2019 WYOMING COMPUTER SCIENCE

CONTENT & PERFORMANCE STANDARDS

DRAFT FOR PUBLIC COMMENT PERIOD (MAY-JUNE 2019)

BOOT UP
WYOMING

Effective MONTH XX, 2019

TO BE FULLY IMPLEMENTED IN DISTRICTS BY THE BEGINNING OF SCHOOL YEAR 2022-23

INTRODUCTION:

The Wyoming Computer Science Content and Performance Standards (WYCPS) were developed in accordance with Wyoming State Statute W.S. 21-2-304(c). The 2019 Wyoming Computer Science Standards were developed collaboratively through the contributions of the Computer Science Standards Review Committee (CSSRC) which included Wyoming parents, educators, and community members, as well as business members from across the state and nation. The committee's work was informed and guided by initial public input through community forums, as well as input solicited from specific stakeholder groups. Additional appendices and teacher resources, created by the CSSRC, are also available on the WDE website at edu.wyoming.gov/standards.

RATIONALE:

The committee's (CSSRC) vision is that every student in every school has the opportunity to learn computer science. We believe that computing is fundamental to understanding and participating in an increasingly technological society, and it is essential for every Wyoming student to learn as part of a modern education. We see computer science as 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 we (CSSRC) present here provide the necessary foundation for local school district decisions about curriculum, assessment, and instruction. 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.

In grades K-5, the benchmarks are coded to represent supporting benchmarks, priority benchmarks, and enhanced benchmarks. It is the committee's expectation that all students receive instruction for the supporting and priority benchmarks and have an opportunity to demonstrate mastery of the content and performance expectations included in the priority benchmarks. Students may also have the opportunity to receive enrichment through the enhanced benchmarks.

In grades 6-8, the committee (CSSRC) determined the benchmark to be met by the end of this grade-band and also provides suggested progressions, which can be found on Appendix C: Teacher Resource Progression Document. (see Appendices on pg. 4)

In grades 9-12, the committee provides level 1 and level 2 benchmarks. Level 1 benchmarks include introductory skills. The level 2 benchmarks are intended for students who wish to advance their study of Computer Science. All level 1 and level 2 benchmarks are intended to be assessed for students taking courses covering the skills described in the benchmark.

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.

ORGANIZATION OF THE COMPUTER SCIENCE (CS) STANDARDS:

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.

Content Standards

Content standards define what students are expected to know and be able to do throughout their study of computer science. They do not dictate what methodology or instructional materials should be used, nor how the material is delivered.

Benchmarks

Benchmarks are the skills students must master in order to demonstrate proficiency of the content standards throughout the grade band. In grades 9-12, benchmarks are organized into 2 levels. Mostly, Level 1 is intended to represent the introductory level while Level 2 reaches a deeper level.

For the K-5 grade-band, each benchmark is labeled identifying it as a *priority (shaded in gold)*, *supporting*, or *enhanced* benchmark.

- *Priority* Benchmark (Gold) - All students are expected to be instructed on and demonstrate mastery of the content and performance expectations included in these benchmarks.
- *Supporting* Benchmark - All students are expected to be instructed on these standards, taught within the context of the priority standards.
- *Enhanced* Benchmark - Students have an opportunity for enrichment above what all students are expected to know and do as required by the priority benchmarks.

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

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.

WYOMING 2019 COMPUTER SCIENCE DOMAINS & STANDARDS

Computing Systems	Networks & The Internet	Data Analysis	Algorithms & Programming	Impacts of Computing
CS.D—Devices CS.HS—Hardware & Software CS.T—Troubleshooting	NI.NCO—Network Communication & Organization NI.C—Cybersecurity	DA.S—Storage DA.CVT—Collection, Visualization, & Transformation DA.IM—Inference & Models	AP.A—Algorithms AP.V—Variables AP.C—Control AP.M—Modularity AP.PD—Program Development	IC.C—Culture IC.SI—Social Interactions IC.SLE—Safety, Law, & Ethics

Benchmark Code: Grade.Domain.Standard.Benchmark#
Key: 2.CS.D.01 = 2nd Grade.Computing Systems.Devices.Benchmark #1

COMPUTER SCIENCE (CS) PRACTICES:

There are seven (7) CS Practices that can be used as the standards and benchmarks are taught and measured. The seven (7) CS Practices are listed below and are more deeply explored in Appendix A: Descriptions of the CS Practices. (see Appendices 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

APPENDICES - found at edu.wyoming.gov/standards

- APPENDIX A: DESCRIPTIONS OF COMPUTER SCIENCE (CS) PRACTICES
- APPENDIX B: GLOSSARY
- APPENDIX C: TEACHER RESOURCE PROGRESSION DOCUMENT
- APPENDIX D: ADMINISTRATOR K-12 CS STANDARDS OVERVIEW
- APPENDIX E: WYOMING DIGITAL LEARNING GUIDELINES (based on the 2016 ISTE Standards for Students)

RESOURCES / REFERENCES

- K-12 Computer Science Framework, (2016). Retrieved from http://k12cs.org/. [Ch. 5 Practices].
- Computer Science Teachers Association (CSTA), (2017). Retrieved from http://www.csteachers.org/page/standards.

Computer Science | K-2 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.

By the end of 2nd grade, students can:

- Protect and safeguard their information
- Follow and write step-by-step instructions
- Create programs to accomplish tasks
- Work respectfully and responsibly with others in an online environment

Computer Science | 3-5 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.

By the end of 5th grade, students can:

- Model how information is translated, transmitted, and processed
- Identify and implement strategies for protecting personal information
- Justify the format and location for storage
- Create and modify (remix) programs through an iterative process
- Develop, test, and refine digital artifacts
- Work respectfully and responsibly with others in an online environment and discuss the social impact of violating intellectual property rights

DOMAIN KEY	COMPUTING SYSTEMS	NETWORKS & THE INTERNET	DATA & ANALYSIS	ALGORITHMS & PROGRAMMING	IMPACTS OF COMPUTING

COMPUTING SYSTEMS	End of Grade 2	End of Grade 5
DEVICES (CS.D)	**SUPPORTING** **2.CS.D.01** Independently select and use a computing device to perform a variety of tasks for an intended outcome (e.g., create an artifact). **Practice 1.1** Fostering an Inclusive Computing Culture	**(+) ENHANCED** **5.CS.D.01** Independently, describe how internal and external parts of computing devices function to form a system. **Practice 7.2** Communicating About Computing
HARDWARE & SOFTWARE (CS.HS)	**SUPPORTING** **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	**PRIORITY** **5.CS.HS.01** Model how information is translated, transmitted, and processed in order to flow through hardware and software to accomplish tasks. **Practice 4.4** Developing and Using Abstractions
TROUBLESHOOTING	**SUPPORTING** **2.CS.T.01** Recognize computing systems might not work as expected and identify and effectively communicate simple hardware or software problems and implement solutions (e.g., app or program is not working as expected, no sound is coming from the device, caps lock turned on) and discuss problems with peers and adults. **Practice 6.2** Testing and Refining Computational Artifacts **Practice 7.2** Communicating About Computing	**SUPPORTING** **5.CS.T.01** Identify hardware and software problems that may occur during everyday use, then develop, apply, and explain strategies for solving these problems. **Practice 6.2** Testing and Refining Computational Artifacts

NETWORKS & THE INTERNET	End of Grade 2	End of Grade 5
NETWORK COMMUNICATION & ORGANIZATION (NI.NCO)	**SUPPORTING** **2.NI.NCO.01** Identify and describe that computing devices can be connected in a variety of ways (e.g., Bluetooth, Wi-Fi, home and school networks, the internet). **Practice 6.2** Testing and Refining Computational Artifacts	**SUPPORTING** **5.NI.NCO.01** 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. **Practice 4.4** Developing and Using Abstractions
CYBERSECURITY (NI.C)	**PRIORITY** **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** Communicating About Computing	**PRIORITY** **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	End of Grade 2	End of Grade 5
STORAGE (DA.S)	**(+) ENHANCED** **2.DA.S.01** With guidance, develop and modify an organizational structure by creating, copying, moving, and deleting files and folders. **Practice 4.2** Developing and Using Abstractions 🔌	**PRIORITY** **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 🔌
COLLECTION, VISUALIZATION, & TRANSFORMATION (DA.CVT)	**SUPPORTING** **2.DA.CVT.01** With guidance, collect data and independently present the same data in various visual formats. **Practice 4.4** Developing and Using Abstractions **Practice 7.1** Communicating About Computing	**SUPPORTING** **5.DA.CVT.01** Organize and present collected data to highlight relationships and support a claim. **Practice 7.1** Communicating About Computing
INFERENCE & MODELS (DA.IM)	**SUPPORTING** **2.DA.IM.01** With guidance, interpret data and present it in a chart or graph (visualization) in order to make a prediction, with or without a computing device. **Practice 4.1** Developing and Using Abstractions	**SUPPORTING** **5.DA.IM.01** Use data to highlight or propose relationships, predict outcomes, or communicate an idea. **Practice 7.1** Communicating About Computing

ALGORITHMS & PROGRAMMING	End of Grade 2	End of Grade 5
ALGORITHMS (AP.A)	**PRIORITY** **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	**PRIORITY** **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
VARIABLES (AP.V)	**SUPPORTING** **2.AP.V.01** Model the way programs store and manipulate data by using numbers or other symbols to represent information (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). **Practice 4.1** Developing and Using Abstractions	**PRIORITY** **5.AP.V.01** Using grade appropriate content and complexity, create programs that use variables to store and modify data. **Practice 5.2** Creating Computational Artifacts 🔌
CONTROL (AP.C)	**PRIORITY** **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	**PRIORITY** **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 🔌
MODULARITY (AP.M)	**(+) ENHANCED** **2.AP.M.01** Using 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). **Practice 3.2** Recognizing and Defining Computational Problems	**SUPPORTING** **5.AP.M.01** Using grade appropriate content and complexity, decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. **Practice 3.2** Recognizing and Defining Computational Problems
		SUPPORTING **5.AP.M.02** Using 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. **Practice 5.3** Creating Computational Artifacts

ALGORITHMS & PROGRAMMING	End of Grade 2	End of Grade 5
PROGRAM DEVELOPMENT (AP.PD)	**SUPPORTING** **2.AP.PD.01** Develop plans that describe a program's sequence of events, goals, and expected outcomes. **Practice 5.1** Creating Computational Artifacts **Practice 7.2** Communicating About Computing	**PRIORITY** **5.AP.PD.01** Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences. **Practice 6.2** Testing and Refining Computational Artifacts
	SUPPORTING **2.AP.PD.02** Give credit to ideas, creations, and solutions of others while writing and developing programs. **Practice 7.3** Communicating About Computing	**SUPPORTING** **5.AP.PD.02** Using grade appropriate content and complexity, observe intellectual property rights and give appropriate credit when creating or remixing programs. **Practice 5.2** Creating Computational Artifacts **Practice 7.3** Communicating About Computing
	SUPPORTING **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	**SUPPORTING** **5.AP.PD.03** Using grade appropriate content and complexity, test and debug (i.e., identify and fix errors) a program or algorithm to ensure it runs as intended. **Practice 6.1 & 6.2** Testing and Refining Computational Artifacts
	(+) ENHANCED **2.AP.PD.04** 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 a programming language). **Practice 7.2** Communicating About Computing	**SUPPORTING** **5.AP.PD.04** Using grade appropriate content and complexity, describe choices made during program development using code comments, presentations, and demonstrations. **Practice 7.2** Communicating About Computing
		(+) ENHANCED **5.AP.PD.05** Using grade appropriate content and complexity, with teacher guidance, perform varying roles when collaborating with peers during the design, implementation, and review stages of program development. **Practice 2.2** Collaborating Around Computing

IMPACTS OF COMPUTING	End of Grade 2	End of Grade 5
CULTURE (IC.C)	**SUPPORTING** **2.IC.C.01** Describe how people use different types of technologies in their daily work and personal lives. **Practice 3.1** Recognizing and Defining Computational Problems	**(+) ENHANCED** **5.IC.C.01** Give examples and explain how computing technologies have changed the world and express how those technologies influence and are influenced by cultural practices. **Practice 3.1** Recognizing and Defining Computational Problems
		PRIORITY **5.IC.C.02** Develop, test, and refine digital artifacts or devices to improve accessibility and usability for diverse end users. **Practice 1.2** Fostering an Inclusive Computing Culture
SOCIAL INTERACTIONS (IC.SI)		**(+) ENHANCED** **5.IC.SI.01** Seek diverse perspectives for the purpose of improving computational artifacts. **Practice 1.1** Fostering an Inclusive Computing Culture
	PRIORITY **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	**PRIORITY** **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
SAFETY, LAW, & ETHICS (IC.SLE)		**SUPPORTING** **5.IC.SLE.01** Recognize and appropriately use public domain and creative commons media and discuss the social impact of violating intellectual property rights. **Practice 7.3** Communicating About Computing

DOMAIN - KEY	COMPUTING SYSTEMS	NETWORKS & THE INTERNET	DATA & ANALYSIS	ALGORITHMS & PROGRAMMING	IMPACTS OF COMPUTING

COMPUTING SYSTEMS	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
DEVICES (CS.D)	**2.CS.D.01** Independently select and use a computing device to perform a variety of tasks for an intended outcome (e.g., create an artifact).	provides little to no evidence in addressing the expectation(s).	with guidance, uses a computing device to complete assignments or teacher led activities.	regularly uses a computing device to independently: - power on and off devices. - authenticate, when applicable. - open appropriate programs. - complete assignments or teacher led activities.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., can recognize capabilities of multiple devices and can perform similar tasks with them).
	5.CS.D.01 Independently, describe how internal and external parts of computing devices function to form a system.	provides little to no evidence in addressing the expectation(s).	with guidance, describes with some errors how internal and external parts of computing devices function to form a system.	independently describes with few to no errors how internal and external parts of computing devices function to form a system.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., demonstrates on different types of devices).
HARDWARE & SOFTWARE (CS.HS)	**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).	provides little to no evidence in addressing the expectation(s).	with guidance: - identifies hardware components and software applications. - utilizes hardware components and software applications.	can identify and utilize: - a variety of hardware components (e.g., input devices, printers). - software applications (e.g., browsers, apps). - navigation to browser search engines and applications.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., justifies hardware and software choices).

COMPUTING SYSTEMS	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
HARDWARE & SOFTWARE Continued **(CS.HS)**	**5.CS.HS.01** Model how information is translated, transmitted, and processed in order to flow through hardware and software to accomplish tasks.	provides little to no evidence in addressing the expectation(s).	partially models how information is translated, transmitted, and processed in order to flow through hardware and software to accomplish tasks.	accurately models how information is translated, transmitted, and processed in order to flow through hardware and software to accomplish tasks.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., compare and contrast different devices).
TROUBLESHOOTING (CS.T)	**2.CS.T.01** Recognize that computing systems might not work as expected and identify and effectively communicate simple hardware or software problems and implement solutions (e.g., app or program is not working as expected, no sound is coming from the device, caps lock turned on) and discuss problems with peers and adults.	provides little to no evidence in addressing the expectation(s).	- can recognize that computing systems may not work as expected. - with guidance, identifies and effectively communicates simple hardware and software problems. - with guidance, implements solutions to simple hardware or software issues.	- can recognize that computing systems may not work as expected. - identifies and effectively communicates simple hardware and software problems. - implements solutions to simple hardware or software issues.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., helps others in troubleshooting issues, can troubleshoot more complex issues like connectivity or advanced software features).
	5.CS.T.01 Identify hardware and software problems that may occur during everyday use, then develop, apply, and explain strategies for solving these problems.	provides little to no evidence in addressing the expectation(s).	partially: - identifies hardware and software problems that may occur during everyday use. - attempts to solve identified problems, when applicable.	accurately: - identifies hardware and software problems that may occur during everyday use. - develops, applies, and explains strategies for solving identified problems, when applicable.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., develops a troubleshooting guide, helps others with troubleshooting issues efficiently, suggests preventative measures).

NETWORKS & THE INTERNET	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
NETWORK COMMUNICATION & ORGANIZATION (NI.NCO)	**2.NI.NCO.01** Identify and describe that computing devices can be connected in a variety of ways (e.g., Bluetooth, Wi-Fi, home and school networks, the internet).	provides little to no evidence in addressing the expectation(s).	- can identify that computing devices can be connected in a variety of ways. - with guidance, can describe a connectivity option (e.g., Wi-Fi or Bluetooth).	- can identify that computing devices can be connected in a variety of ways. - can describe different connectivity options (e.g., Bluetooth, internet).	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., evaluates the appropriateness of different connectivity options for a variety of tasks).
	5.NI.NCO.01 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.	provides little to no evidence in addressing the expectation(s).	partially models and explains how information is: - broken down into smaller pieces, **and** - transmitted as packets through multiple devices over networks and the internet, **and/or** - reassembled at the destination.	accurately models and explains how information is: - broken down into smaller pieces. - transmitted as packets through multiple devices over networks and the internet. - reassembled at the destination.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., compares and contrasts different connection types).
CYBERSECURITY (NI.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.	provides little to no evidence in addressing the expectation(s).	-identifies what authentication factors are. - with guidance, applies authentication factors to appropriate apps and devices.	- explains what authentication factors are and why we use them. - applies authentication factors to appropriate apps and devices.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., can compare authentication methods, one factor versus two factors).

NETWORKS & THE INTERNET	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
CYBERSECURITY Continued **(NI.C)**	**5.NI.C.01** Discuss real-world cybersecurity problems and identify and implement appropriate strategies for how personal information can be protected.	provides little to no evidence in addressing the expectation(s).	- generally discusses real-world cybersecurity problems, **and/or** - identifies appropriate strategies for how personal information can be protected.	- discusses with specificity real-world cybersecurity problems. - discusses personal consequences of inappropriate use. - identifies and implements appropriate strategies for how personal information can be protected.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., compares and contrasts a variety of approaches to authentication, evaluates current practices).

DATA & ANALYSIS	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
STORAGE **(DA.S)**	**2.DA.S.01** With guidance, develop and modify an organizational structure by creating, copying, moving, and deleting files and folders.	provides little to no evidence in addressing the expectation(s).	while working with a computing device and with guidance: - locates existing files. - opens existing files. - modifies existing files. - saves changes to a file.	with guidance, develops and modifies an organizational structure by: - creating folders. - copying existing folders and files. - moving existing folders and files. - deleting folders and files.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., can independently create organizational structure).
	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).	provides little to no evidence in addressing the expectation(s).	describes the format, location, sharing requirements, or the type of information when storing data.	justifies the format and location for storing data based on sharing requirements and the type of information.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., determines the best file type for a given purpose, suggests strategies to solve a problem, creates a document in a variety of formats, converts files).

DATA & ANALYSIS	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
COLLECTION, VISUALIZATION, & TRANSFORMATION (DA.CVT)	**2.DA.CVT.01** With guidance, collect data and independently present the same data in various visual formats.	provides little to no evidence in addressing the expectation(s).	with guidance: - creates a data set, **and** - presents that data.	- with guidance, creates a data set, **and** - independently presents that data in multiple formats (e.g., as a table and graph or as a table and chart).	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., independently creates and presents their own data sets).
	5.DA.CVT.01 Organize and present collected data to highlight relationships and support a claim.	provides little to no evidence in addressing the expectation(s).	organizes and presents collected data.	organizes and presents collected data to: - highlight comparisons. - highlight relationships. - to support a claim.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., helps others organize collected data, suggests improvements on how to organize collected data to a specific audience).
INFERENCE & MODELS (DA.IM)	**2.DA.IM.01** With guidance, interpret data and present it in a chart or graph (visualization) in order to make a prediction, with or without a computing device.	provides little to no evidence in addressing the expectation(s).	with guidance: - interprets data, **and** - presents it in a chart or graph (visualization).	with guidance: - interprets data. - presents it in a chart or graph (visualization). - makes a prediction based on the data.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., independently perform any of the proficient student steps).
	5.DA.IM.01 Use data to highlight or propose relationships, predict outcomes, or communicate an idea.	provides little to no evidence in addressing the expectation(s).	with guidance, uses data to: - highlight relationships, **and/or** - communicate an idea.	independently uses data to: - highlight or propose relationships, **and/or** - predict outcomes, **and/or** - communicate an idea.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., proposes alternative models, proposes additional factors that could affect a relationship).

ALGORITHMS & PROGRAMMING	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
ALGORITHMS (AP.A)	**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 programing language).	provides little to no evidence in addressing the expectation(s).	with guidance, follows algorithms to complete tasks.	with guidance: - follows algorithms to complete tasks. - creates algorithms to complete tasks.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., independently creates algorithms via one or more of the following techniques: verbally, kinesthetically, with robot devices, or a programing language).
	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.	provides little to no evidence in addressing the expectation(s).	compares simple algorithms for the same task.	- compares and refines multiple algorithms for the same task. - determines which algorithm is the most appropriate for the same task.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., develop alternative algorithms).
VARIABLES (AP.V)	**2.AP.V.01** Model the way programs store and manipulate data by using numbers or other symbols to represent information (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).	provides little to no evidence in addressing the expectation(s).	with guidance: - uses symbols to represent information. - understands that inferred meanings of the symbols can change or can represent missing information. - creates expressions with symbols to convey data, information, or processes.	- uses symbols to represent information. - understands that inferred meanings of the symbols can change or can represent missing information. - creates expressions with symbols to convey data, information, or processes.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., creates complex expressions with symbols).

ALGORITHMS & PROGRAMMING	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
VARIABLES Continued **(AP.V)**	**5.AP.V.01** Using grade appropriate content and complexity, create programs that use variables to store and modify data.	provides little to no evidence in addressing the expectation(s).	modifies programs that use variables to: - store data. - modify data.	creates programs that use variables to: - store data. - modify data.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., uses a variety of variable types).
CONTROL (AP.C)	**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.	provides little to no evidence in addressing the expectation(s).	with guidance, create: - programs that include sequencing, conditionals, or repetition. - tasks that include sequencing, conditionals, or repetition.	with guidance: - individually create programs or tasks that include sequencing, conditionals, and repetition. - collaboratively create programs that include sequencing, conditionals, and repetition.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., independently creates programs that include sequencing, conditionals, and repetition).
	5.AP.C.01 Using grade appropriate content and complexity, create programs that include sequences, events, loops, and conditionals, both individually and collaboratively.	provides little to no evidence in addressing the expectation(s).	- independently, create programs that include sequences and events. - collaboratively, create programs that include sequences and events.	- independently, create programs that include combinations of sequences, events, loops, and conditionals. - collaboratively, create programs that include combinations of sequences, events, loops, and conditionals.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., incorporating nested loops and complex conditionals).
MODULARITY (AP.M)	**2.AP.M.01** Using 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).	provides little to no evidence in addressing the expectation(s).	with guidance: - decomposes a problem. - creates a precise sequence of instructions to solve that problem.	- decomposes a problem. - creates a precise sequence of instructions to solve that problem.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., can create different instruction sets that accomplish the same task).

ALGORITHMS & PROGRAMMING	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
MODULARITY Continued **(AP.M)**	**5.AP.M.01** Using grade appropriate content and complexity, decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. **5.AP.M.02** Using 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.	provides little to no evidence in addressing the expectation(s).	- decomposes (breaks down) problems into smaller, manageable subproblems to facilitate the program development process, **and/or** - modifies, remixes, or incorporates portions of an existing program into one's own work.	- decomposes (breaks down) problems into smaller, manageable subproblems to facilitate the program development process. - modifies, remixes, or incorporates portions of an existing program into one's own work to develop something new or add more advanced features.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., helps others modify code, incorporates portions of multiple programs).
PROGRAM DEVELOPMENT (AP.PD)	**2.AP.PD.01** Develop plans that describe a program's sequence of events, goals, and expected outcomes. **2.AP.PD.02** Give credit to ideas, creations, and solutions of others while writing and developing programs. **2.AP.PD.03** Independently and collaboratively debug (identify and fix errors) programs using a programming language. **2.AP.PD.04** 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 a programming language).	provides little to no evidence in addressing the expectation(s).	with guidance, demonstrates program development by: - creating a plan for a program. - writing the program. - giving credit for the resources used. - debugging the program.	demonstrates program development by: - creating a plan for a program. - writing the program. - giving credit for the resources used. - debugging the program.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., demonstrates the development process on different platforms, languages, or mediums).

ALGORITHMS & PROGRAMMING	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
PROGRAM DEVELOPMENT Continued **(AP.PD)**	**5.AP.PD.01** Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences. **5.AP.PD.02** Using grade appropriate content and complexity, observe intellectual property rights and give appropriate credit when creating or remixing programs. **5.AP.PD.03** Using grade appropriate content and complexity, test and debug (i.e., identify and fix errors) a program or algorithm to ensure it runs as intended. **5.AP.PD.04** Using grade appropriate content and complexity, describe choices made during program development using code comments, presentations, and demonstrations. **5.AP.PD.05** Using grade appropriate content and complexity, with teacher guidance, perform varying roles when collaborating with peers during the design, implementation, and review stages of program development.	provides little to no evidence in addressing the expectation(s).	- observes intellectual property rights and gives appropriate credit when creating or remixing programs, **and** - uses an iterative process to plan the development of a program by including other perspectives and considers user preferences, **and/or** - tests and debugs (identify and fix errors) a program or algorithm to ensure it runs as intended, **and/or** - describes choices made during program development using code comments, presentations, and demonstrations, **and/or** - with teacher guidance, performs varying roles when collaborating with peers during the design, implementation, and review stages of program development.	- observes intellectual property rights and gives appropriate credit when creating or remixing programs. - uses an iterative process to plan the development of a program by including other perspectives and considers user preferences. - tests and debugs (identify and fix errors) a program or algorithm to ensure it runs as intended. - describes choices made during program development using code comments, presentations, and demonstrations. - with teacher guidance, performs varying roles when collaborating with peers during the design, implementation, and review stages of program development.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard. By way of examples, - justifies their own copyright on their work; -explains the different types of copyrights and the process of getting permission; - provides guidance to other students when testing and debugging a program or algorithm; - proposes alternatives and justifies why they went with their current code.

IMPACTS OF COMPUTING	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
CULTURE (IC.C)	**2.IC.C.01** Describe how people use different types of technologies in their daily work and personal lives.	provides little to no evidence in addressing the expectation(s).	identifies how people use different types of technologies (e.g., cell phones, computers) in their daily work and personal lives.	describes how people use different types of technologies (e.g., cell phones, computers) in their daily work and personal lives.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., identifies and describes the potential impacts of different technologies).
	5.IC.C.01 Give examples and explain how computing technologies have changed the world and express how those technologies influence and are influenced by cultural practices. **5.IC.C.02** Develop, test, and refine digital artifacts or devices to improve accessibility and usability for diverse end users.	provides little to no evidence in addressing the expectation(s).	- gives examples of how computing technologies have changed the world, **and/or** - expresses how technologies interact with cultural practices, **and/or** - tests digital artifacts or devices for accessibility and usability for diverse end users.	- gives examples and explains how computing technologies have changed the world. - expresses how technologies influence and are influenced by cultural practices. - develops, tests, and refines digital artifacts or devices to improve accessibility and usability for diverse end users.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., makes and justifies predictions based on historical patterns, incorporates multiple forms of accessibility in one artifact).
SOCIAL INTERACTIONS (IC.SI)	**2.IC.SI.01** Practice grade-level appropriate behavior and responsibilities while participating in an online community. Identify and report inappropriate behavior.	provides little to no evidence in addressing the expectation(s).	with guidance: - makes appropriate choices when participating in an online community. - identifies inappropriate behavior and reporting procedures.	- makes appropriate choices when participating in an online community. - identifies inappropriate behavior and reporting procedures.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.

IMPACTS OF COMPUTING	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
SOCIAL INTERACTIONS Continued **(IC.SI)**	**5.IC.SI.01** Seek diverse perspectives for the purpose of improving computational artifacts. **5.IC.SI.02** Practice grade-level appropriate behavior and responsibilities while participating in an online community. Identify and report inappropriate behavior.	provides little to no evidence in addressing the expectation(s).	- practices grade-level appropriate behavior and responsibilities while participating in an online community. - identifies and reports inappropriate behavior, when applicable.	- seeks diverse perspectives for the purpose of improving computational artifacts. - practices grade-level appropriate behavior and responsibilities while participating in an online community. - identifies and reports inappropriate behavior, when applicable.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., creates resources that models or explains to peers how to participate in online communities or independently uses video conferencing tools or other online collaborative spaces, such as blogs, wikis, forums, or website comments, to gather feedback from individuals and groups).
SAFETY, LAW, & ETHICS **(IC.SLE)**	**5.IC.SLE.01** Recognize and appropriately use public domain and creative commons media and discuss the social impact of violating intellectual property rights.	provides little to no evidence in addressing the expectation(s).	identifies types of digital data that may have intellectual property rights that prevent copying or require attribution.	- recognizes and appropriately uses public domain and creative commons media. - discusses the social impact of violating intellectual property rights.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., explain the process of contributing to a public domain or creative commons media, create and use a custom intellectual property rights system used by members of the class).

Computer Science | 6-8 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.

By the end of 8th grade, students can:

- Systematically identify, recommend, resolve, and document increasingly complex software and hardware problems with computing devices and their components
- Model the role of protocols in transmitting data across networks and the internet
- Critique physical and digital procedures that could be implemented to protect electronic data/information
- Use and refine computational tools to transform collected data in order to make it more useful and reliable
- Create flowcharts and pseudocode to design algorithms to solve complex problems
- Create clearly named variables that represent different data types and perform operations on their values
- Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals
- Decompose problems into parts to facilitate the design, implementation, and review of programs
- Create procedures with parameters to organize code and make it easier to reuse
- Seek and incorporate feedback from team members and users to refine a solution to a problem
- Describe impacts associated with computing technologies that affect people's everyday activities and career options along with issues of bias and accessibility in the design of technologies
- Practice grade-level appropriate behavior and responsibilities while participating in an online community, including identifying and reporting inappropriate behavior
- Describe tradeoffs between allowing information to be public and keeping information private and secure
- Discuss the legal, social, and ethical impacts associated with software development and use, including both positive and malicious intent

Computer Science | 9-12 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.

By the end of 12th grade, students can:

- Create a computer program using sequencing, selection, and iteration
- Decompose complex problems into smaller, more manageable sections
- Use tools of coding to create, debug, and document the evolution of an artifact
- Compare and contrast trade-offs in programming techniques
- Develop complex computer program individually and as part of a group
- Recognize how various components of a complex computing system work together
- Use tools to analyze data and know how data is stored
- Explain how cybersecurity issues affect networks and the internet
- Justify how proliferation of computing affects privacy, rights, opportunities, and responsibility

The high school standards are organized into 2 levels. Mostly, Level 1 is intended to be at the introductory level, and Level 2 reaches at a deeper level.

DOMAIN - KEY	COMPUTING SYSTEMS	NETWORKS & THE INTERNET	DATA & ANALYSIS	ALGORITHMS & PROGRAMMING	IMPACTS OF COMPUTING

COMPUTING SYSTEMS	End of Grade 8	High School Level 1	High School Level 2
DEVICES (CS.D)	**8.CS.D.01** Recommend improvements to the design of computing devices based on an analysis of how a variety of users interact with the device. **Practice 3.3** Recognizing and Defining Computational Problems	**L1.CS.D.01** Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects. **Practice 4.1** Developing and Using Abstractions	
HARDWARE & SOFTWARE (CS.HS)	**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	**L1.CS.HS.01** Explain the interactions between application software, system software, and hardware layers. **Practice 4.1** Developing and Using Abstractions	**L2.CS.HS.01** Categorize the roles of operating system software. **Practice 4.1** Developing and Using Abstractions **Practice 7.2** Communicating About Computing
TROUBLESHOOTING (CS.T)	**8.CS.T.01** Systematically identify, resolve, and document increasingly complex software and hardware problems with computing devices and their components. **Practice 6.2** Testing and Refining Computational Artifacts	**L1.CS.T.01** Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and resolve errors. **Practice 6.1 & 6.2** Testing and Refining Computational Artifacts	**L2.CS.T.01** Identify how hardware components facilitate logic, input, output, and storage in computing systems, and their common malfunctions. **Practice 7.2** Communicating About Computing

NETWORKS & THE INTERNET	End of Grade 8	High School Level 1	High School Level 2
NETWORK COMMUNICATION & ORGANIZATION (NI.NCO)	**8.NI.NCO.01** Model the role of protocols in transmitting data across networks and the internet (e.g., explain protocols and their importance to data transmission; model how packets are broken down into smaller pieces and how they are delivered). **Practice 4.4** Developing and Using Abstractions	**L1.NI.NCO.01** Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing. **Practice 4.1** Developing and Using Abstractions **Practice 7.2** Communicating About Computing	**L2.NI.NCO.01** Describe the issues that impact network functionality (e.g., bandwidth, load, latency, topology). **Practice 7.2** Communicating About Computing
CYBERSECURITY (NI.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	**L1.NI.C.01** Give examples to illustrate how sensitive data can be affected by malware and other attacks. **Practice 7.2** Communicating About Computing	**L2.NI.C.01** Compare ways software developers protect devices and information from unauthorized access. **Practice 7.2** Communicating About Computing
	8.NI.C.02 Apply multiple methods of encryption to model the secure transmission of data. **Practice 4.4** Developing and Using Abstractions	**L1.NI.C.02** Recommend cybersecurity measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts. **Practice 3.3** Recognizing and Defining Computational Problems	
		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	
		L1.NI.C.04 Explain trade-offs when selecting and implementing cybersecurity recommendations. **Practice 7.2** Communicating About Computing	

DATA & ANALYSIS	End of Grade 8	High School Level 1	High School Level 2
STORAGE (DA.S)	**8.DA.S.01** Represent data using multiple encoding schemes (e.g., ASCII, binary). **Practice 4.4** Developing and Using Abstractions	**L1.DA.S.01** Translate between different bit representations of real-world phenomena, such as characters, numbers, and images. **Practice 4.1** Developing and Using Abstractions	
		L1.DA.S.02 Evaluate the trade-offs in how data elements are organized and where data is stored. **Practice 3.3** Recognizing and Defining Computational Problems	
COLLECTION, VISUALIZATION, & TRANSFORMATION (DA.CVT)	**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 🔌	**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 🔌	**L2.DA.CVT.01** Use data analysis tools and techniques to identify patterns in data representing complex systems. **Practice 4.1** Developing and Using Abstractions **Practice 7.1** Communicating About Computing 🔌
			L2.DA.CVT.02 Select data collection tools and techniques, and use them to generate data sets that support a claim or communicate information. **Practice 7.1 & 7.2** Communicating About Computing 🔌
INFERENCE & MODELS (DA.IM)	**8.DA.IM.01** Refine computational models based on generated data. **Practice 4.4** Developing and Using Abstractions **Practice 5.3** Creating Computational Artifacts 🔌	**L1.DA.IM.01** Create computational models that represent the relationships among different elements of data collected from a phenomenon or process. **Practice 4.4** Developing and Using Abstractions 🔌	**L2.DA.IM.01** Formulate, refine, and test scientific hypotheses using models and simulations. **Practice 4.4** Developing and Using Abstractions

ALGORITHMS & PROGRAMMING	End of Grade 8	High School Level 1	High School Level 2
ALGORITHMS (AP.A)	**8.AP.A.01** Create flowcharts and pseudocode to design algorithms to solve complex problems. **Practice 4.1 & 4.4** Developing and Using Abstractions	**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 🔌	**L2.AP.A.01** Critically examine and trace classic algorithms. Use and adapt classic algorithms to solve computational problems (e.g., selection sort, insertion sort, binary search, linear search). **Practice 4.2** Developing and Using Abstractions
		L1.AP.A.02 Describe how artificial intelligence algorithms drive many software and physical systems. **Practice 7.2** Communicating About Computing	**L2.AP.A.02** Develop an artificial intelligence algorithm to play a game against a human opponent or solve a real-world problem. **Practice 5.2 & 5.3** Creating Computational Artifacts
			L2.AP.A.03 Evaluate algorithms (e.g., sorting, searching) in terms of their efficiency, correctness, and clarity. **Practice 4.2** Developing and Using Abstractions
VARIABLES (AP.V)	**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	**L1.AP.V.01** Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables. **Practice 4.1** Developing and Using Abstractions 🔌	**L2.AP.V.01** Compare and contrast simple data structures and their uses (e.g., lists, stacks, queues). **Practice 4.2** Developing and Using Abstractions
CONTROL (AP.C)	**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 🔌	**L1.AP.C.01** Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made. **Practice 5.2** Creating Computational Artifacts	

ALGORITHMS & PROGRAMMING	End of Grade 8	High School Level 1	High School Level 2
CONTROL Continued **(AP.C)**		**L1.AP.C.02** Trace the execution of loops and conditional statements, illustrating output and changes in values of named variables. **Practice 3.2** Recognizing and Defining Computational Problems	**L2.AP.C.01** Trace the execution of recursion, illustrating output and changes in values of named variables. **Practice 3.2** Recognizing and Defining Computational Problems
		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 🔌	
MODULARITY **(AP.M)**	**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	**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	**L2.AP.M.01** Construct solutions to problems using student-created components, such as procedures, modules, and/or objects. **Practice 4.3** Developing and Using Abstractions **Practice 5.2** Creating Computational Artifacts
	8.AP.M.02 Using grade appropriate content and complexity, create procedures with parameters to organize code and make it easier to reuse. **Practice 4.1 & 4.3** Developing and Using Abstractions	**L1.AP.M.02** Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs. **Practice 5.2** Creating Computational Artifacts 🔌	**L2.AP.M.02** Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution. **Practice 4.1** Developing and Using Abstractions
			L2.AP.M.03 Demonstrate code reuse by creating programming solutions using libraries and APIs. **Practice 4.2** Developing and Using Abstractions 🔌 **Practice 5.3** Creating Computational Artifacts

ALGORITHMS & PROGRAMMING	End of Grade 8	High School Level 1	High School Level 2
PROGRAM DEVELOPMENT (AP.PD)	**8.AP.PD.01** Using grade appropriate content and complexity, seek and incorporate feedback from team members and users to refine a solution to a problem. **Practice 1.1** Fostering an Inclusive Computing Culture **Practice 2.3** Collaborating Around Computing	**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 🔌	**L2.AP.PD.01** Plan and develop programs that will provide solutions to a variety of users using a software life cycle process. **Practice 5.1** Creating Computational Artifacts 🔌
	8.AP.PD.02 Incorporate existing code, media, and libraries into original programs of increasing complexity and give attribution. **Practice 4.2** Developing and Using Abstractions **Practice 5.2** Creating Computational Artifacts **Practice 7.3** Communicating About Computing 🔌	**L1.AP.PD.02** Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries. **Practice 7.3** Communicating About Computing	**L2.AP.PD.02** Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (e.g., code documentation) in a group software project. **Practice 2.4** Collaborating Around Computing 🔌
	8.AP.PD.03 Systematically test and refine programs using a range of test cases. **Practice 6.1** Testing and Refining Computational Artifacts	**L1.AP.PD.03** Use debugging tools to identify and fix errors in a program. **Practice 6.2** Testing and Refining Computational Artifacts 🔌	
		L1.AP.PD.04 Design and develop computational artifacts, working in team roles, using collaborative tools. **Practice 2.4** Collaborating Around Computing	**L2.AP.PD.03** Develop programs for multiple computing platforms. **Practice 5.2** Creating Computational Artifacts 🔌
	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	**L1.AP.PD.05** Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs. **Practice 7.2** Communicating About Computing	**L2.AP.PD.04** Evaluate key qualities of a program through a process such as a code review (e.g., qualities could include correctness, usability, readability, efficiency, portability, and scalability).

			Practice 6.3 Testing and Refining Computational Artifacts
ALGORITHMS & PROGRAMMING	**End of Grade 8**	**High School Level 1**	**High School Level 2**
PROGRAM DEVELOPMENT Continued **(AP.PD)**	**8.AP.PD.05** Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts. **Practice 2.2** Collaborating Around Computing	**L1.AP.PD.06** Evaluate and refine computational artifacts to make them more usable and accessible. **Practice 6.3** Testing and Refining Computational Artifacts 🔌	**L2.AP.PD.05** Develop and use a series of test cases to verify that a program performs according to its design specifications. **Practice 6.1** Testing and Refining Computational Artifacts 🔌
			L2.AP.PD.06 Explain security issues that might lead to compromised computer programs. **Practice 7.2** Communicating About Computing
			L2.AP.PD.07 Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality). **Practice 5.3** Creating Computational Artifacts
			L2.AP.PD.08 Compare multiple programming languages and discuss how their features make them suitable for solving different types of problems. **Practice 7.2** Communicating About Computing

IMPACTS OF COMPUTING	End of Grade 8	High School Level 1	High School Level 2
CULTURE (IC.C)	**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	**L1.IC.C.01** Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices. **Practice 1.2** Fostering an Inclusive Computing Culture	**L2.IC.C.01** Evaluate the beneficial and harmful effects that computational artifacts and innovations have on society. **Practice 1.2** Fostering an Inclusive Computing Culture
	8.IC.C.02 Describe issues of bias and accessibility in the design of technologies. **Practice 1.2** Fostering an Inclusive Computing Culture	**L1.IC.C.02** Test and refine computational artifacts to reduce bias and equity deficits. **Practice 1.2** Fostering an Inclusive Computing Culture	**L2.IC.C.02** Evaluate the impact of equity, access, and influence on the distribution of computing resources in a global society. **Practice 1.2** Fostering an Inclusive Computing Culture
		L1.IC.C.03 Demonstrate how a given algorithm applies to problems across disciplines. **Practice 3.1** Recognizing and Defining Computational Problems	**L2.IC.C.03** Predict how computational innovations that have revolutionized aspects of our culture might evolve. **Practice 5.2** Creating Computational Artifacts
SOCIAL INTERACTIONS (IC.SI)	**8.IC.SI.01** Using grade appropriate content and complexity, collaborate using tools to connect with peers when creating a computational artifact. **Practice 2.4** Collaborating Around Computing **Practice 5.2** Creating Computational Artifacts	**L1.IC.SI.01** Use tools and methods for collaboration. **Practice 2.4** Collaborating Around Computing	
	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	**L1.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	**L2.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 **Practice 7.3** Communicating About Computing

IMPACTS OF COMPUTING	End of Grade 8	High School Level 1	High School Level 2
SAFETY, LAW, & ETHICS (IC.SLE)	**8.IC.SLE.01** Using grade appropriate content and complexity, describe tradeoffs between allowing information to be public and keeping information private and secure. **Practice 7.2** Communicating About Computing	**L1.IC.SLE.01** Explain the beneficial and harmful effects that intellectual property laws can have on innovation. **Practice 7.3** Communicating About Computing	**L2.IC.SLE.01** Debate laws and regulations that impact the development and use of software and technology. **Practice 3.3** Recognizing and Defining Computational Problems **Practice 7.3** Communicating About Computing
		L1.IC.SLE.02 Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users. **Practice 7.2** Communicating About Computing	
		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	
	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	**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	**L2.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

DOMAIN - KEY	COMPUTING SYSTEMS	NETWORKS & THE INTERNET	DATA & ANALYSIS	ALGORITHMS & PROGRAMMING	IMPACTS OF COMPUTING

COMPUTING SYSTEMS	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
DEVICES (CS.D)	**8.CS.D.01** Recommend improvements to the design of computing devices based on an analysis of how a variety of users interact with the device.	provides little to no evidence in addressing the expectation(s).	- understands the needs of the users, but is unable to analyze, **and/or** - describes the parts of computing devices, but cannot recommend improvements to the design.	- analyzes the needs of the users. - recommends improvements to the design of computing devices based on that analysis.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., recommend improvements to the design in more than one area (input, output, processing, storage) or group (special populations).
	L1.CS.D.01 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.	provides little to no evidence in addressing the expectation(s).	identifies abstractions that hide the underlying implementation details of computing systems embedded in everyday objects.	explains how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.
HARDWARE & SOFTWARE (CS.HS)	**8.CS.HS.01** Design and refine a project that combines hardware and software components to collect and exchange data.	provides little to no evidence in addressing the expectation(s).	- 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.	- 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.	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).

COMPUTING SYSTEMS	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
HARDWARE & SOFTWARE Continued **(CS.HS)**	**L1.CS.HS.01** Explain the interactions between application software, system software, and hardware layers.	provides little to no evidence in addressing the expectation(s).	- identifies application software, system software, and hardware layers. - defines application software, system software, and hardware layers.	- 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.	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).
	L2.CS.HS.01 Categorize the roles of operating system software.	provides little to no evidence in addressing the expectation(s).	categorizes some of the roles of operating system software.	categorizes the roles of the operating system software (e.g., roles could include memory management, data storage/retrieval, process management, and access control).	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.

COMPUTING SYSTEMS	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
TROUBLESHOOTING (CS.T)	**8.CS.T.01** Systematically identify, resolve, and document increasingly complex software and hardware problems with computing devices and their components.	provides little to no evidence in addressing the expectation(s).	can do some of the following: - identify software problems with computing devices and their components, - identify hardware problems with computing devices and their components, - resolve software problems with computing devices and their components, - resolve hardware problems with computing devices and their components, - document software problems with computing devices and their components, - document hardware problems with computing devices and their components.	can systematically: - identify software problems with computing devices and their components, - identify hardware problems with computing devices and their components, - resolve software problems with computing devices and their components, - resolve hardware problems with computing devices and their components, - document software problems with computing devices and their components, - document hardware problems with computing devices and their components.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., systematically assists others with hardware or software problems, creates a detailed troubleshooting document or tutorial, comes up with novel solutions).
	L1.CS.T.01 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and resolve errors.	provides little to no evidence in addressing the expectation(s).	develops guidelines with support that convey systematic troubleshooting strategies that others can use to identify and resolve errors.	develops guidelines independently that convey systematic troubleshooting strategies that others can use to identify and resolve errors (e.g., students could create a flow chart, a job aid for a help desk employee, or an expert system).	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., someone with limited experience or knowledge could follow student developed guidelines).

COMPUTING SYSTEMS	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
TROUBLESHOOTING Continued **(CS.T)**	**L2.CS.T.01** Identify how hardware components facilitate logic, input, output, and storage in computing systems, and their common malfunctions.	provides little to no evidence in addressing the expectation(s).	identifies how some hardware components: - facilitate logic, input, output, and storage in computing systems, **and/or** - some of their common malfunctions.	identifies: - how hardware components facilitate logic, input, output, and storage in computing systems. - hardware components common malfunctions.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.

NETWORKS & THE INTERNET	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
NETWORK COMMUNICATION & ORGANIZATION (NI.NCO)	**8.NI.NCO.01** Model the role of protocols in transmitting data across networks and the internet (e.g., explain protocols and their importance to data transmission; model how packets are broken down into smaller pieces and how they are delivered).	provides little to no evidence in addressing the expectation(s).	- identifies protocols used in transmitting data across networks and the internet, **and/or** - explains the role of protocols in transmitting data across networks and the internet.	- models the role of protocols in transmitting data across networks and the internet.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., research and compare/contrast multiple network protocols).
	L1.NI.NCO.01 Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.	provides little to no evidence in addressing the expectation(s).	- identifies routers, switches, servers, topology, and addressing. - defines routers, switches, servers, topology, and addressing.	by describing the relationship between routers, switches, servers, topology, and addressing, evaluates the: - scalability of networks. - reliability of networks.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., students can discuss different types of routers, switches, servers and/or topologies).
	L2.NI.NCO.01 Describe the issues that impact network functionality (e.g., bandwidth, load, latency, topology).	provides little to no evidence in addressing the expectation(s).	describes a limited number of issues that impact network functionality (e.g., bandwidth, load, latency, topology).	describes common issues that impact network functionality (e.g., bandwidth, load, latency, topology).	demonstrates an understanding of trade-offs between network functionality and design.

NETWORKS & THE INTERNET	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
CYBERSECURITY (NI.C)	**8.NI.C.01** Using grade appropriate content and complexity, create programs that use variables to store and modify data. **8.NI.C.02** Apply multiple methods of encryption to model the secure transmission of data.	provides little to no evidence in addressing the expectation(s).	- 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.	- critiques physical and digital procedures that could be implemented to protect electronic data/information. - applies multiple methods of encryption to model the secure transmission of data.	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).
	L1.NI.C.01 Give examples to illustrate how sensitive data can be affected by malware and other attacks.	provides little to no evidence in addressing the expectation(s).	recalls examples to illustrate how sensitive data can be affected by malware and other attacks.	gives multiple detailed examples to illustrate how sensitive data can be affected by malware and other attacks.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.
	L2.NI.C.01 Compare ways software developers protect devices and information from unauthorized access.	provides little to no evidence in addressing the expectation(s).	lists ways software developers protect: - devices from unauthorized access. - information from unauthorized access.	compares ways software developers protect: - devices from unauthorized access. - information from unauthorized access.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., encryption strategies, authentication strategies).
	L1.NI.C.02 Recommend cybersecurity measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts.	provides little to no evidence in addressing the expectation(s).	identifies cybersecurity measures to address various scenarios.	recommends cybersecurity measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.

NETWORKS & THE INTERNET	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
CYBERSECURITY Continued **(NI.C)**	**L1.NI.C.03** Compare various security measures, considering trade-offs between the usability and security of a computing system.	provides little to no evidence in addressing the expectation(s).	- identifies various security measures. - defines various security measures.	compares various security measures, considering trade-offs between the usability and security of a computing system.	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 trade-off between usability and security).
	L1.NI.C.04 Explain trade-offs when selecting and implementing cybersecurity recommendations.	provides little to no evidence in addressing the expectation(s).	when selecting and implementing cybersecurity recommendations, can give an example of trade-offs: - from a single viewpoint, **and/or** - with inappropriate terminology.	explains trade-offs from multiple perspectives using appropriate terminology when selecting and implementing cybersecurity recommendations.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., make a recommendation and justify).

DATA & ANALYSIS	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
STORAGE **(DA.S)**	**8.DA.S.01** Represent data using multiple encoding schemes (e.g., ASCII, binary).	provides little to no evidence in addressing the expectation(s).	- recognizes data is stored in multiple encoding schemes.	- represents data using multiple encoding schemes.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., convert data between multiple encoding schemes; ASCII to binary, hex to rgb).
	L1.DA.S.01 Translate between different bit representations of real-world phenomena, such as characters, numbers, and images.	provides little to no evidence in addressing the expectation(s).	can translate between a bit representation of real-world phenomena, such as characters, numbers, or images.	translates between different bit representations of real-world phenomena, such as characters, numbers, and images.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.

DATA & ANALYSIS	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
STORAGE Continued **(DA.S)**	**L1.DA.S.02** Evaluate the trade-offs in how data elements are organized and where data is stored.	provides little to no evidence in addressing the expectation(s).	- identifies the trade-offs in how data elements are organized and where data is stored. - describes the trade-offs in how data elements are organized and where data is stored.	evaluates the trade-offs in how data elements are organized and where data is stored.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., research emerging technologies for data storage and evaluate trade-off with current technologies).
COLLECTION, VISUALIZATION, & TRANSFORMATION (DA.CVT)	**8.DA.CVT.01** Using computational tools, transform collected data to make it more useful and reliable.	provides little to no evidence in addressing the expectation(s).	- explores a variety of computational tools and the content of their data. - uses computational tools to collect data.	determines appropriate computational tools to: - transform data to remove errors. - highlight or expose relationships in the data.	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).
	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).	provides little to no evidence in addressing the expectation(s).	creates, with errors, interactive data representations using software tools.	creates, with no or minor errors, appropriate interactive data representations using software tools to help others better understand real-world phenomena.	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).
	L2.DA.CVT.01 Use data analysis tools and techniques to identify patterns in data representing complex systems.	provides little to no evidence in addressing the expectation(s).	uses data analysis tools and techniques to identify patterns in data representing complex systems but draws incorrect conclusions.	uses data analysis tools and techniques to identify correct patterns in data representing complex systems.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., make a plausible predication based on pattern).

DATA & ANALYSIS	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
COLLECTION, VISUALIZATION, & TRANSFORMATION Continued **(DA.CVT)**	**L2.DA.CVT.02** Select data collection tools and techniques, and use them to generate data sets that support a claim or communicate information.	provides little to no evidence in addressing the expectation(s).	- selects data collection tools and techniques. - uses data collection tools and techniques to generate data sets but are unable to support a claim or communicate information.	- selects data collection tools and techniques. - uses data collection tools to generate data sets that support a claim or communicate information.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.
INFERENCE & MODELS (DA.IM)	**8.DA.IM.01** Refine computational models based on generated data.	provides little to no evidence in addressing the expectation(s).	- uses models and simulations to formulate, refine, and test hypotheses, **and/or** - tests and analyzes the effects of changing variables while using computational models.	- refines computational models based on generated data.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., make multiple refinements).
	L1.DA.IM.01 Create computational models that represent the relationships among different elements of data collected from a phenomenon or process.	provides little to no evidence in addressing the expectation(s).	creates computational models that represent the relationships among different elements of data collected from a phenomenon or process.	creates accurate computational models that represent the relationships among different elements of data collected from a phenomenon or process.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.
	L2.DA.IM.01 Formulate, refine, and test scientific hypotheses using models and simulations.	provides little to no evidence in addressing the expectation(s).	formulates scientific hypotheses using models and simulations.	- formulates scientific hypotheses using models and simulations. - refines scientific hypotheses using models and simulations. - tests scientific hypotheses using models and simulations.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.

ALGORITHMS & PROGRAMMING	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
ALGORITHMS (AP.A)	**8.AP.A.01** Create flowcharts and pseudocode to design algorithms to solve complex problems.	provides little to no evidence in addressing the expectation(s).	- uses flowcharts to modify existing algorithms, **and/or** - uses pseudocode to modify existing algorithms, **and/or** - uses natural language to modify existing algorithms.	- creates flowcharts to design algorithms to solve complex problems. - writes pseudocode to design algorithms to solve complex problems.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., design algorithms to solve complex problems in multiple ways and determine and use the most effective planning tool).
	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.	provides little to no evidence in addressing the expectation(s).	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.	creates a prototype that uses multiple algorithms (e.g., searching, sorting, finding shortest distance) to provide a possible solution for a real-world problem relevant to the student.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., student generated problem).
	L2.AP.A.01 Critically examine and trace classic algorithms. Use and adapt classic algorithms to solve computational problems (e.g., selection sort, insertion sort, binary search, linear search).	provides little to no evidence in addressing the expectation(s).	- examines and traces classic algorithms with minor errors. - uses classic algorithms to solve computational problems.	- critically examines and traces classic algorithms. - uses classic algorithms to solve computational problems. - adapts classic algorithms to solve computational problems.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., use and justify why a given algorithm is more efficient than another).
	L1.AP.A.02 Describe how artificial intelligence algorithms drive many software and physical systems.	provides little to no evidence in addressing the expectation(s).	describes how artificial intelligence algorithms drive a software system or physical system.	describes how artificial intelligence algorithms drive many software and physical systems.	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).

ALGORITHMS & PROGRAMMING	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
ALGORITHMS Continued **(AP.A)**	**L2.AP.A.02** Develop an artificial intelligence algorithm to play a game against a human opponent or solve a real-world problem.	provides little to no evidence in addressing the expectation(s).	- develops an artificial intelligence algorithm to play a game against a human opponent or solve a real-world problem. - incorrectly captures some rules of the game.	- develops an artificial intelligence algorithm to play a game against a human opponent or solve a real-world problem. - correctly implements all rules of the game.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., uses heuristics to select the moves of the computer).
	L2.AP.A.03 Evaluate algorithms (e.g., sorting, searching) in terms of their efficiency, correctness, and clarity.	provides little to no evidence in addressing the expectation(s).	evaluates algorithms in terms of their: - efficiency, **or** - correctness, **or** - clarity.	evaluates algorithms in terms of their: - efficiency. - correctness. - clarity.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.
VARIABLES **(AP.V)**	**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.	provides little to no evidence in addressing the expectation(s).	- recognizes that variables can represent different data types, **and/or** - can create a variable, **and/or** - can perform operations on the values of variables.	- clearly names variables. - creates variables that represent different data types. - performs operations on the values of variables.	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).
	L1.AP.V.01 Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.	provides little to no evidence in addressing the expectation(s).	with guidance, uses lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.	independently uses lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., uses standard list operations like filter, map, and reduce).

ALGORITHMS & PROGRAMMING	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
VARIABLES Continued **(AP.V)**	**L2.AP.V.01** Compare and contrast simple data structures and their uses (e.g., lists, stacks, queues).	provides little to no evidence in addressing the expectation(s).	- identifies simple linear data structures and their uses. - explains simple linear data structures and their uses.	compares and contrasts simple linear data structures and their uses.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., trees).
CONTROL **(AP.C)**	**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.	provides little to no evidence in addressing the expectation(s).	designs and iteratively develops programs that: - use simple loops. - use simple conditionals.	designs and iteratively develops programs that include: - nested loops. - compound conditionals.	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).
	L1.AP.C.01 Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.	provides little to no evidence in addressing the expectation(s).	justifies the selection of specific control structures when tradeoffs involve: - implementation, **or** - readability, **or** - program performance.	- justifies the selection of specific control structures when tradeoffs involve implementation, readability, and program performance. - explains the benefits and drawbacks of choices.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., exception handling).
	L1.AP.C.02 Trace the execution of loops and conditional statements, illustrating output and changes in values of named variables.	provides little to no evidence in addressing the expectation(s).	traces the execution of: - loops illustrating output and changes in values of named variables, **or** - conditional statements illustrating output and changes in values of named variables.	traces the execution of: - loops illustrating output and changes in values of named variables, **and** - conditional statements illustrating output and changes in values of named variables.	In addition to the proficient level, student demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.

ALGORITHMS & PROGRAMMING	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
CONTROL Continued **(AP.C)**	**L2.AP.C.01** Trace the execution of recursion, illustrating output and changes in values of named variables.	provides little to no evidence in addressing the expectation(s).	with guidance: - traces the execution of recursion. - illustrates output and changes in values of name variables.	independently: - traces the execution of linear recursion. - illustrates output and changes in values of name variables (e.g., factorial function).	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., Fibonacci).
	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.	provides little to no evidence in addressing the expectation(s).	designs computational artifacts that uses events to initiate instructions.	- designs computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions. - iteratively develops computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., using multiple user interface components).
MODULARITY (AP.M)	**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. **8.AP.M.02** Using grade appropriate content and complexity, create procedures with parameters to organize code and make it easier to reuse.	provides little to no evidence in addressing the expectation(s).	- recognizes the inefficiency of repetition in programming, **and/or** - recognizes the organizational, readability and labor-saving advantages of code reuse.	- decomposes problems and subproblems into parts. - creates procedures with parameters.	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).

ALGORITHMS & PROGRAMMING	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
MODULARITY Continued **(AP.M)**	**L1.AP.M.01** Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.	provides little to no evidence in addressing the expectation(s).	decomposes problems into smaller components that are incohesive or tightly coupled.	decomposes problems into smaller components that are highly cohesive and loosely coupled through systematic analysis, using constructs such as procedures, modules, and/or objects.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., an appropriate class hierarchy).
	L2.AP.M.01 Construct solutions to problems using student-created components, such as procedures, modules, and/or objects.	provides little to no evidence in addressing the expectation(s).	with guidance, constructs solutions to problems using student-created components, such as procedures, modules, and/or objects.	constructs solutions to problems using student-created components, such as procedures, modules, and/or objects.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.
	L1.AP.M.02 Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.	provides little to no evidence in addressing the expectation(s).	with guidance, creates artifacts by using: - procedures within a program, **or** - combinations of data and procedures, **or** - independent but interrelated programs.	independently, creates artifacts by using: - procedures within a program, **or** - combinations of data and procedures, **or** - independent but interrelated programs.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.
	L2.AP.M.02 Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.	provides little to no evidence in addressing the expectation(s).	- analyzes a large-scale computational problem and with guidance. - identifies generalizable patterns that can be applied to a solution.	- analyzes a large-scale computational problem. - independently identifies generalizable patterns that can be applied to a solution.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.
	L2.AP.M.03 Demonstrate code reuse by creating programming solutions using libraries and APIs.	provides little to no evidence in addressing the expectation(s).	with guidance, demonstrates code reuse by creating programming solutions using libraries and APIs.	independently, demonstrates code reuse by creating programming solutions using libraries and APIs.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.

ALGORITHMS & PROGRAMMING	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
PROGRAM DEVELOPMENT (AP.PD)	**8.AP.PD.01** Using grade appropriate content and complexity, seek and incorporate feedback from team members and users to refine a solution to a problem. **8.AP.PD.02** Incorporate existing code, media, and libraries into original programs of increasing complexity and give attribution. **8.AP.PD.03** Systematically test and refine programs using a range of test cases. Program Development: **8.AP.PD.04** Using grade appropriate content and complexity, document programs in order to make them easier to follow, test, and debug.	provides little to no evidence in addressing the expectation(s).	- 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.	- incorporates existing code, media, and libraries into original programs. - systematically tests and refines programs. - documents programs. - seeks and incorporates feedback.	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).
	8.AP.PD.05 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.	provides little to no evidence in addressing the expectation(s).	using a pre-written computational artifact: - identifies the project timeline tasks necessary for program development. - breaks down tasks and follows an individual timeline when developing a computational artifact.	when collaboratively developing computational artifacts: - distributes tasks. - maintains a project timeline.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., adjust the timeline and redistribute tasks to meet the deadline).

ALGORITHMS & PROGRAMMING	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
PROGRAM DEVELOPMENT Continued **(AP.PD)**	**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.	provides little to no evidence in addressing the expectation(s).	with instructor support, plans and develops programs by: - analyzing a problem and/or process. - developing and documenting a solution. - testing outcomes.	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.	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.
	L2.AP.PD.01 Plan and develop programs that will provide solutions to a variety of users using a software life cycle process.	provides little to no evidence in addressing the expectation(s).	with instructor support: - plans a program that will provide solutions to a variety of users using a software life cycle process. - develops a program that will provide solutions to a variety of users using a software life cycle process.	- plans a program that will provide solutions to a variety of users using a software life cycle process. - develops a program that will provide solutions to a variety of users using a software life cycle process.	independently: - plans a program that will provide solutions to a variety of users using a software life cycle process. - develops a program that will provide solutions to a variety of users using a software life cycle process.
	L1.AP.PD.02 Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries.	provides little to no evidence in addressing the expectation(s).	- identifies licenses that limit or restrict use of computational artifacts when using resources such as libraries. - defines licenses that limit or restrict use of computational artifacts when using resources such as libraries.	evaluates licenses that limit or restrict use of computational artifacts when using resources such as libraries (e.g.,students might consider two software libraries that address a similar need, justifying their choice based on the library that has the least restrictive license).	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.

ALGORITHMS & PROGRAMMING	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
PROGRAM DEVELOPMENT Continued **(AP.PD)**	**L2.AP.PD.02** Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (e.g., code documentation) in a group software project.	provides little to no evidence in addressing the expectation(s).	uses: - integrated development environments (IDEs) in a group software project. - collaborative tools or practices (e.g., code documentation) in a group software project.	uses: - version control systems in a group software project. - integrated development environments (IDEs) in a group software project. - collaborative tools and practices (e.g., code documentation) in a group software project.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.
	L1.AP.PD.03 Use debugging tools to identify and fix errors in a program.	provides little to no evidence in addressing the expectation(s).	identifies strategies to test and debug (identify and fix errors) a program or algorithm to ensure it runs.	tests and debugs (identify and fix errors) a program or algorithm to ensure it runs as intended.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.
	L1.AP.PD.04 Design and develop computational artifacts, working in team roles, using collaborative tools.	provides little to no evidence in addressing the expectation(s).	- designs computational artifacts using collaborative tools. - develops computational artifacts using collaborative tools.	designs and develops computational artifacts, working in team roles, using collaborative tools (e.g., team roles in pair programming are driver and navigator but could be more specialized in larger teams).	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.
	L2.AP.PD.03 Develop programs for multiple computing platforms.	provides little to no evidence in addressing the expectation(s).	with instructor support, develops programs for multiple computing platforms.	develops programs for multiple computing platforms (e.g., disparate programs for different platforms: computer desktop, web, or mobile).	develops programs for multiple cross-platform computing platforms (e.g., platforms could include: computer desktop, web, or mobile).

ALGORITHMS & PROGRAMMING	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
PROGRAM DEVELOPMENT Continued **(AP.PD)**	**L1.AP.PD.05** Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.	provides little to no evidence in addressing the expectation(s).	partially documents design decisions using: - text, graphics, presentations, **and/or** - demonstrations in the development of complex programs.	documents design decisions using: - text, graphics, presentations, **and/or** - demonstrations in the development of complex programs.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.
	L2.AP.PD.04 Evaluate key qualities of a program through a process such as a code review (e.g., qualities could include correctness, usability, readability, efficiency, portability, and scalability).	provides little to no evidence in addressing the expectation(s).	- identifies key qualities of a program. - defines key qualities of a program (e.g., correctness, usability, readability, efficiency, portability, and scalability).	evaluates key qualities of a program through a process such as a code review (e.g., correctness, usability, readability, efficiency, portability, and scalability).	evaluates key qualities of a program and makes recommendations to improve that program through a process such as a code review (e.g., correctness, usability, readability, efficiency, portability, and scalability).
	L1.AP.PD.06 Evaluate and refine computational artifacts to make them more usable and accessible.	provides little to no evidence in addressing the expectation(s).	with support: - evaluates computational artifacts to make them more usable and accessible. - refines computational artifacts to make them more usable and accessible.	- evaluates computational artifacts to make them more usable and accessible. - refines computational artifacts to make them more usable and accessible.	supports others as they: - evaluate computational artifacts to make them more usable and accessible. -refine computational artifacts to make them more usable and accessible.

ALGORITHMS & PROGRAMMING	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
PROGRAM DEVELOPMENT Continued **(AP.PD)**	**L2.AP.PD.05** Develop and use a series of test cases to verify that a program performs according to its design specifications.	provides little to no evidence in addressing the expectation(s).	uses a series of test cases to verify that a program performs according to its design specifications.	- develops a series of test cases to verify that a program performs according to its design specifications. - uses a series of test cases to verify that a program performs according to its design specifications. - at this level, students are expected to select their own test cases.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.
	L2.AP.PD.06 Explain security issues that might lead to compromised computer programs.	provides little to no evidence in addressing the expectation(s).	- identifies security issues that might lead to compromised computer programs. - describes security issues that might lead to compromised computer programs.	explains security issues that might lead to compromised computer programs (e.g., lack of bounds checking, poor input validation, and circular references).	explains and provides potential solutions for security issues that might lead to compromised computer programs.
	L2.AP.PD.07 Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality).	provides little to no evidence in addressing the expectation(s).	modifies an existing program to add additional functionality.	- modifies an existing program to add additional functionality. - discusses intended and unintended implications (e.g., breaking other functionality).	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.
	L2.AP.PD.08 Compare multiple programming languages and discuss how their features make them suitable for solving different types of problems.	provides little to no evidence in addressing the expectation(s).	- identifies multiple programming languages. - explains multiple programming languages.	- compares multiple programming languages. - discusses how their features make them suitable for solving different types of problems.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.

IMPACTS OF COMPUTING	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
CULTURE (IC.C)	**8.IC.C.01** Describe impacts associated with computing technologies that affect people's everyday activities and career options. **8.IC.C.02** Describe issues of bias and accessibility in the design of technologies.	provides little to no evidence in addressing the expectation(s).	- 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.	- describes impacts associated with computing technologies that affect people's everyday activities. - describes impacts associated with computing technologies that affect people's career options. - describes issues of bias and accessibility in the design of technologies.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., devise solutions to solve issues of bias in accessibility, reduce negative impacts of computing technology in everyday life).
	L1.IC.C.01 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.	provides little to no evidence in addressing the expectation(s).	- identifies the ways computing impacts personal, ethical, social, economic, and cultural practices. - defines the ways computing impacts personal, ethical, social, economic, and cultural practices.	evaluates the ways computing impacts personal, ethical, social, economic, and cultural practices.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.
	L2.IC.C.01 Evaluate the beneficial and harmful effects that computational artifacts and innovations have on society.	provides little to no evidence in addressing the expectation(s).	- identifies the beneficial and harmful effects that computational artifacts and innovations have on society. - defines the beneficial and harmful effects that computational artifacts and innovations have on society.	evaluates the beneficial and harmful effects that computational artifacts and innovations have on society.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.

IMPACTS OF COMPUTING	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
CULTURE Continued **(IC.C)**	**L1.IC.C.02** Test and refine computational artifacts to reduce bias and equity deficits.	provides little to no evidence in addressing the expectation(s).	identifies how computational artifacts reduce bias and equity deficits.	- tests computational artifacts to reduce bias and equity deficits. - refines computational artifacts to reduce bias and equity deficits.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., student creates a computational artifact that utilizes accepted accessibility standards).
	L2.IC.C.02 Evaluate the impact of equity, access, and influence on the distribution of computing resources in a global society.	provides little to no evidence in addressing the expectation(s).	provides examples for how: - equity impacts the distribution of computing resources in a global society. - access impacts the distribution of computing resources in a global society. - influence impacts the distribution of computing resources in a global society.	evaluates the impact of: - equity on the distribution of computing resources in a global society. - access on the distribution of computing resources in a global society. - influence on the distribution of computing resources in a global society.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.
	L1.IC.C.03 Demonstrate how a given algorithm applies to problems across disciplines.	provides little to no evidence in addressing the expectation(s).	identifies several disciplines a given algorithm applies to.	demonstrates how a given algorithm applies to problems across disciplines.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.
	L2.IC.C.03 Predict how computational innovations that have revolutionized aspects of our culture might evolve.	provides little to no evidence in addressing the expectation(s).	identifies computational innovations that have revolutionized aspects of our culture.	predicts how computational innovations, that have revolutionized aspects of our culture, might evolve.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.

IMPACTS OF COMPUTING	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
SOCIAL INTERACTIONS (IC.SI)	**8.IC.SI.01** Using grade appropriate content and complexity, collaborate using tools to connect with peers when creating a computational artifact. **8.IC.SI.02** Practice grade-level appropriate behavior and responsibilities while participating in an online community. Identify and report inappropriate behavior.	provides little to no evidence in addressing the expectation(s).	- collaborates with peers using a tool in an attempt to create a computational artifact. - intermittently collaborates and behaves within an online community.	- 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.	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).
	L1.IC.SI.01 Use tools and methods for collaboration.	provides little to no evidence in addressing the expectation(s).	uses basic tools and methods for collaboration.	uses a variety of tools and methods for collaboration.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard (e.g., students could compare and recommend ways different tools could help a team become more cohesive).
	L1.IC.SI.02 Practice grade-level appropriate behavior and responsibilities while participating in an online community. Identify and report inappropriate behavior.	provides little to no evidence in addressing the expectation(s).	generally practices grade-level appropriate behavior and responsibilities while participating in an online community.	- practices grade-level appropriate behavior and responsibilities while participating in an online community. - identifies and reports inappropriate behavior.	models grade-level appropriate behavior and responsibilities while participating in an online community.

IMPACTS OF COMPUTING	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
SOCIAL INTERACTIONS Continued **(IC.SI)**	**L2.IC.SI.01** Practice grade-level appropriate behavior and responsibilities while participating in an online community. Identify and report inappropriate behavior.	provides little to no evidence in addressing the expectation(s).	generally practices grade-level appropriate behavior and responsibilities while participating in an online community.	- practices grade-level appropriate behavior and responsibilities while participating in an online community. - identifies and reports inappropriate behavior.	models grade-level appropriate behavior and responsibilities while participating in an online community.
SAFETY, LAW, & ETHICS (IC.SLE)	**8.IC.SLE.01** Using grade appropriate content and complexity, describe tradeoffs between allowing information to be public and keeping information private and secure. **8.IC.SLE.02** Using grade appropriate content and complexity, describe tradeoffs between allowing information to be public and keeping information private and secure.	provides little to no evidence in addressing the expectation(s).	- lists reasons for allowing information to be public and keeping information private and secure, **and/or** With regard to positive and/or malicious intent can: - name the legal impacts associated with software development and use, - name the social impacts associated with software development and use, - name the ethical impacts associated with software development and use.	- 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.	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).
	L1.IC.SLE.01 Explain the beneficial and harmful effects that intellectual property laws can have on innovation.	provides little to no evidence in addressing the expectation(s).	- identifies a beneficial effect intellectual property laws have had on innovation, **and/or** - identifies a harmful effect intellectual property laws have had on innovation.	- identifies a beneficial effect intellectual property laws have had on innovation. - identifies a harmful effect intellectual property laws have had on innovation.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.

IMPACTS OF COMPUTING	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
SAFETY, LAW, & ETHICS Continued **(IC.SLE)**	**L2.IC.SLE.01** Debate laws and regulations that impact the development and use of software and technology.	provides little to no evidence in addressing the expectation(s).	- identifies laws and regulations that impact the development and use of software and technology. - defines laws and regulations that impact the development and use of software and technology.	debates laws and regulations that impact the development and use of software and technology.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.
	L1.IC.SLE.02 Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users.	provides little to no evidence in addressing the expectation(s).	identifies the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users.	explains the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.
	L1.IC.SLE.03 Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.	provides little to no evidence in addressing the expectation(s).	provides examples of the: - social implications of privacy in the context of safety, law, or ethics. - economic implications of privacy in the context of safety, law, or ethics.	evaluates the: - social implications of privacy in the context of safety, law, or ethics. - economic implications of privacy in the context of safety, law, or ethics.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.

IMPACTS OF COMPUTING	Standard Benchmark:	The Below Basic student:	The Basic student:	The Proficient student:	In addition to the Proficient Level, the Advanced student:
SAFETY, LAW, & ETHICS Continued **(IC.SLE)**	**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.	provides little to no evidence in addressing the expectation(s).	provides examples of the: - legal impacts associated with software development and use, including both positive and malicious intent, **or** - social impacts associated with software development and use, including both positive and malicious intent, **or** - ethical impacts associated with software development and use, including both positive and malicious intent.	discusses the: - legal impacts associated with software development and use, including both positive and malicious intent. - social impacts associated with software development and use, including both positive and malicious intent. - ethical impacts associated with software development and use, including both positive and malicious intent.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.
	L2.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.	provides little to no evidence in addressing the expectation(s).	provides examples of the: - legal impacts associated with software development and use, including both positive and malicious intent, **or** - social impacts associated with software development and use, including both positive and malicious intent, **or** - ethical impacts associated with software development and use, including both positive and malicious intent.	discusses the: - legal impacts associated with software development and use, including both positive and malicious intent. - social impacts associated with software development and use, including both positive and malicious intent. - ethical impacts associated with software development and use, including both positive and malicious intent.	demonstrates in-depth inferences and applications that go beyond the understanding or context of the standard.