

CompuScholar, Inc.
NVACS Alignments for Computer Science, 6th - 8th Grade
"Digital Savvy" and "Python Programming"

Nevada Course Details:

Content Area	K-12 Computer Science
Grade Range	6 - 8
Standards Documents	Nevada Academic Content Standards for CS and IT (August 2019)

CompuScholar Course Details:

Course Title:	Digital Savvy
Course ISBN:	978-0-9887070-8-5
Course Year:	2022

Course Title:	Python Programming
Course ISBN:	978-1-946113-00-9
Course Year:	2022

Course Description

The Nevada Academic Content Standards (NVACS) for Computer science represent five core concepts and seven practices. By the end of Grade 8, students should be able to demonstrate an understanding of the topics listed below.

Districts may flexibly select modules from CompuScholar's "**Digital Savvy**" and "**Python Programming**" courses to meet state requirements. The student experience can be spread over multiple years or concentrated into a single school year as desired.

*To meet all Nevada requirements, each topic has **at least one citation** from either course. Sometimes, specific standards are met by both courses, though **duplication from both courses is not necessary**.*

Citation Notes

Note 1: Citation(s) listed may represent a subset of the instances where objectives are met throughout the course.

Note 2: Citation(s) for a "Lesson" refer to the "Lesson Text" elements and associated "Activities" within the course, unless otherwise noted. The "Instructional Video" components are supplements designed to introduce or re-enforce the main lesson concepts, and the Lesson Text contains full details.

Note 3: Citation(s) to "Supplemental" or "Suppl." lessons or chapters can be found in Supplemental chapters at the end of each course.

6th - 8th Grade Computer Science Concepts

Algorithms and Programming		
Indicator and Standard	Digital Savvy Citation(s)	Python Programming Citation(s)
6-8.AP.A.1 - Use flowcharts and/or pseudocode to address complex problems as algorithms.	Chapter 22, Lesson 3	Suppl. Ch. 3, Lesson 3
6-8.AP.V.1 - Create clearly named variables that represent different data types and perform operations on their values.	Chapter 23, Lesson 1	Chapter 2
6-8.AP.C.1 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	Chapter 23, Lessons 2-3	Chapters 4, 6
6-8.AP.M.1 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.		Chapters 9, 10, 11
6-8.AP.M.2 - Create procedures with parameters to organize code and make it easier to reuse.		Chapter 9
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.	Chapters 13, 14, 25	Chapter 13
6-8.AP.PD.2 - Incorporate existing code, media, and libraries into original programs, and give attribution.	Chapters 22, 23, 25	Chapter 13
6-8.AP.PD.3 - Systematically test and refine programs using a range of test cases.	Chapter 14, Activity 3 Chapter 25, Activity 3	Chapter 5
6-8.AP.PD.4 - Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.	Chapter 13, Lessons 1-2 Chapter 14, Activity 1 Chapter 25, Activity 1	Chapter 13
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.	Chapter 14, Activities 1-2 Chapter 25, Activities 1-2	Chapter 13

Computing Systems		
Indicator and Standard	Digital Savvy Citation(s)	Python Programming Citation(s)
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.	Chapter 20, Lesson 3	Chapter 13, Activity 4
6-8.CS.HS.1 - Design and evaluate projects that combine hardware and software components to collect and exchange data.	Chapters 14, 25 (hardware may be incorporated as desired)	Chapter 13 (hardware may be incorporated as desired)
6-8.CS.T.1 - Systematically identify and fix problems with computing devices and their components.	Chapter 5, Lesson 3 Chapters 14 & 25, Act. 3	Chapter 5

Data and Analysis		
Indicator and Standard	Digital Savvy Citation(s)	Python Programming Citation(s)
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).	Suppl. Ch. 2, Lessons 1, 3	Suppl. Ch. 3, Lessons 1, 2
6-8.DA.CVT.1 - Collect data using computational tools and transform the data to make it more meaningful and useful.	Chapter 14 Suppl. Ch. 2, Lesson 5	Chapter 13
6-8.DA.IM.1 - Refine computational models based on the reliability and validity of the data they generate.	Suppl. Ch. 2, Lessons 2, 5	

Impacts of Computing		
Indicator and Standard	Digital Savvy Citation(s)	Python Programming Citation(s)
6-8.IC.C.1 - Compare tradeoffs associated with computing technologies that affect people’s everyday activities and career options.	Chapters 16 - 18 Suppl. Chapter 1	Suppl. Chapter 4
6-8.IC.C.2 - Discuss and evaluate issues of bias and accessibility in the design of existing technologies.	Suppl. Ch. 1, Lesson 3	
6-8.IC.SI.1 - Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.	Chapter 14 (team projects can use any desired data)	Chapter 13
6-8.IC.SLE.1 - Identify risks associated with sharing information digitally (e.g., phishing, identity theft, hacking).	Chapter 8, Lesson 1 Chapters 17 - 18 Suppl. Ch. 1, Lesson 2	Suppl. Ch. 2, Lessons 1, 3, 4 Suppl. Ch. 4, Lesson 2
6-8.IC.SLE.2 - Evaluate how legal and ethical issues shape computing practices.	Chapter 8, Lessons 1-3	Suppl. Ch. 2, Lessons 1, 3, 4 Suppl. Ch. 4, Lesson 2

Networks and the Internet		
Indicator and Standard	Digital Savvy Citation(s)	Python Programming Citation(s)
6-8.NI.NCO.1 - Compare and contrast modeled protocols used in transmitting data across networks and the Internet.	Chapter 6, Lesson 6	
6-8.NI.C.1 - Explain how physical and digital security measures protect electronic information.	Chapter 8, Lessons 2-3 Suppl. Ch. 2, Lesson 3	Suppl. Ch. 2, Lesson 3
6-8.NI.C.2 - Apply multiple methods of encryption to model the secure transmission of information.	Chapter 8, Lesson 3 Suppl. Ch. 2, Lesson 3	

Practices

Practice 1. Fostering an Inclusive Computing Culture		
Indicator and Standard	Digital Savvy Citation(s)	Python Programming Citation(s)
1. Include the unique perspectives of others and reflect on one's own perspectives when designing and developing computational products.	Chapters 13, 14 Suppl. Ch. 1, Lessons 1,3	Chapter 13
2. Address the needs of diverse end users during the design process to produce artifacts with broad accessibility and usability.	Chapter 14 Suppl. Ch. 1, Lesson 3	Chapter 13
3. Employ self- and peer-advocacy to address bias in interactions, product design, and development methods.	Chapters 13, 14 Suppl. Ch. 1, Lessons 1,3	Chapter 13

Practice 2. Collaborating Around Computing		
Indicator and Standard	Digital Savvy Citation(s)	Python Programming Citation(s)
1. Cultivate working relationships with individuals possessing diverse perspectives, skills, and personalities.	Chapters 13, 14, 25	Chapter 13
2. Create team norms, expectations, and equitable workloads to increase efficiency and effectiveness.	Chapters 13, 14, 25	Chapter 13
3. Solicit and incorporate feedback from, and provide constructive feedback to, team members and stakeholders.	Chapters 13, 14, 25	Chapter 13
4. Evaluate and select technological tools that can be used to collaborate on a project.	Chapters 9 - 12, 14, 16 Suppl. Ch. 3, Lessons 3,	Chapter 13

Practice 3. Recognizing and Defining Computational Problems		
Indicator and Standard	Digital Savvy Citation(s)	Python Programming Citation(s)
1. Identify complex, interdisciplinary, real-world problems that can be solved computationally.	Suppl. Ch. 1, Lesson 4 Suppl. Ch. 2, Lessons 2,5	Suppl. Ch. 3, Lesson 3 Suppl. Ch. 4, Lesson 4
2. Decompose complex real-world problems into manageable subproblems that could integrate existing solutions or procedures.	Suppl. Ch. 1, Lesson 4 Suppl. Ch. 2, Lessons 2,5	Chapter 9, Lessons 1,2 Suppl. Ch. 3, Lesson 3 Suppl. Ch. 4, Lesson 4
3. Evaluate whether it is appropriate and feasible to solve a problem computationally.	Suppl. Ch. 1, Lesson 4 Suppl. Ch. 2, Lessons 2,5	Suppl. Ch. 3, Lesson 3 Suppl. Ch. 4, Lesson 4

Practice 4. Developing and Using Abstractions		
Indicator and Standard	Digital Savvy Citation(s)	Python Programming Citation(s)
1. Extract common features from a set of interrelated processes or complex phenomena.		Chapters 9, 10, 11

2. Evaluate existing technological functionalities and incorporate them into new designs.	Chapters 14, 25	Chapter 13
3. Create modules and develop points of interaction that can apply to multiple situations and reduce complexity.		Chapters 9, 10, 11
4. Model phenomena and processes and simulate systems to understand and evaluate potential outcomes.	Suppl. Ch. 1, Lesson 4 Suppl. Ch. 2, Lessons 2,5	Chapter 11 Activity Chapters 12, 13

Practice 5. Creating Computational Artifacts

Indicator and Standard	Digital Savvy Citation(s)	Python Programming Citation(s)
1. Plan the development of a computational artifact using an iterative process that includes reflection on and modification of the plan, taking into account key features, time and resource constraints, and user expectations.	Chapters 13, 14, 25	Chapter 13
2. Create a computational artifact for practical intent, personal expression, or to address a societal issue.	Chapters 13, 14, 25	Chapter 13
3. Modify an existing artifact to improve or customize it.	Chapter 14, Activity 3 Chapter 25, Activity 3	Many Lesson "Work-With-Me" Exercises

Practice 6. Testing and Refining Computational Artifacts

Indicator and Standard	Digital Savvy Citation(s)	Python Programming Citation(s)
1. Systematically test computational artifacts by considering all scenarios and using test cases.	Chapter 14, Activity 3 Chapter 25, Activity 3	Chapter 5 Chapter 13, Activity 4
2. Identify and fix errors using a systematic process.	Chapter 14, Activity 3 Chapter 25, Activity 3	Chapter 5 Chapter 13, Activity 4
3. Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability, and accessibility.	Chapter 14, Activity 3 Chapter 25, Activity 3	Chapter 5 Chapter 13, Activity 4

Practice 7. Communicating About Computing

Indicator and Standard	Digital Savvy Citation(s)	Python Programming Citation(s)
1. Select, organize, and interpret large data sets from multiple sources to support a claim.	Chapter 14 Suppl. Ch. 2, Lesson 5	Chapter 13
2. Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose.	Chapter 13, Lesson 2 Chapters 14, 25	Chapter 1, Lesson 3 Chapter 13 and throughout the course
3. Articulate ideas responsibly by observing intellectual property rights and giving appropriate attribution.	Chapter 7, Lesson 3 Chapter 8, Lesson 5	Suppl. Ch. 2, Lesson 2