2023 Computer Science

Wyoming Content & Performance Standards (WYCPS) with Performance Level Descriptors (PLDs)

Effective - July 17, 2024

To be Fully Implemented in Districts by the Beginning of School Year 2026-27

Rationale:

Computing is fundamental to understanding and participating in an increasingly technological society, and is essential for every Wyoming student to learn as part of a modern education. Computer science is a subject that provides students with a critical lens for interpreting the world around them and challenges them to explore how computing and technology can expand Wyoming's impact on the world.

The standards present here provide the necessary foundation for local school district decisions about curriculum, instruction, and assessment. Implementation of these standards will better prepare Wyoming high school graduates for the rigors of college and/or career. In turn, Wyoming employers will be able to hire workers with a strong foundation in Computer Science—both in specific content areas and in critical thinking and inquiry-based problem solving.

Organization of the Standards:

Standard Code: Grade.Domain.Subconcept.Standard#
Key: 2.CS.HS.01 = 2nd Grade.Computing Systems.Hardware and Software.Standard #1

Domain:

The core concepts to be studied in computer science are as follows: 1) Computing Systems; 2) Networks and the Internet; 3) Data and Analysis; 4) Algorithms and Programming; and 5) Impacts of Computing.

Standards Expectations and Definitions:

The State Board of Education designated the expectations for Computer Science as seen in the table at the beginning of the following page. All students in grades K-5 are expected to be educated on the Content Standards. All students in grades 6-12 who elect to take courses aligned to the Computer Science WYCPS must be instructed on the Content Standards and assessed on the Performance Standards through the District's Assessment System. The terms found in this table are further defined below the table.

Gr.	Math	Science	Health & Safety	PE	Fine & Performing Arts	Computer Science	World Languages & Cultures
K-2	Content & Performance Standards	Content Standards	Content Standards	Content Standards	Content Standards	Content Standards	Content Standards
3-5	Content & Performance Standards	Content & Performance Standards	Content Standards	Content Standards	Content Standards	Content Standards	Content & Performance Standards **Elective**
6-8	Content & Performance Standards	Content & Performance Standards	Content Standards	Content Standards	Content & Performance Standards **Elective**	Content & Performance Standards **Elective**	
9-12	Content & Performance Standards	Content & Performance Standards	Content & Performance Standards	Content & Performance Standards	Content & Performance Standards **Elective**	Content & Performance Standards **Elective**	

Content Standards:
Content Standards define the content knowledge and skills students are expected to know and be able to do by the end of the grade band. They are built foundationally and then in learning progressions. They do not dictate what methodology or instructional materials should be used, nor how the material is delivered. In this standards document, you will find these are broken out into grade bands (K-2), (3-5), (6-8), and (9-12). Schools have local control on how to map out the curriculum across these grade bands.

Performance Level Descriptors (PLDs):
Performance Level Descriptors (PLDs) describe the performance expectations of students for each of the four (4) performance level categories: Advanced, Proficient, Basic, and Below Basic. These are a description of what students within each performance level are expected to know and be able to do.

Performance Standards:
Performance Standards are the standards all students are expected to learn and be assessed on through the district assessment system by the end of the grade band. They specify the specific degree of understanding or demonstration of the knowledge and/or skills at the proficient level. As such, they employ clear action verbs and describe "how good is good enough."

Districts and teachers are expected to give students multiple opportunities to demonstrate proficiency on the Performance Standards through the District Assessment System (DAS) and provide appropriate supports for student success. In the secondary level, only students electing to take a course aligned to these standards need to be assessed in the DAS.

Elective:
Elective means all students must be offered the opportunity to take content area instruction within the indicated grade level or grade band. Districts may choose how to offer elective coursework, but all Content and Performance Standards must be included in a series of courses or the educational program. For this content area, the standards in grades 6-12 are elective.

Plugged in:
This symbol designates when a standard may require hardware, software, or both in order to fully address the intent of the standard.

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.

Computer Science (CS) Practices:
There are seven (7) CS Practices that can be used as the standards are taught and measured. The seven (7) CS Practices are listed below:

Practice 1. Fostering an Inclusive Computing Culture
Practice 2. Collaborating Around Computing
Practice 3. Recognizing and Defining Computational Problems
Practice 4. Developing and Using Abstractions
Practice 5. Creating Computational Artifacts
Practice 6. Testing and Refining Computational Artifacts
Practice 7. Communicating About Computing

Resources/References:

- These Computer Science standards were selected from the 2020 Computer Science Content and Performance Standards during the SBE Standards Reduction effort.
- K-12 Computer Science Framework, (2016). Retrieved from http://k12cs.org/. [Ch. 5 Practices]
- Computer Science Teachers Association (CSTA) K-12 Standards, Revised 2017. Retrieved from https://csteachers.org/k12standards/interactive/.

Grade K-2 Computer Science Content Standards

Introduction:
K-2 Students may be most familiar with touch devices. These students may not yet understand the use of computing devices beyond playing games. They may have emerging problem-solving skills and introductory level sequencing abilities, but their understanding of programming concepts may be limited.

Computing Systems

Devices (D), Hardware & Software (HS), and Troubleshooting (T)

2.CS.HS.01 Demonstrate and describe the function of common components of computing systems (hardware and software) (e.g., use a browser, search engine). [Practice 7.2 Communicating About Computing]

Network & The Internet

Network, Communication, & Organization (NCO) and Cybersecurity (C)

2.NI.C.01 Explain what authentication factors (e.g., login) are, why we use them, and apply authentication to protect devices and information (personal and private) from unauthorized access. [Practice 7.3 Communication About Computing]

Data Analysis

No standards exist for this domain for K-2.

Algorithms & Programming

Algorithms (A), Variables (V), Control (C), Modularity (M), and Program Development (PD)

2.AP.A.01 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). [Practice 4.4 Developing and Using Abstractions]

2.AP.C.01 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. [Practice 5.2 Creating Computational Artifacts]

2.AP.PD.03 Independently and collaboratively debug (identify and fix errors) programs using a programming language. [Practice 6.2 Testing and Refining Computational Artifacts]

Impacts of Computing

Culture (C) and Social Interactions (SI)

2.IC.SI.01 Practice grade-level appropriate behavior and responsibilities while participating in an online community. Identify and report inappropriate behavior. [Practice 2.1 Collaborating Around Computing]

3-5 Computer Science Content Standards

Introduction:
Throughout grades 3-5, students engage in creative applications of Computer Science concepts and practices introduced in K-2. By the end of fifth grade, students will build upon their previous understanding of algorithms, programming (coding), networks, and the Internet. In addition, students will create, modify, and troubleshoot increasingly complex programs for a variety of purposes. Students will be able to explain cultural, social, and ethical impacts of computing.

Computing Systems

Devices (D), Hardware & Software (HS), and Troubleshooting (T)

5.CS.HS.01 Model how information is translated, transmitted, and processed in order to follow through hardware and software to accomplish tasks. [Practice 4.4 Developing and Using Abstractions]

Network & The Internet

Network, Communication, & Organization (NCO) and Cybersecurity (C)

5.NI.C.01 Discuss real-world cybersecurity problems and identify and implement appropriate strategies for how personal information can be protected. [Practice 3.1 Recognizing and Defining Computational Problems]

Data Analysis

Storage (S), Collection, Visualization, & Transformation (CVT), and Inference & Models (IM)

5.DA.S.01 Justify the format and location for storing data based on sharing requirements and the type of information (e.g., images, videos, text). [Practice 4.2 Developing and Using Abstractions]

Algorithms & Programming

Algorithms (A), Variables (V), Control (C), Modularity (M), and Program Development (PD)

5.AP.A.01 Using grade appropriate content and complexity, compare and refine multiple algorithms for the same task and determine which is the most appropriate. [Practice 3.3 Recognizing and Defining Computational Problems] [Practice 6.3 Testing and Refining Computational Artifacts]

5.AP.C.01 Using grade appropriate content and complexity, create programs that include sequences, events, loops, and conditionals, both individually and collaboratively. [Practice 5.2 Creating Computational Artifacts]

Impacts of Computing

Culture (C), Social Interactions (SI), and Safety, Law, and Ethics (SLE)

5.IC.SI.02 Practice grade-level appropriate behavior and responsibilities while participating in an online community. Identify and report inappropriate behavior. [Practice 2.1 Collaborating Around Computing]

6-8 Computer Science Content & Performance Standards

Introduction:
Throughout grades 6-8, students continue to develop their understanding of algorithms and programming (coding). Students work collaboratively and independently to create and modify increasingly complex programs for a variety of purposes introduced in grades 3-5.

Computing Systems

Devices (D), Hardware & Software (HS), and Troubleshooting (T)

8.CS.HS.01 Design and refine a project that combines hardware and software components to collect and exchange data. [Practice 5.1 Creating Computational Artifacts]

- The Advanced student, in addition to meeting the Proficient Level, demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., design a project that combines hardware and software components to collect and exchange data that affects the world around them, refine a project multiple times that combines hardware and software components to collect and exchange data to address real world usage).
- The Proficient student:
  - Designs a project that combines hardware and software components to collect and exchange data.
  - Refines a project that combines hardware and software components to collect and exchange data.
- The Basic student:
  - Describes how hardware and software components collect and exchange data, but cannot design a project, **and/or**
  - Creates a project that combines hardware and software components to collect and exchange data but cannot refine.
- The Below Basic student provides little to no evidence in addressing the expectation(s).

Network & The Internet

Network, Communication, & Organization (NCO) and Cybersecurity (C)

8.NI.C.01 Critique physical and digital procedures that could be implemented to protect electronic data/information. [Practice 7.3 Communicating About Computing]

- The Advanced student, in addition to meeting the Proficient Level, demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., explain the impacts of hacking, ransomware, scams, and ethical/legal concerns; compare the advantages and disadvantages of multiple methods of encryption to model the secure transmission of information).
- The Proficient student critiques physical and digital procedures that could be implemented to protect electronic data/information.
- The Basic student:
  - Lists physical and digital procedures that could be implemented to protect electronic data/information, **and/or**
  - Describes multiple methods of encryption used to secure data.
- The Below Basic student provides little to no evidence in addressing the expectation(s).

Data Analysis

Storage (S), Collection, Visualization, & Transformation (CVT), and Inference & Models (IM)

8.DA.CVT.01 Using computational tools, transform collected data to make it more useful and reliable. [Practice 6.3 Testing and Refining Computational Artifacts]

- The Advanced student, in addition to meeting the Proficient Level, demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., error checking input during data collection process, export data to another format).
- The Proficient student uses computational tools to:
  - Transform data to improve reliability by removing errors.
  - Highlight or expose relationships in the data.
- The Basic student:
  - Explores a variety of computational tools and the content of their data.
  - Uses computational tools to collect data.
- The Below Basic student provides little to no evidence in addressing the expectation(s).

Algorithms & Programming

Algorithms (A), Variables (V), Control (C), Modularity (M), and Program Development (PD)

8.AP.V.01 Using grade appropriate content and complexity, create clearly named variables that represent different data types and perform operations on their values. [Practice 5.1 & 5.2 Creating Computational Artifacts]

- The Advanced student, in addition to meeting the Proficient Level, demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., explain types of errors that can occur if improper data types are used in operations, understand structures or classes can contain multiple data types).
- The Proficient student:
  - Creates clearly named variables.
  - Creates variables that represent different data types.
  - Performs operations on the values of variables.
- The Basic student:
  - Recognizes that variables can represent different data types, **and/or**
  - Can create a variable, **and/or**
  - Can perform operations on the values of variables.
- The Below Basic student provides little to no evidence in addressing the expectation(s).

8.AP.C.01 Using grade appropriate content and complexity, design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. [Practice 5.1 & 5.2 Creating Computational Artifacts]

- The Advanced student, in addition to meeting the Proficient Level, demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., multiple examples of nested loops and compound conditions in a program, evidence of efficient code, clear documentation).
- The Proficient student designs and iteratively develops programs that include:
  - Nested loops.
  - Compound conditionals.
- The Basic student designs and iteratively develops programs that:
  - Use simple loops.
  - Use simple conditionals.
- The Below Basic student provides little to no evidence in addressing the expectation(s).

8.AP.M.01 Using grade appropriate content and complexity, decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. [Practice 3.2 Recognizing and Defining Computational Problems]

- The Advanced student, in addition to meeting the Proficient Level, demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., create procedures with multiple parameters and/or return values)
- The Proficient student decomposes problems and subproblems into parts for the design, implantation, and review of programs.
- The Basic student:
  - Recognizes the inefficiency of repetition in programming, **and/or**
  - Recognizes the organizational, readability and labor-saving advantages of code reuse.
- The Below Basic student provides little to no evidence in addressing the expectation(s).

8.AP.PD.04 Using grade appropriate content and complexity, document programs in order to make them easier to follow, test, and debug. [Practice 7.2 Communicating About Computing]

- The Advanced student, in addition to meeting the Proficient Level, demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., seek open source libraries to include in their program, seek feedback from a wide audience).
- The Proficient student documents programs in order to make them easier to follow, test, and debug.
- The Basic student:
  - Recognizes the advantage of using existing code.
  - Recognizes reasons for testing and refining programs.
  - Recognizes the advantage of documenting programs.
  - Recognizes the role of using feedback.
- The Below Basic student provides little to no evidence in addressing the expectation(s).

Impacts of Computing

Culture (C), Social Interactions (SI), and Safety, Law, and Ethics (SLE)

8.IC.C.01 Describe impacts associated with computing technologies that affect people's everyday activities and career options. [Practice 7.2 Communicating About Computing]

- The Advanced student, in addition to meeting the Proficient Level, demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., devises solutions to solve issues of bias in accessibility, reduce negative impacts of computing technology in everyday life).
- The Proficient student:
  - Describes impacts associated with computing technologies that affect people's everyday activities.
  - Describes impacts associated with computing technologies that affect people's career options.
- The Basic student:
  - Lists computing technologies that affect people's everyday activities, **and/or**
  - Lists computing technologies that affect people's career options, **and/or**
  - Identifies an accessibility issue related to technology.
- The Below Basic student provides little to no evidence in addressing the expectation(s).

8.IC.SI.02 Practice grade-level appropriate behavior and responsibilities while participating in an online community. Identify and report inappropriate behavior. [Practice 2.1 Collaborating Around Computing] [Practice 7.3 Communicating About Computing]

- The Advanced student, in addition to meeting the Proficient Level, demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., moderate, model appropriate behavior, and facilitate discussions in an online community).
- The Proficient student:
  - Collaborates using tools to connect with peers when creating a computational artifact.
  - Practices grade-level appropriate behavior and responsibilities while participating in an online community.
  - Identifies and reports inappropriate behavior while participating in an online community, when applicable.
- The Basic student:
  - Collaborates with peers using a tool in an attempt to create a computational artifact.
  - Intermittently collaborates and behaves within an online community.
- The Below Basic student provides little to no evidence in addressing the expectation(s).

8.IC.SLE.02 Using grade level appropriate content and complexity, discuss the legal, social, and ethical impacts associated with software development and use, including both positive and malicious intent. [Practice 1.1 Fostering an Inclusive Computing Culture] [Practice 7.2 Communicating About Computing]

- The Advanced student demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard. (e.g. research and report on current legal, social, and ethical worldwide trends in software development; construct an argument for or against the use of personal data by commercial entities or government).
- The Proficient student:
  - Describes tradeoffs between allowing information to be public and keeping information private and secure, **and**
  - With regard to positive and malicious intent:
    - » Discusses the legal impacts associated with software development and use,
    - » Discusses the social impacts associated with software development and use,
    - » Discusses the ethical impacts associated with software development and use.
- The Basic student:
  - Lists reasons for allowing information to be public and/or keeping information private and secure **and/or**
  - With regard to positive and/or malicious intent:
    - » Names the legal impacts associated with software development and use,
    - » Names the social impacts associated with software development and use,
    - » Names the ethical impacts associated with software development and use.
- The Below Basic student provides little to no evidence in addressing the expectation(s).

9-12 Computer Science Content & Performance Standards

Introduction:
In high school, students will continue to develop their knowledge of computing systems, their components, and how systems interact. Students will use their understanding about the basic principles of computation, that algorithms describe a step-by-step solution to a problem, that programs are algorithms written in a language that a computer can understand, and that the solution to many problems can be described as a program. A solid foundation of algebraic concepts is important for success in high school computer science courses. Students will expand their ability to identify patterns and create algorithms that can model the observed patterns.

Computing Systems

Devices (D), Hardware & Software (HS), and Troubleshooting (T)

L1.CS.HS.01 Explain the interactions between application software, system software, and hardware layers. [Practice 4.1 Developing and Using Abstractions]

- The Advanced student, in addition to meeting the Proficient Level, demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., student demonstrates knowledge of specific, advanced terms for computer architecture, such as BIOS, kernel, or bus).
- The Proficient student:
  - Identifies the interactions between application software, system software, and hardware layers.
  - Defines the interactions between application software, system software, and hardware layers.
  - Explains the interactions between application software, system software, and hardware layers. For example, text editing software interacts with the operating system to receive input from the keyboard, convert the input to bits for storage, and interpret the bits as readable text to display on the monitor.
- The Basic student:
  - Identifies application software, system software, and hardware layers.
  - Defines application software, system software, and hardware layers.
- The Below Basic student provides little to no evidence in addressing the expectation(s).

Network & The Internet

Network, Communication, & Organization (NCO) and Cybersecurity (C)

L1.NI.C.03 Compare various security measures, considering trade-offs between the usability and security of a computing system. [Practice 6.3 Testing and Refining Computational Artifacts]

- The Advanced student, in addition to meeting the Proficient Level, demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., discuss security policies that are in place that present a tradeoff between usability and security).
- The Proficient student compares two or more security measures, considering trade-offs between the usability and security of a computing system.
- The Basic student:
  - Identifies various security measures.
  - Defines various security measures.
- The Below Basic student provides little to no evidence in addressing the expectation(s).

Data Analysis

Storage (S), Collection, Visualization, & Transformation (CVT), and Inference & Models (IM)

L1.DA.CVT.01 Create interactive data representations using software tools to help others better understand real-world phenomena (e.g., paper surveys and online data sets). [Practice 4.4 Developing and Using Abstractions]

- The Advanced student, in addition to meeting the Proficient Level, demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., research emerging visualization techniques and use them to create new data representations).
- The Proficient student creates appropriate interactive data representations using software tools to help others better understand real-world phenomena.
- The Basic student creates, with errors, interactive data representations using software tools.
- The Below Basic student provides little to no evidence in addressing the expectation(s).

Algorithms & Programming

Algorithms (A), Variables (V), Control (C), Modularity (M), and Program Development (PD)

L1.AP.A.01 Create a prototype that uses algorithms (e.g., searching, sorting, finding shortest distance) to provide a possible solution for a real-world problem relevant to the student. [Practice 5.2 Creating Computational Artifacts]

- The Advanced student, in addition to meeting the Proficient Level, demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., student generated problem).
- The Proficient student creates a prototype that uses appropriate algorithms (e.g., searching, sorting, finding shortest distance) to provide a possible solution for a real-world problem relevant to the student.
- The Basic student creates a prototype that uses an algorithm (e.g., searching, sorting, finding shortest distance) to provide a possible solution for a real-world problem relevant to the student.
- The Below Basic student provides little to no evidence in addressing the expectation(s).

L1.AP.A.02 Describe how artificial intelligence algorithms drive many software and physical systems. [Practice 7.2 Communicating About Computing]

- The Advanced student, in addition to meeting the Proficient Level, demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., student discusses different types of artificial intelligence algorithms).
- The Proficient student describes how artificial intelligence algorithms drive many software and physical systems.
- The Basic student describes how artificial intelligence algorithms drive a software system or physical system.
- The Below Basic student provides little to no evidence in addressing the expectation(s).

L1.AP.C.03 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions. [Practice 5.2 Creating Computational Artifacts]

- The Advanced student, in addition to meeting the Proficient Level, demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., using multiple user interface components).
- The Proficient student:
  - Designs and iteratively develops a computational artifact that uses events to initiate instructions for:
    - » Practical intent,
    - » Personal expression, **or**
    - » Societal issues.
- The Basic student designs a computational artifact that uses events to initiate instructions.
- The Below Basic student provides little to no evidence in addressing the expectation(s).

L1.AP.M.01 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. [Practice 3.2 Recognizing and Defining Computational Problems]

- The Advanced student, in addition to meeting the Proficient Level, demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., an appropriate class hierarchy).
- The Proficient student decomposes problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
- The Basic student partially decomposes problems into smaller components.
- The Below Basic student provides little to no evidence in addressing the expectation(s)

L1.AP.PD.01 Plan and develop programs by analyzing a problem and/or process, developing and documenting a solution, testing outcomes, and adapting the program for a variety of users. [Practice 5.1 Creating Computational Artifacts]

- The Advanced student, in addition to meeting the Proficient Level, **independently** plans and develops programs by:
  - Analyzing a problem and/or process.
  - Developing and documenting a solution.
  - Testing outcomes.
  - Adapting the program for a variety of users.
- The Proficient student plans and develops programs by:
  - Analyzing a problem and/or process.
  - Developing and documenting a solution.
  - Testing outcomes.
  - Adapting the program for a variety of users.
- The Basic student, with instructor support, plans and develops programs by:
  - Analyzing a problem and/or process.
  - Developing and documenting a solution.
  - Testing outcomes.
- The Below Basic student provides little to no evidence in addressing the expectation(s).

L1.AP.PD.04 Design and develop computational artifacts, working in team roles, using collaborative tools. [Practice 2.4 Collaborating Around Computing]

- The Advanced student, in addition to meeting the Proficient Level, demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard. As programs grow more complex, the choice of resources that aid program development becomes increasingly important and should be made by the students.
- The Proficient student designs and develops computational artifacts while working in teams and using collaborative tools.
- The Basic student:
  - Designs computational artifacts using collaborative tools.
  - Develops computational artifacts using collaborative tools.
- The Below Basic student provides little to no evidence in addressing the expectation(s).

Impacts of Computing

Culture (C), Social Interactions (SI), and Safety, Law, and Ethics (SLE)

L1.IC.SLE.03 Evaluate the social and economic implications of privacy in the context of safety, law, or ethics. [Practice 7.3 Communicating About Computing]

- The Advanced student, in addition to meeting the Proficient Level, demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.
- The Proficient student evaluates the following implications of privacy in the context of safety, law, or ethics:
  - Social implications
  - Economic implications
- The Basic student provides examples of the following implications of privacy in the context of safety, law, or ethics:
  - Social implications
  - Economic implications
- The Below Basic student provides little to no evidence in addressing the expectation(s).

L1.IC.SLE.04 Using grade level appropriate content and complexity, discuss the legal, social, and ethical impacts associated with software development and use, including both positive and malicious intent. [Practice 1.1 Fostering an Inclusive Computing Culture] [Practice 7.2 Communicating About Computing]

- The Advanced student, in addition to meeting the Proficient Level, demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.
- The Proficient student, when considering both positive and malicious intent regarding software development and use, discusses the:
  - Legal impacts
  - Social impacts
  - Ethical impacts
- The Basic student, when considering both positive and malicious intent regarding software development and use, provides examples of the:
  - Legal impacts, **or**
  - Social impacts, **or**
  - Ethical impacts
- The Below Basic student provides little to no evidence in addressing the expectation(s).