot problems that may occur. Students will be able explore and discuss real-world problems and processes related to networks and the internet. In addition, students will also be able to collect and analyze data to support inferences and models. Building on their previous understanding of algorithms and programming, students will work collaboratively and independently to create and modify by increasing complex programs for a variety of purposes. Students will be able to explain the cultural, social, and ethical impacts of computing.

#### **Fifth Grade**

Students in fifth grade will be able to model and discuss internal and external computing systems, as well as troubleshoot problems that may occur. Students will be able explore and discuss real-world problems and processes related to networks and the internet. In addition, students will also be able to collect and analyze data to support inferences and models. Building on their previous understanding of algorithms and programming, students will work collaboratively and independently to create and modify by increasing complex programs for a variety of purposes. Students will be able to explain the cultural, social, and ethical impacts of computing.

#### **Sixth Grade**

Students in sixth grade will design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. Students will decompose problems into parts to facilitate the design, implementation, and review of the program and create procedures with parameters to organize code and make it easier to reuse. Students will describe the trade-off between allowing information to be public and keeping information private and secure. Students will discuss the legal, social, and ethical impacts associated with software development and use, including both positive and malicious intent. Students will continue to work collaboratively and independently to create and modify increasingly complex programs for a variety of purposes introduced in grades three to five.

#### **Seventh Grade**

Students in seventh grade will model the role of protocols in transmitting data across networks and the internet (start of cybersecurity, networks and the internet). Students will critique physical and digital procedures that could be implemented to protect electronic data/information. (Sounds like data footprint or cyber security) Students will use and refine computational tools to transform collected data in order to make it more useful and

reliable(Sounds like creating graphs), create flowcharts and pseudocode to design algorithms to solve complex problems.(Sounds like intro to coding) Students will create clearly named variables that represent different data types and perform operations on their values. (could be excel) Students will practice grade level appropriate behavior and responsibilities while participating in an online community, (data footprint) including identifying and Reporting inappropriate behavior.

### **Eighth Grade**

Students in eighth grade will work collaboratively and independently to create and modify increasingly complex programs for a variety of purposes. (Sounds like coding) Students will systematically identify, recommend, resolve, and document increasingly complex software and Hardware problems with Computing devices and their components. (Physical computing) Students will also discuss the legal, social, and ethical impacts associated with software development and use, including both positive and malicious intent. (Cyber security) Students will also practice grade level appropriate behavior and responsibilities while participating in an online community, including identifying and Reporting inappropriate behavior. (digital footprint) Students will describe impacts associated with Computing technologies (impacts of computing) that affect people's everyday activities and career options along with issues with bias and accessibility in the design and technology. (creating apps with bias-ex: facial recognition) Students will seek and incorporate feedback from team members and users to refine a solution to a problem. (teamwork, collaboration)

### **Computer Science Principals**

### **Web Page Design**

## **Kindergarten**

### **Kindergarten Computer Science Terms and Definitions**

**Word** - definition

**Outcome K.CS.D.01:**

Students will, with guidance, follow directions and start to make appropriate choices to use computing devices to perform a variety of tasks (e.g., turn on, select, open and close programs, logon and logoff). (1A-CS-01)

**\*Outcome K.CS.HS.01:**

Students will use appropriate terminology to identify and use common computing devices, components, and software in a variety of environments (e.g., desktop computer, laptop computer, tablet device, monitor, keyboard, mouse, or printer). (1A-CS-02)

**Outcome K.CS.T.01:**

Students will recognize computing systems might not work as expected and identify and effectively communicate simple hardware or software problems. Implement solutions with guidance (e.g., volume turned down on headphones, monitor turned off). (1A-CS-03)

**Outcome K.NI.NCO.01:**

Students will recognize and discuss that computing devices can be connected together.

**\*Outcome K.NI.C.01:**

Students will discuss what authentication factors are and why we do not share them with others. With guidance, use them to access technological devices, apps, etc. (e.g., log in cards or user name and password) (1A-NI-04)

**Outcome K.DA.S.01:**

Students will, with guidance, locate, open, modify and save an existing file with a computing device. (1A-DA-05)

**Outcome K.DA.CVT.01:**

Students will, with guidance, collect data and present it visually. (1A-DA-06)

**Outcome K.DA.IM.01:**

Students will, with guidance, draw conclusions and make predictions based on picture graphs or patterns with or without a computing device (e.g., make predictions based on weather data presented in a picture graph or complete a pattern). (1A-DA-07)

**\*Outcome K.AP.A.01:**

Students will, with guidance, model daily processes and follow algorithms (sets of step-by-step instructions) to complete tasks (e.g., verbally, kinesthetically, with robot devices, or a programming language). (1A-AP-08)

**Outcome K.AP.V.01:**

Students will, with guidance, demonstrate that data may be represented by symbols (e.g., thumbs up/down as representations of yes/no, arrows when writing algorithms to represent



direction, or encode and decode words using numbers, pictographs, or other symbols to represent letters or words). (1A-AP-09)

**\*Outcome K.AP.C.01:**

Students will, with guidance, independently or collaboratively create programs to accomplish tasks using sequencing (emphasizing the beginning, middle, and end). (1A-AP-10)

**\*Outcome K.AP.M.01:**

Students will use grade appropriate content and complexity, decompose (breakdown) the steps needed to solve a problem into a precise sequence of instructions (e.g., to show the life cycle of a plant - plant seed in dirt, water dirt, plant begins to grow with sunlight). (1A-AP-11)

**Outcome K.AP.PD.01:**

Students will, with guidance, develop plans that describe a program's sequence of events, goals, and expected outcomes. (1A-AP-12)

**Outcome K.AP.PD.02:**

Students will, with guidance, give credit to ideas, creations, and solutions of others while developing algorithms. (1A-AP-13)

**\*Outcome K.AP.PD.03:**

Students will, with guidance, independently or collaboratively debug (identify and fix errors) algorithms using a programming language and/or unplugged activity. (1A-AP-14)

**Outcome K.AP.PD.04:**

Students will use correct terminology (beginning, middle, end) in the development of an algorithm. (1A-AP-15)

**Outcome K.IC.C.01:**

Students will discuss different ways in which technology is used in your daily life. (1A-IC-16)

**\*Outcome K.IC.SI.01:**

Students will, with guidance, identify appropriate manners while participating in an online environment. (1A-IC-17)

## **First Grade**

### **1<sup>st</sup> grade Computer Science Terms and Definitions**

**Word - definition**

**Outcome 1.CS.D.01:**

Students will, with guidance, select and use a computing device to perform a variety of tasks for an intended outcome (e.g., turn on, select, open and close programs, logon and logoff). (1A-CS-01)

**\*Outcome 1.CS.HS.01:**

Students will use appropriate terminology in naming and demonstrate the function of common computing devices, components, and software (e.g., use of a printer, appropriate input device use, or common operating system features). (1A-CS-02)

**Outcome 1.CS.T.01:**

Students will recognize computing systems might not work as expected and identify and effectively communicate simple hardware or software problems. Implement solutions with guidance (e.g., app or program is not working as expected, no sound is coming from the device, caps lock turned on). (1A-CS-03)

**Outcome 1.NI.NCO.01:**

Students will identify and describe that by connecting computing devices together they can share information (e.g., remote storage, printing, the internet).

**\*Outcome 1.NI.C.01:**

Students will identify what authentication factors are, explain why they are not shared, and discuss what makes authentication effective. Independently use them to access technological devices, apps, etc. (1A-NI-04)

**Outcome 1.DA.S.01:**

Students will, with guidance, locate, open, modify and save an existing file, and use appropriate file-naming conventions. Recognize that the file exists within an organizational structure (drive, folder, file). (1A-DA-05)

**Outcome 1.DA.CVT.01:**

Students will, with guidance, collect data and present it in more than one way (e.g. written and visual presentation). (1A-DA-06)

**Outcome 1.DA.IM.01:**

Students will, with guidance, identify and interpret data from a chart or graph (visualization) in order to make a prediction, with or without a computing device. (1A-DA-07)

**\*Outcome 1.AP.A.01:**

Students will, with guidance, identify and model daily processes and follow algorithms (sets of step-by-step instructions) to complete tasks (e.g., verbally, kinesthetically, with robot devices, or a programming language). (1A-AP-08)

**Outcome 1.AP.V.01:**

Students will, with guidance, demonstrate that computers represent data using numbers, letters, words, or other symbols (e.g., thumbs up/down as representations of yes/no, arrows when writing algorithms to represent direction, or encode and decode words using numbers, pictographs, or other symbols to represent letters or words). (1A-AP-09)

**\*Outcome 1.AP.C.01:**

Students will, with guidance, independently or collaboratively create programs to accomplish tasks using sequencing, conditionals, and repetition (e.g., program a robot device, or algorithmically describe an unplugged activity). (1A-AP-10)

**\*Outcome 1.AP.M.01:**

Students will use grade appropriate content and complexity, decompose (breakdown) the steps needed to solve a problem into a precise sequence of instructions (e.g., given a deck of cards, have students sort them by multiple methods - color, suit, or rank). (1A-AP-11)

**Outcome 1.AP.PD.01:**

Students will independently or with guidance, develop plans that describe a program's sequence of events, goals, and expected outcomes. (1A-AP-12)

**Outcome 1.AP.PD.02:**

Students will independently or with guidance, give credit to ideas, creations, and solutions of others while writing and/or developing programs. (1A-AP-13)

**\*Outcome 1.AP.PD.03**

Students will, with guidance, independently or collaboratively debug (identify and fix errors) programs using a programming language and/or unplugged activity. (1A-AP-14)

**Outcome 1.AP.PD.04:**

Students will use correct terminology (first step, second step, third step) and explain the choices made in the development of an algorithm. (1A-AP-15)

**Outcome 1.IC.C.01:**

Students will identify how people use different types of technologies in their daily work and personal lives. (1A-IC-16)

**\*Outcome 1.IC.SI.01:**

Students will, with guidance, identify appropriate and inappropriate behavior. Act responsibly while participating in an online community and know how to report concerns. (1A-IC-17)

## **Second Grade**

## 2<sup>nd</sup> grade Computer Science Terms and Definitions

### Word - definition

#### Outcome 2.CS.D.01:

Students will independently select and use a computing device to perform a variety of tasks for an intended outcome. (1A-CS-01).

#### Outcome 2.CS.HS.01:

Students will demonstrate and describe the function of common components of a computing system (hardware and software) (e.g., use a browser, search engine) (1A-CS-02).

#### Outcome 2.CS.T.01:

Students will recognize computing systems might not work as expected. Identify and effectively communicate simple hardware or software problems to implement solutions and discuss problems with peers and adults. (1A-CS-03).

#### Outcome 2.NI.NCO.01:

Students will identify and describe that computing devices can be connected in a variety of ways (e.g., Bluetooth, Wi-Fi, home and school network, the internet).

#### Outcome 2.NI.C.01:

Students will explain what authentication factors (e.g., login) are, why we use them, apply authentication to protect devices and information (personal and private) from unauthorized access. (1A-NI-04)

#### Outcome 2.DA.S.01:

Students will, with guidance, develop and modify an organizational structure by: - creating folders. - copying existing folders and files. - moving existing folders and files. - deleting folders and files. (1A-DA-05).

#### Outcome 2.DA.CVT.01:

Students will independently present data in multiple formats (e.g., as a table and graph or as a table and chart). (1A-DA-06)

#### Outcome 2.DA.IM.01:

Students will, with guidance: - interpret data. - present it in a chart or graph (visualization). - make a prediction based on the data, with or without a computing device. (1A-DA-07).

#### \*Outcome 2.APA.01:

Students will, with guidance, identify and model daily processes by creating and following algorithms (sets of step-by- step instructions) to complete tasks (e.g., verbally, kinesthetically, with robot devices, or a programming language) (1A-AP-08).

**Outcome 2.AP.V.01:**

Students will model the way programs store and manipulate data by using numbers or other symbols to represent information (e.g., thumbs up/ down as representation of yes / no, arrows when writing algorithms to represent direction, or encode and decode words using numbers, pictographs, or other symbols to represent letters or words ). (1A-AP-09)

**\*Outcome 2.AP.C.01:**

Students will, with guidance, independently and collaboratively create programs to accomplish tasks using a programming language, robot device, or unplugged activity that includes sequencing, conditionals, and repetition (1A-AP-10).

**\*Outcome 2.AP.M.01:**

Students will use grade appropriate content and complexity, decompose (breakdown) the steps needed to solve a problem into a precise sequence of instructions (e.g., develop a set of instructions on how to play your favorite game). (1A-AP-11)

**Outcome 2.AP.PD.01:**

Students will develop plans that describe the program's sequence of events, goals, and expected outcomes.(1A-AP-12).

**Outcome 2.AP.PD.02:**

Students will give credit to ideas, creations, and solutions of others while writing and developing programs. (1A-AP-13)

**\*Outcome 2.AP.PD.03:**

Students will independently and collaboratively debug (identify and fix errors) programs using a programming language. (1A-AP-14)

**Outcome 2.AP.PD.04:**

Students will use correct terminology (debug, program input/output, code) to explain the development of a program or an algorithm (e.g., in an unplugged activity, Hands-On manipulatives, or programming language ). (1A-AP-15)

**Outcome 2.IC.C.01**

Students describe how people use different types of technology in their daily work and personal lives. (1A-IC-16)

**\*Outcome 2.IC.SI.01:**

Students will practice grade-level appropriate behavior and responsibilities while participating in an online community. Identify and report inappropriate behavior (1A-IC-17).

## Third Grade

### 3<sup>rd</sup> grade Computer Science Terms and Definitions

#### Word - definition

##### Outcome 3.CS.D. 01:

With guidance, students will follow directions, identify how internal and external parts of computing devices function to form a system. (1B-CS-01).

##### \*Outcome 3.CS.HS.01:

Students will model how information flows through hardware and software to accomplish tasks (1B-CS-02).

##### Outcome 3.CS.T.01

Students will identify hardware and software problems that may occur during everyday use, then develop, apply, and explain strategies for solving these problems (e.g., Refresh the screen, closing and reopening an application or file, unyielding or adjusting the volume on headphones). (1B-CS-03)

##### Outcome 3.NI.NCO.01:

students will use observations of everyday situations to illustrate that information is sent and received over physical and wireless paths (1B-NI-04).

##### \*Outcome 3.NI.C.01:

Students will identify cybersecurity problems that relate to inappropriate use of computing devices and networks. (1B-NI-05)

##### \*Outcome 3.DA.S.01:

Students will demonstrate that different types of information are stored in different formats that have Associated programs (e.g., documents open in word processor) and different storage requirements.

##### Outcome 3.DA.CVT.01:

Students will independently collect and present data in various visual formats. (1B-DA-06)

##### Outcome 3.DA.IM.01:

Students will, with guidance, use that data to make predictions and discuss whether there is adequate data to make reliable predictions. (1B-DA-07)

**\*Outcome 3.AP.A.01:**

Students will use grade appropriate content and complexity, compare and refine multiple algorithms for the same task and determine which is the most appropriate. (1B-AP-08).

**\*Outcome 3.AP.V.0:**

Students will use grade appropriate content and complexity, create programs that use variables to store and modify data. (1B-AP-09)

**\*Outcome 3.AP.C.01:**

Students will use appropriate content in complexity, create programs that include sequences, events, loops, and conditionals both individually and collaboratively.(1B-AP-10)

**Outcome 3.AP.M.01:**

Students will be able to decompose problems into smaller, manageable subproblems to facilitate the program development process. (1B-AP-11)

**Outcome 3.AP.M.02**

Students will be able to modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features. (1B-AP-12).

**Outcome 3.AP.PD.01:**

Students will use an iterative process to plan the development of a program (i.e., flowchart, storyboard, story map, or pseudocode). (1B-AP-13)

**Outcome 3.AP.PD.02**

Students will observe intellectual property rights and give appropriate credit when creating or remixing programs. (1B-AP-14)

**Outcome 3.AP.PD.03**

Students will test and debug (i.e., identify and fix errors) a program or algorithm to ensure it runs as intended. (1B-AP-15)

**Outcome 3.AP.PD.04**

Students will use grade appropriate content and complexity, describe choices made during program development using code comments, presentation, and demonstrations. (1B-AP-16)

**Outcome 3.AP.PD.05**

With teacher guidance, perform varying rolls when collaborating with peers during the design, implementation, and review stages of program development. (1B-AP-17)

**Outcome 3.IC.C.01:**

Students will identify computing technologies that have changed the world and express how those Technologies influence and are influenced by cultural practices.

**Outcome 3.IC.C.02**

Students will identify possible problems and propose how computing devices could have built-in features for increasing accessibility to all users. (1B-IC-19)

**Outcome 3.IC.SI.01:**

Students will identify how computational products may be, or have been, improved to incorporate diverse perspectives. (1B-IC-20)

**Outcome 3.IC.SI.02:**

Students will practice grade-level appropriate behavior and responsibilities while participating in an online community. Identify and Report inappropriate behavior following District protocol.

**Outcome 3.IC.SLE.01:**

Students will identify types of Digital Data that may have intellectual property rights that prevent copying or require attribution. (1B-IC-21)

## **Fourth Grade**

### **4<sup>th</sup> grade Computer Science Terms and Definitions**

**Word - definition**

**Outcome 4.CS.D.01:**

With guidance, students will describe how internal and external parts of computing devices function to form a system. (1B-CS-01)

**\*Outcome 4.CS.HS.01:**

Students will model how computer hardware and software work together as a system to accomplish tasks. (1B-CS-02)

**Outcome 4.CS.T.01:**

Students will identify hardware and software problems that may occur during everyday use, then develop, apply, and explain strategies for solving these problems (e.g., rebooting the device, checking the power, force shut down of an application). (1B-CS-03)

**Outcome 4.NI.NCO.01:**

Students discuss how information is sent and received across physical or wireless path (i.e., it is broken down into smaller pieces called packets and transmitted from one location to another). (1B-NI-04)



**\*Outcome 4.NI.C.01:**

Students will identify and explain cybersecurity issues related to responsible use of technology and information, and describe personal consequences of inappropriate use. (1B-NI-05)

**\*Outcome 4.DA.S.01:**

Students will choose different storage locations (physical, shared, or cloud) based on the type of file, storage requirements (e.g., file size, availability, or available memory), and sharing requirements. (1A-DA-05)

**Outcome 4.DA.CVT.01:**

Students will organize and present collected data in a variety of ways (e.g., sonification, visualization) to highlight relationships. (1B-DA-06)

**Outcome 4.DA.IM.01:**

Students will determine how the accuracy of conclusions is influenced by the amount of data collected. (1B-DA-07)

**\*Outcome 4.AP.A.01:**

Students will use grade appropriate content and complexity, compare and refine multiple algorithms for the same task and determine which is the most appropriate. (1B-AP-08)

**\*Outcome 4.AP.V.01:**

Students will use grade appropriate content and complexity, create programs that use variables to store and modify data. (1B-AP-09)

**\*Outcome 4.AP.C.01:**

Students will use appropriate content and complexity, create programs that include sequences, events, loops, and conditionals both individually and collaboratively. (1B-AP-10)

**Outcome 4.AP.M.01:**

Students will be able to decompose problems into smaller, manageable subproblems to facilitate the program development process. (1B-AP-11).

**Outcome 4.AP.M.02**

Students will use grade appropriate content and complexity, modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features. (1B-AP-12)

**Outcome 4.AP.PD.01:**

Students will use an iterative process to plan the development of a program to include the user preferences. (1B-AP-13)

**Outcome 4.AP.PD.02:**

Students will observe intellectual property rights and give appropriate credit when creating or remixing programs. (1B-AP-14)

**\*Outcome 4.AP.PD.03**

Students will test and debug a program or algorithm to ensure it runs as intended. (1B-AP-15)

**Outcome 3.AP.PD.04 :**

Students will use grade appropriate content and complexity, describe choices made during program development using code comments, presentation, and demonstrations. (1B-AP-17)

**Outcome 4.AP.PD.05:**

With teacher guidance, students will perform varying rolls when collaborating with peers during the design, implementation, and review stages of program development. (1B-AP-16)

**Outcome 4.IC.C.01:**

Students will provide examples of computing technologies that have changed the world and express how those technologies influence and are influenced by cultural practices. (1B-IC-18)

**Outcome 4.IC.C.02:**

Students will brainstorm problems and ways to improve computing devices to increase accessibility to all users. (1B-IC-19)

**Outcome 4.IC.SI.01:**

Students will work with a team or individually to consider other perspectives on improving a computational product. (1B-IC-20)

**Outcome 4.IC.SI.02:**

Students will practice grade-level appropriate behavior and responsibilities while participating in an online community. Identify and Report inappropriate behavior following District protocol.

**Outcome 4.IC.SLE.01:**

Students will recognize and appropriately use public domain and/ or creative commons media and discuss the social impact of violating intellectual property rights. (1B-IC-21)

## **Fifth Grade**

### **5<sup>th</sup> grade Computer Science Terms and Definitions**

**Word - definition**

**Outcome 5.CS.D.01:**

Students will independently describe how internal and external parts of computing devices function to form a system. (1B-CS-01)

**\*Outcome 5.CS.HS.01:**

Students will accurately model how information is translated, transmitted, and processed in order to flow through hardware and software to accomplish tasks. (1B-CS-02)

**Outcome 5.CS.T.01:**

Students will identify hardware and software problems that may occur during everyday use, then develop, apply, and explain strategies for solving identified problems, when applicable. (1B-CS-03)

**Outcome 5.NI.NCO.01:**

Students will model and explain how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the internet, and reassembled at the destination. (1B-NI-04)

**\*Outcome 5.NI.C.01:**

Students will discuss with specificity real world cybersecurity problems, personal consequences of inappropriate use, and identify and implement appropriate strategies for how personal information can be protected. (1B-NI-05)

**Outcome 5.DA.S.01:**

Students will justify the format and location for storing data based on sharing requirements and the type of information (e.g., images, videos, text). (1A-DA-05)

**Outcome 5.DA.CVT.01:**

Students will organize and present collected data to highlight comparisons, relationships, and support a claim. (1B-DA-06)

**Outcome 5.DA.IM.01:**

Students will use data to highlight or propose relationships, predict outcomes, and communicate an idea. (1B-DA-07)

**\*Outcome 5.AP.A.01:**

Students will use grade appropriate content and complexity, compare and refine multiple algorithms for the same task and determine which is the most appropriate for the task. (1B-AP-08)

**\*Outcome 5.AP.V.01:**

Using grade appropriate content and complexity, students will create programs that use variables to store data and modify data. (1B-AP-09)

**\*Outcome 5.AP.C.01:**

Students will create programs that include combinations of sequences, events, loops, and conditionals both individually and collaboratively. (1B-AP-10)

**Outcome 5.AP.M.01:**

Students will decompose problems into smaller, manageable subproblems to facilitate the program development process. (1B-AP-11)

**Outcome 5.AP.M.02:**

Students will then modify, remix, or incorporate portions of an existing program into one's own work to develop something new or add more advanced features. (1B-AP-12)

**Outcome 5.AP.PD.01:**

Students will use an iterative process to plan the development of a program to include the user preferences. (1B-AP-13)

**Outcome 5.AP.PD.02:**

Students will observe intellectual property rights and give appropriate credit when creating or remixing programs. (1B-AP-14)

**\*Outcome 5.AP.PD.03**

Students will test and debug a program or algorithm to ensure it runs as intended. (1B-AP-15)

**Outcome 5.AP.PD.04 :**

Students will use grade appropriate content and complexity, describe choices made during program development using code comments, presentation, and demonstrations. (1B-AP-17)

**Outcome 5.AP.PD.05:**

With teacher guidance, students will perform varying rolls when collaborating with peers during the design, implementation, and review stages of program development. (1B-AP-16)

**Outcome 5.IC.C.01:**

Students will provide examples of computing technologies that have changed the world and express how those technologies influence and are influenced by cultural practices. (1B-IC-18)

**Outcome 5. IC.C.02:**

Students will develop, test, and refine digital artifacts or devices to improve accessibility and usability for diverse end users. (1B-IC-19)

**Outcome 5.IC.SI.01:**

Students will seek diverse perspectives for the purpose of improving computational artifacts. (1B-IC-20)

**\*Outcome 5.IC.SI.02:**

**Students will practice grade-level appropriate behavior and responsibilities while participating in an online community. Identify and report inappropriate behavior following District protocol.**

**Outcome 5.IC.SLE.01:**

**Students will recognize and appropriately use public domain and/ or creative commons media and discuss the social impact of violating intellectual property rights. (1B-IC-21)**

## Sixth Grade

### 6<sup>th</sup> grade Computer Science Terms and Definitions

#### Word - definition

##### Outcome 6.CS.D.01:

Students will recommend improvements to the design of computing devices based on analysis of personal interaction with the device. (2.CS.01)

##### \*Outcome 6.CS.HS.01:

Students will identify ways that hardware and software are combined to collect and exchange data. (2.CS.02)

##### Outcome 6.CS.T.01

Students will identify and determine potential solutions for increasingly complex software and hardware problems with computing devices and their components. (2.CS.03)

##### Outcome 6.NI.NCO.01:

Students will model the role of protocols in transmitting data across networks and the internet (e.g., Model A simple protocol for transferring information using packets. (2.NI.04)

##### \*Outcome 6.NI.C.01:

Students will identify existing cyber security concerns with the internet. (2.NI.05)

##### Outcome 6.NI.C.02:

Students will explain the importance of cybersecurity and describe how one method of encryption works. (2.NI.06)

##### Outcome 6.DA.S.01:

Students will represent data using multiple encoding schemes. (2-DA-07)

##### \*Outcome 6.DA.CVT.01:

Students will explore a variety of computational tools and content of their data. (2-DA-08)

##### Outcome 6.DA.IM.01:

Students will use models and simulations to formulate, define, and test hypotheses. (2-DA-09)

##### Outcome 6.AP.A.01:

Students will use existing algorithms in natural language, flowcharts, or pseudocode to solve complex problems. (2-AP-10)

##### \*Outcome 6.AP.V.01:

Using grade appropriate content and complexity, students will create clearly named variables that represent different data types and perform operations on their values. (2-AP-11)

**\*Outcome 6.AP.C.01:**

Using grade appropriate content and complexity, design and iteratively develop programs that combine control structures, including nested loops and compound conditionals (2-AP-12)

**\*Outcome 6.AP.M.01:**

Using grade appropriate content and complexity, students will decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs (2-AP-13)

**Outcome 6.AP.M.02:**

Using grade appropriate content and complexity, students will create procedures with parameters to organize code and make it easier to reuse (2-AP-14)

**Outcome 6.AP.PD.01:**

Using grade appropriate content and complexity, students will seek and incorporate feedback from team members and users to refine a solution to a problem. (2-AP-15)

**Outcome 6.AP.PD.02:**

Students will incorporate existing code, media, or libraries into original programs and give attribution (2-AP-16)

**Outcome 6.AP.PD.03:**

Students will test and refine programs using teacher provided inputs (2-AP-17).

**\*Outcome 6.AP.PD.04:**

Using grade appropriate content and complexity, students will document programs in order to make them easier to follow, test, and debug (2-AP-18).

**Outcome 6.AP.PD.05:**

Students will use a pre-written computational artifact, identify the project timeline tasks necessary for program development (2-AP-19).

**Outcome 6.IC.C.01:**

Students will explain how computing impacts people's everyday activities (2-IC-20).

**Outcome 6.IC.C.02:**

Students will explore issues of bias and accessibility in the design of technologies (2-IC-21).

**Outcome 6.IC.SI.01:**

Students will use grade appropriate content and complexity, collaborate using tools to connect with peers when creating a computational artifact (2-IC-22).

**Outcome 6.IC.SI.02:**

**Students will practice grade-level appropriate behavior and responsibilities while participating in an online community. Identify and report inappropriate behavior.**

**\*Outcome 6.IC.SLE.01:**

**Students will use grade appropriate content and complexity, describe tradeoffs between allowing information to be public and keeping information private and secure. (2-IC-23)**

**Outcome 6.IC.SLE.02:**

**Students will use grade appropriate content and complexity, discuss legal, social, and ethical impacts associated with software development and use, including both positive and malicious intent.**



## Grade 7

### Computer Science Course Name

#### **Terms and Definitions**

Word - definition

**Outcome CS-7-1:  
Outcome**

CS-7-1-1      Component

**Outcome CS-4-2:  
Outcome**

CS-7-2-1      Component

## Grade 8

### Computer Science Course Name Terms and Definitions

Word - definition

**Outcome CS-8-1:  
Outcome**

CS-8-1-1      Component

**Outcome CS-8-2:  
Outcome**

CS-8-2-1      Component

## Grade 9-12

### Computer Science Principles Terms and Definitions

**\*Miss Mitchell will complete the glossary after the course information is completed, please send me a message when you are ready for the glossary to be updated\***

Binary -

Control Structures -

Digital Artifact -

Digital Media -

Functions -

Inputs -

Interfaces -

Machine Learning Models -

Outputs -

Programming Languages -

Prototypes -

Randomness -

Sprites -

Useability -

User Input -

Variables -

Mary.jacobsen@laramie2.org

J@c616dpj

#### **Outcome CS-9-1:**

**Students will use the problem-solving cycle to solve real-world problems.**

Students will create a collaborative classroom environment where students view computer science as relevant, fun, and empowering.

Students will use a structured problem solving process to address problems and design solutions that use computing technology.

Students will identify the defined characteristics of a computer and how it is used to solve information problems.

#### **Standards met**

**1B-AP-08, 2-AP-10, 1B-AP-11, 2-AP-15 1B-AP-16, 2-AP-18**

**1B-CS-01, 1B-CS-02,**

**2-IC-20,**

**2-CS-02**

#### **Outcome CS-9-2:**

Students will create digital artifacts that use multiple computer languages to control the structure and style of their content.

Students will create a website as a form of personal expression.

Students will use different programming languages to solve different problems.

Students will examine their role and responsibilities as both creators and consumers of digital media

**Standards met**

**1B-AP-11, 1B-AP-12, 1B-AP-14, 1B-AP-16,  
2-AP-13, 2-AP-15, 2-AP-16, 2-AP-17, 2-AP-18, 2-AP-19  
3A-AP-20  
1B-CS-01, 1B-CS-02  
1B-IC-18, 1B-IC-21  
2-IC-20, 2-IC-21 2-IC-23  
2-CS-02  
1B-NI-05**

**Outcome CS-9-3:**

Student will create an interactive animation or game that includes basic programming concepts such as control structures, variables, user input, and randomness.

Student will work with others to break down programming projects using sprites and functions.

Student will give and respond constructively to peer feedback, and work with their teammates to complete a project.

Student will view himself/herself as a computer programmer, and see programming as a fun and creative form of expression.

**Outcome CS-9-4:**

Student will see the design process as a form of problem solving that prioritizes the needs of a user.

Student will identify user needs and assess how well different designs address them.

Student will develop paper and digital prototypes, gather and respond to feedback about a prototype, and consider ways different user interfaces do or do not affect the usability of their apps.

Student will understand other roles in software development, such as product management, marketing, design, and testing, and how to use what they have learned about computer science as a tool for social impact.

**Outcome CS-9-5:**

Student will understand the role of data and data representation in solving information problems.

Student will explain the necessary components of any data representation scheme, as well as the particulars of binary and the common ways that various types of simple and complex data are represented in binary code.

Student will design and implement a data-based solution to a given problem and determine how the different aspects of the problem solving process could be automated.

### **Outcome CS-9-6:**

Student will design and build a physical computing device that integrates physical inputs and outputs with digital apps.

Student will create app prototypes that use a physical device to solve real-world problems

Student will use physical computing to solve problems in fun and innovative ways.

### **Outcome CS-9-7:**

Student will create a machine learning model in AI Lab to solve a problem, and use App Lab to create an app that uses their model.

Student will understand how machine learning models make decisions from data

Student will create machine learning models from their own data to solve problems in their community.

## **Grade 10 - 12**

### **Web Page Design**

**\*Miss Mitchell will complete the glossary after the course information is completed, please send me a message when you are ready for the glossary to be updated\***

**Computing -**

**HTML -**

**Protocols -**

**Systematic Troubleshooting -**

### **Outcome CS- WPD-1:**

**Students will describe the components of an accessible website and create a plan for a website for a target audience.**

CS- WPD-1-1 Define the Internet and associated key terms(L2.NI.NCO.01)

CS- WPD-1-2 Recognize Internet protocols

CS- WPD-1-3 Discuss web browsers and identify their main features

CS- WPD-1-4 Describe the types and purposes of websites

CS- WPD-1-5 Define a wireframe and a site map

- CS- WPD-1-6 Explain how websites use graphics, navigation tools, typography, and color (L2.AP.PD.04)
- CS- WPD-1-7 Design for accessibility (L2.IC.C.02)

**Outcome CS- WPD-2:**

**Students will create and view a basic HTML webpage.**

- CS- WPD-2-1 Define Hypertext Markup Language (HTML) and HTML elements
- CS- WPD-2-2 Recognize HTML versions and web programming languages
- CS- WPD-2-3 Identify web authoring tools
- CS- WPD-2-4 Download and use a web authoring tool

**Outcome CS- WPD-3:**

**Students will create a webpage template using semantic elements.**

- CS-WPD-3-1 Explain how to manage website files
- CS-WPD-3-2 Describe and use HTML 5 semantic elements
- CS-WPD-3-3 Determine the elements to use when setting the structure of a webpage
- CS-WPD-3-4 Design and build a semantic wireframe

**Outcome CS-WPD-4:**

**Students will create and customize a home page from an HTML template.**

- CS-WPD-4-1 Insert comments in an HTML document
- CS-WPD-4-2 Add static content to a webpage template
- CS-WPD-4-3 Insert symbol codes and other character entities
- CS-WPD-4-4 Describe and use heading elements
- CS-WPD-4-5 Describe the benefits of validating web documents
- CS-WPD-4-6 Validate an HTML template

**Outcome CS-WPD-5:**

**Students will create a comprehensive webpage with accessible attributes including graphics, div elements, hyperlinks, lists, and various styles.**

- CS-WPD-5-1 Describe image file formats
- CS-WPD-5-2 Describe the image tag and its attributes
- CS-WPD-5-3 Add images to a website
- CS-WPD-5-4 Explain div elements and attributes
- CS-WPD-5-5 Use a div element within a webpage
- CS-WPD-5-6 Describe types of hyperlinks
- CS-WPD-5-7 Create relative links, absolute links, bookmark links, and email links
- CS-WPD-5-8 Describe the types of lists in an HTML document
- CS-WPD-5-9 Create an unordered list and a description list
- CS-WPD-5-10 Embed a map within a webpage Test and validate links on a webpage

**Outcome CS-WPD-6:**

**Students will utilize Cascading Style Sheets (CSS) to enhance a webpage.**

- Explain the importance of separating design from content
- Describe Cascading Style Sheets (CSS)
- Define inline, embedded, and external styles and their order of precedence
- Describe a CSS rule and its syntax
- Explain the difference between a selector, property, and value
- Create styles that use text and color properties
- Explain the difference between inline and block content
- Describe and use the CSS box model to apply margins, padding, and borders
- Create an external style sheet and link it to an HTML page
- Create responsive images
- Use float and clear properties
- Create and style id and class attributes
- Use a span element
- Add comments to an external style sheet and validate a CSS file

**Outcome CS-WPD-7:**

**Students will describe the principles of responsive design necessary for webpages to be viewed from any device.**

- Explain the principles of responsive design
- Describe the pros and cons of a mobile website
- Explain the design principles of a mobile website
- Describe a mobile-first strategy
- Define a viewport
- Insert a meta viewport element
- Use a mobile simulator
- Create a sticky element
- Integrate custom fonts
- Use a pseudo-class
- Create a mobile-friendly navigation system
- Add a telephone link
- Make rounded corners

**Outcome CS-WPD-8:**

**Students will apply the principles of responsive design to create webpages viewed from any device.**

- Identify and use media query expressions
- Explain the design principles of a tablet website
- Insert a media query to target tablet viewports
- Create style rules for tablet viewports
- Explain the design principles of a desktop website
- Insert a media query and create style rules to target desktop viewports
- Insert a media query and create a style rule for large desktop viewports
- Create a media query for print

- Identify and modify breakpoints
- Describe and add dynamic pseudo-classes to a website
- Explain linear and radial gradients
- Apply a linear gradient to a webpage for a desktop viewport

**Students will enhance webpages using common properties and elements, tables and forms, and audio and video.**

- Describe and use article, aside, and section elements
  - Describe and use figure and fig caption elements
  - Describe and use the CSS grid layout
  - Describe and use the opacity property
  - Describe and use the box sizing property
  - Describe and use the text shadow property
  - Describe and use the box shadow property
  - Insert and style figure and fig caption elements
  - Insert a section element
  - Insert and style an article element
  - Insert and style an aside element
  - Describe and add a favicon to a webpage
- 
- Define table elements
  - Describe the steps used to plan, design, and code a table
  - Create a table with rows and data
  - Insert a table caption
  - Style a table for tablet and desktop viewports
  - Describe form controls and their uses
  - Use the form and input elements
  - Create text input controls, labels, and check boxes
  - Create a selection menu with multiple options
  - Use the text area element
  - Create a Submit button
- 
- Describe the benefits and limitations of multimedia in websites
  - Identify audio formats
  - Identify video formats
  - Describe a plug-in



- Understand codecs
- Understand and create audio elements
- Understand and create video elements
- Identify common audio attributes
- Identify common video attributes
- Understand the source element
- Test audio elements
- Test video elements
- Make videos accessible

**Students will use JavaScript code to create JavaScript functions.**

- Use the CSS transform property
- Create animations with CSS keyframes
- Understand JavaScript
- Describe JavaScript code
- Understand and use the script element
- Understand where JavaScript code may be written
- Create an external JavaScript file
- Create JavaScript functions
- Understand events
- Use an onclick event handler

**Students will....search engines, publishing,**

- Define, identify, and describe forms of social media
- Describe a blog
- Describe search engines
- Explain search engine optimization (SEO)
- Create description meta tags
- Create a sitemap file
- 
- Describe a domain name and top-level domains
- Explain the role of a web hosting service
- Describe a File Transfer Protocol (FTP) client
- Publish a website with an FTP client
- Explain how to register a website with a search engine
- Describe a website development life cycle
- 
- Add a Skip to Content Link
- Minify a CSS file
- Explain project management for a website
- Define copyright and e-commerce

**Students will...use Bootstrap to style and create a webpage, navbar, navigation links, and tables.**

- Describe the Bootstrap framework
- Create a webpage using Bootstrap's starter template
- Create a Bootstrap navbar
- Integrate a Bootstrap hamburger menu
- Style navigation links using Bootstrap
- Define and create a jumbotron
- Explain how to style text using Bootstrap
- Write a jQuery document ready event
- Describe and use the Bootstrap grid system
- Explain how to style images using Bootstrap
- Style a table using Bootstrap
- Describe a content management system

**Outcome L1.IC.C.01:**

Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.

**Outcome: L2.IC.C.02**

Evaluate the impact of equity, access, and influence on the distribution of computing resource in a global society.

**Outcome L1.NI.NCO.01:**

Students will understand the issues that impact the functionality of the Internet including protocols, web browsers, and accessibility.

**Outcome L1.CS.T.01:**

Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and resolve errors.

**Outcome L1.CS.HS.01:**

Students will explain and recognize HTML and the interactions between HTML and HTML elements.

**Outcome L1.AP.M.02:**

Students will create and view basic HTML web pages by using procedures within a program, combinations of data and procedures, or independent but interrelated programs

**Outcome L1.AP.PD.01:**

Students will plan and develop webpages by managing files, using HTML semantic elements, heading elements, and webpage templates.

**Outcome L1.AP.PD.05:**

Students will document webpage design decisions using text, graphics, presentations and/or demonstration in the development of complex problems

**Outcome L2.AP.PD.03:**

Develop programs for multiple computing platforms.

**Outcome L2.AP.PD.05:**

Develop and use a series of test cases to verify that a program performs according to its design specifications.

**Outcome L1.AP.PD.06:**

Evaluate and refine designed HTML webpages to make them useable and accessible.

**Outcome L2.IC.C.03:**

Use tools and methods for collaboration.

**Outcome L1.IC.SLE.02**

Explain the privacy concerns related to the collection and generation of data through automate processes that may not be evident to users.

**Outcome L2.IC.SLE.02**

Using grade level appropriate content and complexity, discuss the legal, social, and ethical impacts associated with webpage development and use including both positive and malicious intent.

**Outcome CS-19-1:**

**Outcome**

CS-10-1-1      Component

**Outcome CS-10-2:**

**Outcome**

CS-10-2-1      Component

## **Appendix A:** GLOSSARY for COMPUTER SCIENCE STANDARDS—

**Abstraction** (Process): The process of reducing complexity by focusing on the main idea. By hiding details irrelevant to the question at hand and bringing together related and useful details, abstraction reduces complexity and allows one to focus on the problem. (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. [MDESE, 2016]

**Accessibility** The design of products, devices, services, or environments for people who experience disabilities. Accessibility standards that are generally accepted by professional groups include the Web Content Accessibility Guidelines (WCAG) 2.0 and Accessible Rich Internet Applications (ARIA) standards. [Wikipedia]

**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. [Techopedia]

**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. [Techopedia]

**Artifact** Anything created by a human. See computational artifact for the definition used in computer science.

**Application Programming Interface** (API) A set of subroutine definitions, communication protocols, and tools for building software. [Wikipedia]

**Audience** Expected end users of a computational artifact or system.

**Authentication** (verb): The verification of the identity of a person or process. [FOLDOC]

**Authentication Factor(s)** (noun): may include password, face recognition, fingerprints, PIN numbers, biometrics, smartcard, Virtual Private Networking (VPN) and Remote Access Services (RAS), etc.

**Automate** To link disparate systems and software so that they become self-acting or self-regulating. [Ross, 2016]

**Automation** The process of automating.

**Boolean** A type of data or expression with two possible values: true and false. [FOLDOC]

**Bug** An error in a software program. It may cause a program to unexpectedly quit or behave in an unintended manner. [Tech Terms] The process of finding and correcting errors (bugs) is called debugging. [Wikipedia]

**Code** Any set of instructions expressed in a programming language. [MDESE, 2016]

**Comment** A programmer-readable annotation in the code of a computer program added to make the code easier to understand. Comments are generally ignored by machines. [Wikipedia]

**Complexity** The minimum amount of resources, such as memory, time, or messages, needed to solve a problem or execute an algorithm. [NIST/ DADS]

**Component** An element of a larger group. Usually, a component provides a particular service or group of related services. [Tech Terms, TechTarget]

**Computational** Relating to computers or computing methods.

**Computational Artifact** Anything created by a human using a computational thinking process and a computing device. A computational artifact can be, but is not limited to, a program, image, audio, video, presentation, or web page file. [College Board, 2016]

**Computational Thinking** The thought processes involved in formulating a problem and expressing its solutions in such a way that a computer (human or machine) can effectively carry them out.

**Computer** A machine or device that performs processes, calculations, and operations based on instructions provided by a software or hardware program. [Techopedia]

**Computer Science** The study of computing principles, design, and applications (hardware & software); the creation, access, and use of information through algorithms and problem solving, and the impact of computing on society.

**Computing** Any goal-oriented activity requiring, benefiting from, or creating algorithmic processes. [MDESE, 2016]

**Computing Device** A physical device that uses hardware and software to receive, process, and output information. Computers, mobile phones, and computer chips inside appliances are all examples of computing devices. [CSTA, 2016]

**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. [CSTA, 2016]

**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. [MDESE, 2016] (A conditional could refer to a conditional statement, conditional expression, or conditional construct.)

**Configuration** (process): Defining the options that are provided when installing or modifying hardware and software or the process of creating the configuration (product). [TechTarget] (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. [TechTarget]

**Connection** A physical or wireless attachment between multiple computing systems, computers, or computing devices. [CSTA]

**Connectivity** A program's or device's ability to link with other programs and devices. [Webopedia]

**Control** (in general) The 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. [CSTA, 2016]

**Control Structure** A programming (code) structure that implements control. Conditionals and loops are examples of control structures. [CSTA, 2016]

**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. [NCSS, 2013]

**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. [TechTarget]

**Data** Information that is collected and used for reference or analysis. Data can be digital or nondigital and can be in many forms, including numbers, text, show of hands, images, sounds, or video. [CAS, 2013; Tech Terms]

**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. [TechTarget]

**Data Type** A classification of data that is distinguished by its attributes and the types of operations that can be performed on it. Some common data types are integer, string, Boolean (true or false), and floating-point. [CSTA, 2016]

**Debugging** The process of finding and correcting errors (bugs) in programs. [MDESE, 2016]

**Decompose** To break down into components. [MDESE, 2016]

**Decomposition** Breaking down a problem or system into components. [MDESE, 2016]

**Device** A unit of physical hardware that provides one or more computing functions within a computing system. It can provide input to the computer, accept output, or both. [Techopedia]

**Document / Documentation** written text or illustration that accompanies computer software or is embedded in the source code. It either explains how it operates or how to use it, and may mean different things to people in different roles [Wikipedia]

**Digital** A characteristic of electronic technology that uses discrete values, generally 0 and 1, to generate, store, and process data. [Techopedia]



**Digital Citizenship** The norms of appropriate, responsible behavior with regard to the use of technology. [MDESE, 2016]

**Efficiency** A measure of the amount of resources an algorithm uses to find an answer. It is usually expressed in terms of the theoretical computations, the memory used, the number of messages passed, the number of disk accesses, etc. [NIST/DADS]

**Encapsulation** The technique of combining data and the procedures that act on it to create a type. [FOLDOC]

**Encryption** The conversion of electronic data into another form, called ciphertext, which cannot be easily understood by anyone except authorized parties. [TechTarget]

**End User (or User)** A person for whom a hardware or software product is designed (as distinguished from the developers). [TechTarget]

**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. [TechTarget]

**Event Handler** A procedure that specifies what should happen when a specific event occurs. [CSTA, 2016]

**Execute** To carry out (or “run”) an instruction or set of instructions (program, app, etc.). [FOLDOC]

**Execution** The process of executing an instruction or set of instructions. [FOLDOC]

**Hardware** The physical components that make up a computing system, computer, or computing device. [MDESE, 2016]

**Hierarchy** An organizational structure in which items are ranked according to levels of importance. [TechTarget]

**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. [TechTarget]

**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. [Techopedia]

**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. [Oxford] DRAFT Document for Public Input Page 200 2019 Computer Science Standards [edu.wyoming.gov/standards](http://edu.wyoming.gov/standards)

**Input** (verb): The signals or instructions sent to a computer. [Techopedia]; (noun): A device or component that allows information to be given to a computer [code.org]

Integrity The overall completeness, accuracy, and consistency of data. [Techopedia] Internet The global collection of computer networks and their connections, all using shared protocols to communicate. [CAS, 2013] Interactive Involving the repeating of a process with the aim of approaching a desired goal, target, or result. [MDESE, 2016] Loop A programming structure that repeats a sequence of instructions as long as a specific condition is true. [Tech Terms] Memory Temporary storage used by computing devices. [MDESE, 2016] Model A representation of some part of a problem or a system. [MDESE, 2016] Note: This definition differs from that used in science. Modularity The characteristic of a software/web application that has been divided (decomposed) into smaller modules. An application might have several procedures that are called from inside its main procedure. Existing procedures could be reused by recombining them in a new application. [Techopedia] 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. [Techopedia] 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. [CSTA, 2016] Operation An action, resulting from a single instruction, that changes the state of data. [Free Dictionary] Output Any device or component that receives information from a computer [Code.org] Packet The unit of data sent over a network. [Tech Terms] Password A password is a string of characters used to verify the identity of a user during the authentication process. Password is an example of one authentication factor. [TechTarget] 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. [MDESE, 2016] Piracy The illegal copying, distribution, or use of software. [TechTarget] Procedure An independent code module that fulfills some concrete task and is referenced within a larger body of program code. The fundamental role of a procedure is to offer a single point of reference for some small goal or task that the developer or programmer can trigger by invoking the procedure itself. [Techopedia] In this framework, 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. [Oxford] Program (noun): A set of instructions that the computer executes to achieve a particular objective. [MDESE, 2016]; (verb): To produce a program by programming. Programming The craft of analyzing problems and designing,

writing, testing, and maintaining programs to solve them. [MDESE, 2016] Protocol The special set of rules used by endpoints in a telecommunication connection when they communicate. Protocols specify interactions between the communicating entities. [TechTarget] DRAFT Document for Public Input Page 201 2019 Computer Science Standards edu.wyoming.gov/standards APPENDIX A: GLOSSARY for COMPUTER SCIENCE STANDARDS—page 5 of 5 Prototype A prototype is an early sample, model, or release of a product built to test a concept or process or to act as a thing to be replicated or learned from. [Wikipedia] Redundancy A system design in which a component is duplicated, so if it fails, there will be a backup. [TechTarget] Reliability Consistently produces the same results, preferably meeting or exceeding its requirements. [FOLDOC] Remix The process of creating something new from something old. Originally a process that involved music, remixing involves creating a new version of a program by recombining and modifying parts of existing programs, and often adding new pieces, to form new solutions. [Kafai & Burke, 2014] Router A device or software that determines the path that data packets travel from source to destination. [TechTarget] Scalability The capability of a network to handle a growing amount of work or its potential to be enlarged to accommodate that growth. [Wikipedia] Simulate To imitate the operation of a real-world process or system. Simulation Imitation of the operation of a real-world process or system. [MDESE, 2016] Software Programs that run on a computing system, computer, or other computing device. Storage (noun): 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. [FOLDOC] storage (verb): A process through which digital data is saved within a data storage device by means of computing technology. Storage is a mechanism that enables a computer to retain data, either temporarily or permanently. [Techopedia] String A sequence of letters, numbers, and/or other symbols. A string might represent, for example, a name, address, or song title. Some functions commonly associated with strings are length, concatenation, and substring. [TechTarget] Structure A general term used in the framework to discuss the concept of encapsulation without specifying a particular programming methodology. Switch A high-speed device that receives incoming data packets and redirects them to their destination on a local area network (LAN). [Techopedia] System A collection of elements or components that work together for a common purpose. [TechTarget] See also the definition for 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. [STF] Topology The physical and logical configuration of a network; the arrangement of a network, including its nodes

and connecting links. A logical topology is the way devices appear connected to the user. A physical topology is the way they are actually interconnected with wires and cables. [PCMag] 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. [Techopedia, TechTarget] Variable A symbolic name that is used to keep track of a value that can change while a program is running. Variables are not just used for numbers; they can also hold text, including whole sentences (strings) or logical values (true or false). A variable has a data type and is associated with a data storage location; its value is normally changed during the course of program execution. [CAS, 2013; Techopedia] Note: This definition differs from that used in math.