Standards numbers: grade.concept.subconcept.#

Concept	Subconcept	High School - All Students	High School - CS Career Focused Students
Computing Systems	Devices	L1.CS.D.01 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.	L2.CS.D.02 Describe how internal and external parts of computing devices function to form a system.
	Hardware and Software	L1.CS.HS.01 Compare levels of abstraction and interactions between application software, system software, and hardware layers.	L2.CS.HS.01 Categorize the roles of operating system software.
		L2.CS.HS.02 Compare computer systems and determine advantages and drawbacks of each system.	L2.CS.HS.02 Compare options for building a computer systems and determine advantages and drawbacks of each piece and how it will affect the overall performance.
	Troubleshooting	L1.CS.T.01 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.	L2.CS.T.01 Illustrate ways computing systems implement logic, input, and output through hardware components.
Networks & the Internet	Network Communication & Organization	L1.NI.NCO.01 Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.	L2.NI.NCO.01 Describe the issues that impact network functionality (e.g., bandwidth, load, delay, topology).
		L1.NI.NCO.02 Compare various security measures, considering tradeoffs between the usability and security of a computing system.	L2.NI.NCO.02 Give examples to illustrate how sensitive data can be affected by malware and other attacks.
	Cybersecurity	L1.NI.C.01 Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts.	L2.NI.C.01 Compare ways software developers protect devices and information from unauthorized access.
		L1.NI.C.02 Explain tradeoffs when selecting and implementing cybersecurity recommendations.	L2.NI.C.02 Use encryption and decryption algorithms to transmit/receive an encrypted message.
Data Analysis	Storage	L1.DA.S.01 Analyze storage types and locations.	L2.DA.S.01 Translate and compare different bit representations of data types, such as characters, numbers, and images.
		L1.DA.S.02 Evaluate the tradeoffs in how data elements are organized and where data is stored.	L2.DA.S.02 Analyze file systems created for keeping track of files on the hard disk.
	Collection	L1.DA.CVT.01 Collect and analyze data.	L2.DA.CVT.01 Select data collection tools and techniques to generate data sets that support a claim or communicate information.
	Visualization & Transformation	L1.DA.CVT.02 Create interactive data visualizations using software tools to help others better understand real-world phenomena.	L2.DA.CVT.02 Use data analysis tools and techniques to identify patterns in data representing complex systems.
	Inference and Models	L1.DA.IM.01 Create computational models that represent the relationships among different elements of data collected from a phenomenon or process.	L2.DA.IM.01 Evaluate the ability of models and simulations to test and support the refinement of hypotheses. (e.g., flocking behaviors, life cycles, etc.)
		L1.AP.A.01 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.	L2.AP.A.01 Describe how artificial intelligence algorithms drive many software and physical systems (e.g., digital advertising, autonomous robots, computer vision, pattern recognition, text analysis).
			L2.AP.A.02 Describe how artificial intelligence drives many software and physical systems.

Algorithms and Programming	Algorithms		L2.AP.A.03 Critically examine and trace classic algorithms (e.g., selection sort, insertion sort, binary search, linear search).
			L2.AP.A.04 Implement an artificial intelligence algorithm to play a game against a human opponent or solve a problem.
			L2.AP.A.05 Use and adapt classic algorithms to solve computational problems.
			L2.AP.A.06 Evaluate algorithms in terms of their efficiency, correctness, and clarity.
	Variables	L1.AP.V.01 Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.	L2.AP.V.01 Compare and contrast simple data structures and their uses to simplify solutions, generalizing computational problems instead of repeatedly using primitive variables.
	Control	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.	L2.AP.C.01 Trace the execution of repetition (e.g., loops, recursion), illustrating output and changes in values of named variables.
		L1.AP.C.02 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.	
		L1.AP.C.03 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.	
	Modularity	L1.AP.M.01 Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.	L2.AP.M.01 Construct solutions to problems using student-created components, such as procedures, modules and/or objects.
		L1.AP.M.02 Create computational artifacts by systematically organizing, manipulating and/or processing data.	L2.AP.M.02 Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.
		L1.AP.M.03 Systematically design and develop programs for broad audiences by incorporating feedback from users.	L2.AP.M.03 Demonstrate code reuse by creating programming solutions using libraries and APIs.
	Program Development	L1.AP.PD.01 Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries.	L2.AP.PD.01 Plan and develop programs for broad audiences using a software life cycle process.
		L1.AP.PD.02 Evaluate and refine computational artifacts to make them more usable and accessible.	L2.AP.PD.02 Explain security issues that might lead to compromised computer programs.
		L1.AP.PD.03 Design and develop computational artifacts working in team roles using collaborative tools.	L2.AP.PD.03 Develop programs for multiple computing platforms.
		L1.AP.PD.04 Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.	L2.AP.PD.04 Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project.
			L2.AP.PD.05 Develop and use a series of test cases to verify that a program performs according to its design specifications.
			L2.AP.PD.06 Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality).

			L2.AP.PD.07 Evaluate key qualities of a program through a process such as a code review.
			L2.AP.PD.08 Compare multiple programming languages and discuss how their features make them suitable for solving different types of problems.
Impacts of Computing	Culture	L1.IC.C.01 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.	L2.IC.C.01 Evaluate the beneficial and harmful effects that computational artifacts and innovations have on society.
		L1.IC.C.02 Test and refine computational artifacts to reduce bias and equity deficits.	L2.IC.C.02 Evaluate the impact of equity, access, and influence on the distribution of computing resources in a global society.
		L1.IC.C.03 Demonstrate how a given algorithm applies to problems across disciplines.	L2.IC.C.03 Design and implement a study that evaluates or predicts how computing has revolutionized an aspect of our culture and how it might evolve (e.g., education, healthcare, art/entertainment, energy).
	Social Interactions	L2.IC.SI.01 Compare and contrast the benefits and drawbacks of social media.	L1.IC.SI.01 Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields.
	History	L1.IC.H.01 Hypothesize the impact of the innovations of computing systems for the next decade.	L2.IC.H.01 Analyze trends of computing and how those trends have changed over time.
	Safety, Law, & Ethics	L1.IC.SLE.01 Explain the beneficial and harmful effects that intellectual property laws can have on innovation.	L2.IC.SLE.01 Debate laws and regulations that impact the development and use of software.
		L1.IC.SLE.02 Explain the privacy concerns related to the collection and generation of data through automated processes (e.g., how businesses, social media, and the government collects and uses data) that may not be evident to users.	L2.IC.SLE.02 Determine ways to test the validity of information located online.
		L1.IC.SLE.03 Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.	L2.IC.SLE.03 Evaluate the social and economic consequences of how law and ethics interact with digital aspects of privacy, data, property, information, and identity.
	Community Partnerships	L1.IC.CP.01 Explore computing, software, and data storage systems in local industries.	L2.IC.CP.01 Collaborate with local industry partners to design and implement a viable mentorship.
		**Review committee: The majority of these standards are verbatium CSTA standards which is a change from draft 1. After a converstaion at our fall meeting we decided a few less standards, but the ability to show validity in the standards by standing on the shoulders of CSTA was the direction we wanted to go. Please feel free to comment, as we can change and make modifications. We felt this was our best bet in having a strong implementation guide to go along with our standards while still giving teachers the freedom to build in-depth projects to engage students in meaningful learning rather than just simply checking off each standard as they progressed through the year.	