# What to Look for in the 2018 Computer Science Standards

The following document explains each aspect of the new voluntary high school Computer Science standards. The purpose of the document is to help educators interpret the terminology of the standards document and provide examples of the application of the standards into the classroom.

## Standards

Firstly, it is noteworthy that the Computer Science standards were designed to be implemented in all high school grades throughout computer science courses. All three standards may be found in one course, or perhaps, just one standard might be implemented in a high school course. The standards, essentially distilled from the K-12 Computer Science Framework and CSTA standards, provide primary goals for computer science content. The following are the three high school computer science standards.

1. **Computational Thinking** Includes concepts related to the use of algorithms and data in different ways to generate new knowledge and articulate solutions to real world problems.

Computational thinking will likely be included in all computer science courses.

1. **Computing Systems and Networks** Includes concepts related to the development and communication between software and hardware, and systems thinking around data protection and recovery.
2. **Computer Programming** Includes concepts related to creating computer programs and applications, working collaboratively to engage in client-based problem solving, and internet security.

## Prepared Graduate Statements

The Prepared Graduate Statements offer educators a final goal for the development of knowledge and skills throughout high school computer science courses. By the time students are seniors, graduating, and moving on to work or college, they should know how to:

Computational Thinking

1. Develop, utilize, and evaluate algorithms, to model and solve problems.
2. Systematically analyze a problem using decomposition and abstraction to formulate a solution.
3. Represent and analyze data in order to generate new knowledge and capability.

Computing Systems and Networks

1. Use systems thinking to describe networks and common software and hardware components.
2. Develop systems solutions from a set of specifications to complete a design process.
3. Recognize and analyze security concepts.

Computer Programming

1. Design and create programs, individually and collaboratively, for a variety of disciplines.
2. Create computational artifacts that consider security from tampering, malicious or otherwise.

## Grade Level Expectations

With knowledge of the “big picture”, the standards and the prepared graduate statements for computer science, educators can begin to identify specific learning objectives for each course. The broad goals of the standards turn into learning objectives when each standard is delineated into Grade Level Expectations (GLE’s).

**Computational Thinking – GLE’s**

Focusing on how to solve problems with computational artifacts requires understanding of concepts such as algorithms, decomposition, abstraction, pattern recognition, data representation, and data analysis.

**Computing Systems and Networks – GLE’s**

Knowledge of computing systems and networks is essential to developing useful and effective computational solutions. Concepts, such as, hardware, software, systems thinking, design thinking, cybersecurity, data recovery, and data protection are the primary learning objectives for studying Computing Systems and Networks.

**Computer Programming – GLE’s**

Programming is the means to creating computational solutions. In the third standard, Computer Programming, the following GLE’s topics can help guide course design:

* design process
* security and licensing
* computational problems
* human problems
* personal, ethical, social, and cultural implications of computing
* collaboration and the team design process
* client-based solutions

The grade level expectations can provide educators with a list of more detailed thinking skills, knowledge about how computers work, and important aspects to learning any computer language that can be targeted objectives in high school computer science courses.

## Evidence Outcomes

The evidence outcomes (EO’s) are measurable descriptions of the grade level expectations. Course objectives are more specific than standards and provide educational goas. Outcomes identify the measurable evidence of learning the objective.

An example of the progress from content area to standard to prepared graduate statement to objective (GLE) to outcome (EO) is detailed below.

Computer Science

High School, Standard 3. Computer Programming

Prepared Graduates:

8. Create computational artifacts that consider security from tampering, malicious or otherwise.

Grade Level Expectation:

6. Security and software licensing can present constraints and restrictions in computational design and development.

**Evidence Outcomes**

Students Can:

a. Describe how software licensing influences program development.

b. Investigate and develop solutions that discourage online software piracy.

c. Explore and integrate security measures such as encryption, authentication, and verification strategies to secure developed computer programs.

d. Research and abide by intellectual property laws & patents.

Using the EO’s listed above, teachers could create rubrics, make test questions, assign short response topics, create projects, and through summative and formative methods assess student attainment.

## Academic Context and Connections

**Colorado Essential Skills:**

The Colorado Department of Education asked each standards review committee to apply the [Colorado Essential Skills](https://www.cde.state.co.us/standardsandinstruction/essentialskills) to all standards as required by SB-212. In each GLE section, applications of Colorado Essential Skills are offered as examples. Consider the following Colorado Essential Skills form the same standard – Computer Programming – and GLE – (6. Security and software licensing can present constraints and restrictions in computational design and development.) listed above.

Colorado Essential Skills

1. Learn about the steps required for protecting intellectual rights of your computational artifact. (Entrepreneurial Skills: Critical thinking/problem solving; Inquiry and analysis; Risk taking)

2. Analyze licensing agreements from a software vendor. (Civic/Interpersonal Skills: Collaboration; Communication; Global/cultural awareness; Civic engagement; Character)

3. Evaluate the benefits of open-source and proprietary software to the developer. (Professional Skills: Career awareness; Information Literacy; Use information/communications technologies; Self-advocacy)

**Elaboration on the GLE:**

The elaboration on the GLE provides greater context around the expectations associated with the standard and provides educators more information on developing and planning curriculum. Educators can also find connections to the CSTA standards within the elaboration. Here is the elaboration on the previous standard and GLE - Computer Programming – and GLE – (6. Security and software licensing can present constraints and restrictions in computational design and development.)

Elaboration on the GLE

1. After finishing a computer program, students should consider how they would potentially distribute their product. Whether they determine to sell it at a price on an app store or distribute it for free, a license of some sort is required and a process for which consumers can access the program. Alternatively, students need to be mindful that pirating occurs and should think about ways they can secure their programs to not be unlawfully distributed such as licensing codes, attachment to connected services, distribute software, etc. Students need to be aware of laws and patents that govern / protect intellectual property (CSTA 3A-AP-20 & 3A-IC-28).

**Computer Science Practices:**

Aligning with the Core Computing Practices described in the K-12 CS Framework and the Practices described in the CSTA standards, the Computer Science Practices, identified on each GLE section, offer a summary or snapshot of student behaviors and thinking skills endemic to the GLE. The following example of Computer Science Practices relates to the standard Computer Programming and GLE – (6. Security and software licensing can present constraints and restrictions in computational design and development.)

Computer Science Practices

1. Recognizing and defining computational problems

2. Testing and refining a computational artifact

3. Communicating about computing