PROPOSED DRAFT VERSION
2017 NEVADA K-12 COMPUTER SCIENCE STANDARDS

By the end of the designated grade level, students will be able to…

Concept	Subconcept	Kindergarten	Grade 1	Grade 2
Algorithms and Programming	Algorithms	**K.AP.A.1** – Model daily processes by creating and following sets of step-by-step instructions (algorithms) to complete tasks. (P4.4)		
	Program Development	**K.AP.PD.1** - Identify and fix (debug) errors in a sequence of instructions (algorithms) that includes loops. (P6.2)	**1.AP.PD.1** - Describe the iterative process of program development (including terminology, steps taken, and the logic of choices). (P7.2)	**2.AP.PD.1** - Develop plans that describe a program's sequence of events, goals, and expected outcomes. (P5.1, 7.2) **2.AP.PD.2** - Give attribution (credit) when using the ideas and creations of others while developing programs. (P7.3)
	Variables		**1.AP.V.1** - Model the way programs store and manipulate data by using numbers or other symbols to represent information. (P4.4)	
	Control			**2.AP.C.1** - Develop programs with sequences and loops, to express ideas or address a problem. (P5.2)
	Modularity			**2.AP.M.1** - Break down (decompose) the steps needed to solve a problem into a precise sequence of instructions. (P3/2)
Computing Systems	Hardware and Software	**K.CS.HS.1** - Use appropriate terminology in identifying and describing the function of common physical components of computing systems (hardware). For example: monitor, keyboard, mouse, earbuds, headphones, printer. (P7.2) **K.CS.HS.2** - Recognize some computing devices (e.g., computer, smartphone) can perform a variety of tasks and some computing devices are specialized (e.g., navigation system, game controller). (P7.2)		
	Devices		**1.CS.D.1** - Select and operate appropriate device and software to perform a variety of tasks, and recognize that users have different needs and preferences for the technology they use.(P1.1)	
	Troubleshooting			**2.CS.T.1** - Describe basic hardware and software problems using accurate terminology. (P6.2, 7.2)
Data and Analysis	Storage	**K.DA.S.1** - Recognize that data can be collected and stored on different computing devices over time. (P4.2)	**1.DA.S.1** - Recognize that a variety of data (e.g., music, video, images, text) can be stored in and retrieved from a computing device. (P4.2)	**2.DA.S.1** - Store, copy, search, retrieve, modify, and delete information using a computing device and define the

Concept	Subconcept	Kindergarten	Grade 1	Grade 2
				information stored as data. (P4.2)
Impacts of Computing	Culture	**K.IC.C.1** - Understand how computing devices have changed people's lives. (P1.1)		**2.IC.C.1** - Compare how people live and work before and after the implementation or adoption of new computing technology. (P1.1, 1.3)
	Social Interactions	**K.IC.SI.1** - Exhibit good digital citizenship using technology safely, responsibly, and ethically. (P2.1)	**1.IC.SI.1** - Work respectfully and responsibly with others online. (P2.1)	
	Safety, Law, and Ethics			**2.IC.SLE.1** - Identify safe and unsafe examples of online communications. (P2.1, 7.3)
Networks and the Internet	Cybersecurity	**K.NI.C.1** - Explain that a password helps protect the privacy of information. (P7.3)	**1.NI.C.1** - Explain why we keep personal information (e.g., name, location, phone number, home address) private. (P7.3)	**2.NI.C.1** - 1 Explain what passwords are and why we use them; use strong passwords to protect devices and information from unauthorized access. (P7.3)

Practices:

P1: Fostering an Inclusive Computing Culture
P2: Collaborating Around Computing
P3: Recognizing and Defining Computational Problems
P4: Developing and Using Abstractions
P5: Creating Computational Artifacts
P6: Testing and Refining Computational Artifacts
P7: Communication About Computing

PROPOSED DRAFT VERSION
2017 NEVADA K-12 COMPUTER SCIENCE STANDARDS

By the end of the designated grade level, students will be able to…

Concept	Subconcept	Grade 3	Grade 4	Grade 5
Algorithms and Programming	Algorithms		**4.AP.A.1** - Test, compare, and refine multiple algorithms for the same task and determine which is the most appropriate. (P3.3, 6.1-3)	
	Program Development	**3.AP.PD.1** - Debug (identify and fix) errors in an algorithm or program that includes sequences and loops. (P6.1-.2) **3.AP.PD.2** - Take on varying roles (e.g., researcher, programmer, test developer, designer, recorder) with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. (P2.2)	**4.AP.PD.1** - Test and debug (identify and fix) errors in a program or algorithm to ensure it runs as intended. (P6.1-2)	**5.AP.PD.1** – Use the iterative process to develop a program to express an idea or address a problem while considering others' perspectives and preferences. (P1.1, 5.1) **5.AP.PD.2** – Describe choices made during program development using code comments, presentations, and demonstrations. (P7.2) **5.AP.PD.3** – Observe intellectual property rights and give appropriate attribution (credit) when creating or remixing programs. (P5.2, 7.3)
	Variables	**3.AP.V.1 -** Create programs that use variables to store and modify data. (P5.2)		
	Control		**4.AP.C.1 -** Develop programs that include sequences, events, loops, and conditionals. (P5.2)	
	Modularity		**4.AP.M.1 -** Explore how complex tasks can be decomposed into simple tasks and how simple tasks can be composed into complex tasks. (P3.2)	**5.AP.M.1** - Demonstrate how to decompose a task of complexity into simple tasks and compose a simple task into tasks of complexity. (P3.2) **5.AP.M.2** - Modify, incorporate, and test portions of an existing program into their own work, to develop something new or add more advanced features. (P5.3)
Computing Systems	Hardware and Software		**4.CS.HS.1** - Model how computer hardware and software work together as a system to accomplish tasks. (P4.4)	
	Devices	**3.CS.D.1 -** Describe how internal and external parts of computing devices function to form a system. (P7.2)		
	Troubleshooting			**5.CS.T.1** - Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies. (P6.2)
Data and Analysis	Storage			

Concept	Subconcept	Grade 3	Grade 4	Grade 5
	Collection, Visualization, and Transformation	**3.DA.CVT.1** - Organize and present collected data visually to highlight relationships and support a claim. (P7.1)		
	Inference and Models		**4.DA.IM.1** - Use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate ideas. (P7.1)	**5.DA.IM.1** - Recognize how text, images, and sounds are represented as binary numbers in computing devices. (P4.1)
Impacts of Computing	Culture	**3.IC.C.1** - Discuss computing technologies that have changed the world, and express how those technologies influence and are influenced by cultural practices. (P3.1)	**4.IC.C.1** - Compare and contrast how computing has changed society from the past to the present. (P3.1)	**5.IC.C.1 –** Brainstorm ways to improve the accessibility and usability of technology products for the diverse needs and wants of users. (P1.2)
	Social Interactions			**5.IC.SI.1** - Seek diverse perspectives for the purpose of improving computational artifacts. (P1.1)
	Safety, Law, and Ethics	**3.IC.SLE.1 -** Use public domain or creative commons media, and refrain from copying or using material created by others without permission. (P7.3)		
Networks and the Internet	Cybersecurity	**3.NI.C.1** - Discuss real-world cybersecurity problems and how personal information can be protected. (P3.1)		
	Network, Communication, and Organization		**4.NI.NCO.1** - Model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the internet, and reassembled at the destination. (P4.4)	**5.NI.NCO.1 –** Explain the concept of network protocols. (P4.4) **5.NI.NCO.2 –** Identify the advantages and disadvantages of various network types (e.g., wire, WiFi, cellular data). (P4.1)

Practices:

P1: Fostering an Inclusive Computing Culture
P2: Collaborating Around Computing
P3: Recognizing and Defining Computational Problems
P4: Developing and Using Abstractions
P5: Creating Computational Artifacts
P6: Testing and Refining Computational Artifacts
P7: Communication About Computing

By the end of the designated grade band, students will be able to…

Concept	Subconcept	Grades 6-8	Grades 9-12	Advanced 9-12
Algorithms and Programming	Algorithms	**6-8.AP.A.1** - Use flowcharts and/or pseudocode to address complex problems as algorithms. (P4.1, 4.4)	**9-12.AP.A.1** - Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests. (P5.2)	**A9-12.AP.A.1 –** Describe how artificial intelligence drives many software and physical systems. (P7.2) **A9-12.AP.A.2 –** Implement an artificial intelligence algorithm to play a game against a human opponent or solve a problem. (P5.3) **A9-12.AP.A.3 –** Use and adapt classic algorithms to solve computational problems. (P4.2) **A9-12.AP.A.4 -** Evaluate algorithms in terms of their efficiency, correctness, and clarity**. (P4.2)**
	Program Development	**6-8.AP.PD.1 –** Design meaningful solutions for others, incorporating data from collaborative team members and the end user, to meet the end user's needs. (P1.1, 2.3) **6-8.AP.PD.2 –** Incorporate existing code, media, and libraries into original programs, and give attribution. (P4.2, 5.2, 7.3) **6-8.AP.PD.3 –** Systematically test and refine programs using a range of test cases. (P6.1) **6-8.AP.PD.4 –** Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts**. (P2.2)** **6-8.AP.PD.5 -** Document programs (throughout the design, development, troubleshooting, and user experience phases) in order to make them easier to follow, test, and debug by others. (P7.2)	**9-12.AP.PD.1** – Systematically design and develop programs for broad audiences by incorporating feedback from users. (P5.1) **9-12.AP.PD.2** – Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries. (P6.3, 7.3) **9-12.AP.PD.3** – Evaluate and refine computational artifacts to make them more usable by all and accessible to people with disabilities. **9-12.AP.PD.4** – Design and develop computational artifacts working in team roles using collaborative tools. (P2.4) **9-12.AP.PD.5** - Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs. (P7.2)	**A9-12.AP.PD.1** – Plan and develop programs for broad audiences using a software life cycle process. (P5.1) **A9-12.AP.PD.2** – Explain security issues that might lead to compromised computer programs. (P7.2) **A9-12.AP.PD.3** – Develop programs for multiple computing platforms. (P5.2) **A9-12.AP.PD.4** – Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project. (P2.4) **A9-12.AP.PD.5** - Develop and use a series of test cases to verify that a program performs according to its design specifications. (P6.1) **A9-12.AP.PD.6** – Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality). (P5.3) **A9-12.AP.PD.7** – Evaluate key qualities of a program through a process such as a code review. (P6.3) **A9-12.AP.PD.8** - Compare multiple programming languages and discuss how their features make them suitable for solving different types of problems. (P7.2)

Concept	Subconcept	Grades 6-8	Grades 9-12	Advanced 9-12
	Variables	**6-8.AP.V.1** - Create clearly named variables that represent different data types and perform operations on their values. (P5.1, 5.2)	**9-12.AP.V.1** - Demonstrate the use of both linked lists and arrays to simplify solutions, generalizing computational problems instead of repeatedly using simple variables. (P4.1)	**A9-12.AP.V.1** - Compare and contrast fundamental data structures and their uses. (P4.2)
	Control	**6-8.AP.C.1 -** Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. (P5.1-2)	**9-12.AP.C.1 –** Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made. (P3.2, 5.2) **9-12.AP.C.2 -** Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions. (P5.2)	**A9-12.AP.C.1** - Illustrate the flow of execution of a recursive algorithm. (P3.2)
	Modularity	**6-8.AP.M.1 –** Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. (P3.2) **6-8.AP.M.2 -** Create procedures with parameters to organize code and make it easier to reuse. (P4.1, 4.3)	**9-12.AP.M.1 –** Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. (P3.2) **9-12.AP.M.2 -** Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs. (P5.2)	**A9-12.AP.M.1 –** Construct solutions to problems using student-created components, such as procedures, modules and/or objects. (P5.2) **A9-12.AP.M.2 –** Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution. (P4.1) **A9-12.AP.M.3 –** Demonstrate code reuse by creating programming solutions using libraries and APIs. (P5.3)
Computing Systems	Hardware and Software	**6-8.CS.HS.1** - Design and evaluate projects that combine hardware and software components to collect and exchange data. (P5.1)	**9-12.CS.HS.1** - Compare levels of abstraction and interactions between application software, system software, and hardware layers. (P4.1)	**A9-12.CS.HS.**1 - Categorize the roles of operating system software. (P7.2)
	Devices	**6-8.CS.D.1 -** Recommend improvements to the design of computing devices based on an analysis of how users interact with the devices, noting that advantages may have disadvantages and unintended consequences. (P3.3)	**9-12.CS.D.1** - Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects. (P4.1)	
	Troubleshooting	**6-8.CS.T.1 -** Systematically identify and fix problems with computing devices and their components. (P6.2)	**9-12.CS.T.1 -** Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors. (P6.2)	**A9-12.CS.T.1** - Illustrate ways computing systems implement logic, input, and output through hardware components. (P7.2)
Data and Analysis	Storage	**6-8.DA.S.1** - Model encoding schema used by software tools to access data, stored as bits, into forms more easily understood by people (e.g., encoding schema include binary and ASCII). (P4.1, 4.4)	**9-12.DA.S.1 –** Translate between different bit representations of real-world phenomena, such as characters, numbers, and images (e.g., convert hexadecimal colors to decimal percentages, ASCII/Unicode representation). (P4.1) **9-12.DA.S.2** - Evaluate the tradeoffs in how data elements are organized and where data is stored. (P3.3)	
	Collection,	**6-8.DA.CVT.1** - Collect data using computational	**9-12.DA.CVT.1** - Create interactive data	**A9-12.DA.CVT.1 –** Use data analysis tools

Concept	Subconcept	Grades 6-8	Grades 9-12	Advanced 9-12
	Visualization, and Transformation	tools and transform the data to make it more meaningful and useful. (P6.3)	visualizations or alternative representations using software tools to help others better understand real-world phenomena. (P4.4)	and techniques to identify patterns in data representing complex systems. (P4.1) **A9-12.DA.CVT.2** - Select data collection tools and techniques to generate data sets that support a claim or communicate information. (P7.2)
	Inference and Models	**6-8.DA.IM.1 -** Refine computational models based on the reliability and validity of the data they generate. (P4.4, 5.3)	**9-12.DA.IM.1** - Create computational models that represent the relationships among different elements of data collected from a phenomenon, process, or model. (P4.4)	**A9-12.DA.IM.1** - Evaluate the ability of models and simulations to test and support the refinement of hypotheses. (P4.4)
Impacts of Computing	Culture	**6-8.IC.C.1 –** Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options. (P7.2) **6-8.IC.C.2** - Discuss and evaluate issues of bias and accessibility in the design of existing technologies. (P1.2)	**9-12.IC.C.1 –** Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices. (P1.2) **9-12.IC.C.2 –** Test and refine computational artifacts to reduce bias and equity deficits. (P1.2) **9-12.IC.C.3 –** Demonstrate ways a given algorithm applies to problems across disciplines. (P3.1) **9-12.IC.C.4** - Explain the potential impacts of artificial intelligence on society. (P1.1)	**A9-12.IC.C.1** – Evaluate computational artifacts to maximize their beneficial effects and minimize harmful effects on society. (P1.2, 6.1) **A9-12.IC.C.2 –** Evaluate the impact of equity, access, and influence on the distribution of computing resources in a global society. (P1.2) **A9-12.IC.C.3** - Predict how computational innovations that have revolutionized aspects of our culture might evolve. (P7.2)
	Social Interactions	**6-8.IC.SI.1** - Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact. (P2.4, 5.2)	**9-12.IC.SI.1** - Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields. (P2.4)	
	Safety, Law, and Ethics	**6-8.IC.SLE.1 –** Identify risks associated with sharing information digitally (e.g., phishing, identity theft, hacking). (P7.2) **6-8.IC.SLE.2** - Evaluate how legal and ethical issues shape computing practices. (P7.3)	**9-12.IC.SLE.1 –** Explain the beneficial and harmful effects that intellectual property laws can have on innovation. (P7.3) **9-12.IC.SLE.2 –** Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users. (P7.2) **9-12.IC.SLE.3 -** Evaluate the social and economic implications of privacy in the context of safety, law, or ethics. (P7.3)	**A9-12.IC.SLE.1** - Debate laws and regulations that impact the development and use of software. (P3.3, 7.3)
Networks and the Internet	Cybersecurity	**6-8.NI.C.1 –** Explain how physical and digital security measures protect electronic information. (P7.2) **6-8.NI.C.2** - Apply multiple methods of encryption to model the secure transmission of information. (P4.4)	**9-12.NI.C.1** – Give examples to illustrate how sensitive data can be affected by malware and other attacks. (P7.2) **9-12.NI.C.2** – Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts. (P3.3) **9-12.NI.C.3** – Compare various security measures, considering tradeoffs between the usability and security of a computing system. (P6.3)	**A9-12.NI.C.1** - Compare ways software developers protect devices and information from unauthorized access. (P7.2)

Concept	Subconcept	Grades 6-8	Grades 9-12	Advanced 9-12
			9-12.NI.C.4 - Explain tradeoffs when selecting and implementing cybersecurity recommendations. (P7.2)	
	Network, Communication, and Organization	**6-8.NI.NCO.1 -** Compare and contrast modeled protocols used in transmitting data across networks and the Internet. (P4.4)	**9-12.NI.NCO.1** - Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing. (P4.1)	**A9-12.NI.NCO.1** - Describe the issues that impact network functionality (e.g., bandwidth, load, delay, topology). (P7.2)

Practices:

P1: Fostering an Inclusive Computing Culture
P2: Collaborating Around Computing
P3: Recognizing and Defining Computational Problems
P4: Developing and Using Abstractions
P5: Creating Computational Artifacts
P6: Testing and Refining Computational Artifacts
P7: Communication About Computing