Nevada Standards Alignment with CodeX Curriculum

By the end of Grade 8, students who demonstrate understanding can:	Unit 1	Unit 2	Unit 3
Algorithms and Programming			
6-8.AP.A.1 Use flowcharts and/or pseudocode to address complex problems as algorithms.	[1]		
6-8.AP.V.1 Create clearly named variables that represent different data types and perform operations on their values.	[2]		
6-8.AP.C.1 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	[3]		
6-8.AP.M.1 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.			
6-8.AP.M.2 Create procedures with parameters to organize code and make it easier to reuse.			[4]
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.			
6-8.AP.PD.2 Incorporate existing code, media, and libraries into original programs, and give attribution.	[5]		
6-8.AP.PD.3 Systematically test and refine programs using a range of test cases.			
6-8.AP.PD.4 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.			
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.	[6]		
Computing Systems			
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.			
6-8.CS.HS.1 Design and evaluate projects that combine hardware and software components to collect and exchange data.			
6-8.CS.T.1 Systematically identify and fix problems with computing devices and their components.	[7]		
Data and Analysis			
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).			
6-8.DA.CVT.1 Collect data using computational tools and transform the data to make it more meaningful and useful.			
6-8.DA.IM.1 Refine computational models based on the reliability and validity of the data they generate.			
Impacts of Computing			
6-8.IC.C.1 Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.			

Nevada Standards Alignment with CodeX Curriculum

By the end of Grade 8, students who demonstrate understanding can:	Unit 1	Unit 2	Unit 3
6-8.IC.C.2 Discuss and evaluate issues of bias and accessibility in the design of existing technologies.			
6-8.IC.SI.1 Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.			
6-8.IC.SLE.1 Identify risks associated with sharing information digitally (e.g., phishing, identity theft, hacking).			
6-8.IC.SLE.2 Evaluate how legal and ethical issues shape computing practices.			
Networks & the Internet			
6-8.NI.NCO.1 Compare and contrast modeled protocols used in transmitting data across networks and the Internet.			
6-8.NI.C.1 Explain how physical and digital security measures protect electronic information.			
6-8.NI.C.2 Apply multiple methods of encryption to model the secure transmission of information.			

Nevada Standards Alignment with CodeX Curriculum

By the end of Grade 12, students who demonstrate understanding can:	Unit 1	Unit 2	Unit 3
Algorithms and Programming			
9-12.AP.A.1 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.	[8]		
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.		[9]	
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.			
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.		[10]	
9-12.AP.M.1 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.			
9-12.AP.M.2 Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.	[11]		
9-12.AP.PD.1 Systematically design and develop programs for broad audiences by incorporating feedback from users.			
9-12.AP.PD.2 Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries.			
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.			
9-12.AP.PD.5 Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.	[12]		
Computing Systems			
9-12.CS.D.1 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.			
9-12.CS.HS.1 Compare levels of abstraction and interactions between application software, system software, and hardware layers.			
9-12.CS.T.1 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.	[13]		
Data and Analysis			
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).			

Nevada Standards Alignment with CodeX Curriculum

By the end of Grade 12, students who demonstrate understanding can:	Unit 1	Unit 2	Unit 3
9-12.DA.S.2 Evaluate the tradeoffs in how data elements are organized and where data is stored.			
9-12.DA.CVT.1 Create interactive data visualizations or alternative representations using software tools to help others better understand real-world phenomena.			
9-12.DA.IM.1 Create computational models that represent the relationships among different elements of data collected from a phenomenon, process, or model.			
Impacts of Computing			
9-12.IC.C.1 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.			
9-12.IC.C.2 Test and refine computational artifacts to reduce bias and equity deficits.			
9-12.IC.C.3 Demonstrate ways a given algorithm applies to problems across disciplines.			
9-12.IC.C.4 Explain the potential impacts of artificial intelligence on society.			
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.			
9-12.IC.SLE.1 Explain the beneficial and harmful effects that intellectual property laws can have on innovation.			
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.			
9-12.IC.SLE.3 Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.			
Networks & the Internet			
9-12.NI.NCO.1 Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing			
9-12.NI.C.1 Give examples to illustrate how sensitive data can be affected by malware and other attacks.			
9-12.NI.C.2 Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts.			
9-12.NI.C.3 Compare various security measures, considering tradeoffs between the usability and security of a computing system.			
9-12.NI.C.4 Explain tradeoffs when selecting and implementing cybersecurity recommendations.			

Nevada Standards Alignment with CodeX Curriculum

By the end of Grade 12, students who demonstrate understanding can:	Unit 1	Unit 2	Unit 3
Algorithms and Programming			
A9-12.AP.A.1 Describe how artificial intelligence drives many software and physical systems.			
A9-12.AP.A.2 Implement an artificial intelligence algorithm to play a game against a human opponent or solve a problem.			
A9-12.AP.A.3 Use and adapt classic algorithms to solve computational problems.			
A9-12.AP.A.4 Evaluate algorithms in terms of their efficiency, correctness, and clarity.			
A9-12.AP.V.1 Compare and contrast fundamental data structures and their uses.			
A9-12.AP.C.1 Illustrate the flow of execution of a recursive algorithm.		[14]	
A9-12.AP.M.1 Construct solutions to problems using student-created components, such as procedures, modules and/or objects.		[15]	
A9-12.AP.M.2 Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.			
A9-12.AP.M.3 Demonstrate code reuse by creating programming solutions using libraries and APIs.	[16]		
A9-12.AP.PD.1 Plan and develop programs for broad audiences using a software life cycle process.			
A9-12.AP.PD.2 Explain security issues that might lead to compromised computer programs.			
A9-12.AP.PD.3 Develop programs for multiple computing platforms.			
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.			
A9-12.AP.PD.5 Develop and use a series of test cases to verify that a program performs according to its design specifications.			
A9-12.AP.PD.6 Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality).			
A9-12.AP.PD.7 Evaluate key qualities of a program through a process such as a code review.	[17]		
A9-12.AP.PD.8 Compare multiple programming languages and discuss how their features make them suitable for solving different types of problems.			
Computing Systems			
A9-12.CS.HS.1 Categorize the roles of operating system software.			

Nevada Standards Alignment with CodeX Curriculum

By the end of Grade 12, students who demonstrate understanding can:	Unit 1	Unit 2	Unit 3
A9-12.CS.T.1 Illustrate ways computing systems implement logic, input, and output through hardware components.	[18]		
Data and Analysis			
A9-12.DA.CVT.1 Use data analysis tools and techniques to identify patterns in data representing complex systems.			
A9-12.DA.CVT.2 Select data collection tools and techniques to generate data sets that support a claim or communicate information.			
A9-12.DA.IM.1 Evaluate the ability of models and simulations to test and support the refinement of hypotheses.			
Impacts of Computing			
A9-12.IC.C.1 Evaluate computational artifacts to maximize their beneficial effects and minimize harmful effects on society.			
A9-12.IC.C.2 Evaluate the impact of equity, access, and influence on the distribution of computing resources in a global society.			
A9-12.IC.C.3 Predict how computational innovations that have revolutionized aspects of our culture might evolve.			
A9-12.IC.SLE.1 Debate laws and regulations that impact the development and use of software.			
Networks & the Internet			
A9-12.NI.NCO.1 Describe the issues that impact network functionality (e.g., bandwidth, load, delay, topology).			
A9-12.NI.C.1 Compare ways software developers protect devices and information from unauthorized access.			

[1] These are introduced in the teachers' manual

[2] 3.8 introduces the use of variables
5.5 discusses descriptive variable naming

[3] Mission 6 uses nested loops but does not discuss them
Mission 9 introduces compound conditionals

[4] 9.3 Function creation is introduced
Mission 10 discusses maintaining and organizing code for reuse

[5] All lessons use libraries and every time a new one is introduced it is explained

[6] 5.5 introduces the use of comments

[7] Mission 2 introduces troubleshooting techniques as does the teachers' manual
3.5 introduces the debugger

[8] These are the remixes that begin in Mission 4

[9] 7.5 introduces the use of lists

[10] This can be the remixes depending on the rubrics the teachers use

[11] These are all of our missions starting with Mission 3

[12] This is done with pseudocodes and flowcharts that are introduced in the teachers' manual

[13] This is talked about in the teachers' manual in the troubleshooting part as well as Code Tracing Charts

[14] This would be a flowchart of a recursive algorithm like in Mission 7

[15] These are the remixes

[16] All missions use libraries and every time a new one is introduced, it is explained

[17] Code Tracing Charts are in the teachers' manual

[18] This is done with flowcharts of the code