

**CompuScholar, Inc.**Correlations to the Nevada Academic Content Standards (NVACS):  
Computer Science, Grades 9 - 12***"Java Programming"*****Nevada Course Details:**

<b>Content Area</b>	Computer Science
<b>Grade Range</b>	9 - 12
<b>Standards Document</b>	<a href="#">Nevada Academic Content Standards for CS and IT (August 2019)</a>

**CompuScholar Course Details:**

<b>Course Title:</b>	Java Programming
<b>Course ISBN:</b>	978-1-946113-99-3
<b>Course Year:</b>	2022

**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.

**Concepts**

<b>Algorithms and Programming</b>	
<b>Indicator and Standard</b>	<b>Citation(s)</b>
9-12.AP.A.1 - Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.	Chapter 13 Chapter 18, Lessons 4,5
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.	Chapters 18, 19
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.	Chapters 8, 9, 12, 24
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.	Chapter 27 (student-directed project with full SDLC).

9-12.AP.M.1 - Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.	Chapters 5, 14, 15, 22, 23
9-12.AP.M.2 - Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.	Chapters 14, 15
9-12.AP.PD.1 - Systematically design and develop programs for broad audiences by incorporating feedback from users.	Chapter 27 (student-directed project includes feedback) Suppl. Chapter 2, Lesson 1
9-12.AP.PD.2 - Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries.	Chapter 1, Lesson 4
9-12.AP.PD.3 - Evaluate and refine computational artifacts to make them more usable by all and accessible to people with disabilities.	N/A (See our Web Design course)
9-12.AP.PD.4 - Design and develop computational artifacts working in team roles using collaborative tools.	Chapter 27 Suppl. Chapter 3, Lesson 4
9-12.AP.PD.5 - Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.	Chapter 27 Suppl. Chapter 1, Lessons 2, 6 Suppl. Chapter 2, Lesson 1

<b>Computing Systems</b>	
<b>Indicator and Standard</b>	<b>Citation(s)</b>
9-12.CS.D.1 - Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.	Chapter 1, Lessons 2 - 3
9-12.CS.HS.1 - Compare levels of abstraction and interactions between application software, system software, and hardware layers.	Chapter 1, Lesson 3
9-12.CS.T.1 - Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.	Chapter 11

<b>Data and Analysis</b>	
<b>Indicator and Standard</b>	<b>Citation(s)</b>
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).	Chapter 7, Lesson 2 Suppl. Chapter 1, Lesson 1
9-12.DA.S.2 - Evaluate the tradeoffs in how data elements are organized and where data is stored.	Chapters 14, 18, 19 Suppl. Chapter 1, Lesson 5
9-12.DA.CVT.1 - Create interactive data visualizations or alternative representations using software tools to help others better understand real-world phenomena.	Chapter 27 (student-directed topics) Suppl. Chapter 1, Lesson 4

9-12.DA.IM.1 - Create computational models that represent the relationships among different elements of data collected from a phenomenon, process, or model.	Chapter 27 (student-directed topics) Suppl. Chapter 1, Lesson 4
--	--

<b>Impacts of Computing</b>	
<b>Indicator and Standard</b>	<b>Citation(s)</b>
9-12.IC.C.1 - Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.	Suppl. Chapter 3
9-12.IC.C.2 - Test and refine computational artifacts to reduce bias and equity deficits.	N/A
9-12.IC.C.3 - Demonstrate ways a given algorithm applies to problems across disciplines.	Chapter 13, Lesson 3
9-12.IC.C.4 - Explain the potential impacts of artificial intelligence on society.	Suppl. Chapter 3, Lesson 3
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.	Chapter 27 Suppl. Chapter 3, Lesson 4
9-12.IC.SLE.1 - Explain the beneficial and harmful effects that intellectual property laws can have on innovation.	Chapter 1, Lesson 4
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.	Suppl. Chapter 3, Lesson 1
9-12.IC.SLE.3 - Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.	Chapter 1, Lessons 4 - 5

<b>Networks and the Internet</b>	
<b>Indicator and Standard</b>	<b>Citation(s)</b>
9-12.NI.NCO.1 - Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.	Suppl. Chapter 4, Lessons 1 - 4
9-12.NI.C.1 - Give examples to illustrate how sensitive data can be affected by malware and other attacks.	Chapter 1, Lesson 5
9-12.NI.C.2 - Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts.	Chapter 1, Lesson 5
9-12.NI.C.3 - Compare various security measures, considering tradeoffs between the usability and security of a computing system.	Chapter 1, Lesson 5
9-12.NI.C.4 - Explain tradeoffs when selecting and implementing cybersecurity recommendations.	Chapter 1, Lesson 5

## Practices

<b>Practice 1. Fostering an Inclusive Computing Culture</b>	
<b>Description</b>	<b>Citation(s)</b>
1. Include the unique perspectives of others and reflect on one's own perspectives when designing and developing computational products.	Chapter 27 Suppl. Chapter 2, Lesson 1
2. Address the needs of diverse end users during the design process to produce artifacts with broad accessibility and usability.	Chapter 27 Suppl. Chapter 2, Lesson 1
3. Employ self- and peer-advocacy to address bias in interactions, product design, and development methods.	Chapter 27 Suppl. Chapter 2, Lesson 1

<b>Practice 2. Collaborating Around Computing</b>	
<b>Description</b>	<b>Citation(s)</b>
1. Cultivate working relationships with individuals possessing diverse perspectives, skills, and personalities.	Chapter 27
2. Create team norms, expectations, and equitable workloads to increase efficiency and effectiveness.	Chapter 27
3. Solicit and incorporate feedback from, and provide constructive feedback to, team members and other stakeholders.	Chapter 27
4. Evaluate and select technological tools that can be used to collaborate on a project.	Chapter 27 Suppl. Chapter 3, Lesson 4

<b>Practice 3. Recognizing and Defining Computational Problems</b>	
<b>Description</b>	<b>Citation(s)</b>
1. Identify complex, interdisciplinary, real-world problems that can be solved computationally.	Chapter 13, Lessons 3, 4 Chapter 20 Suppl. Chapter 1, Lesson 4 Suppl. Chapter 3, Lesson 3
2. Decompose complex real-world problems into manageable subproblems that could integrate existing solutions or procedures.	Chapter 13, Lessons 3, 4 Chapters 14, 15, 22, 23 Suppl. Chapter 1, Lesson 4 Suppl. Chapter 3, Lesson 3
3. Evaluate whether it is appropriate and feasible to solve a problem computationally.	Chapter 33 Suppl. Chapter 3, Lesson 3

<b>Practice 4. Developing and Using Abstractions</b>	
<b>Description</b>	<b>Citation(s)</b>
1. Extract common features from a set of interrelated processes or complex phenomena.	Chapters 14, 15, 16, 22, 23

2. Evaluate existing technological functionalities and incorporate them into new designs.	Objects from the Java Class Library are incorporated into programs throughout the course (e.g. Chapter 7, Lesson 3)
3. Create modules and develop points of interaction that can apply to multiple situations and reduce complexity.	Chapters 13, 14, 15, 16, 22, 23
4. Model phenomena and processes and simulate systems to understand and evaluate potential outcomes.	Suppl. Chapter 1, Lesson 4

### Practice 5. Creating Computational Artifacts

Description	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.	Chapter 27 Suppl. Chapter 2, Lesson 1
2. Create a computational artifact for practical intent, personal expression, or to address a societal issue.	Chapter 27
3. Modify an existing artifact to improve or customize it.	Activities in Chapters 15, 16, 18, etc.

### Practice 6. Testing and Refining Computational Artifacts

Description	Citation(s)
1. Systematically test computational artifacts by considering all scenarios and using test cases.	Chapter 11, Lesson 1 Chapter 27, Lesson 3
2. Identify and fix errors using a systematic process.	Chapter 11, Lesson 1 Chapter 27, Lesson 3
3. Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability, and accessibility.	Chapter 11, Lesson 1 Chapter 27, Lesson 3

### Practice 7. Communicating About Computing

Description	Citation(s)
1. Select, organize, and interpret large data sets from multiple sources to support a claim.	Chapter 27 (if desired) Suppl. Chapter 1, Lesson 4
2. Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose.	Chapter 13, Lessons 1 - 2 Chapter 27, Lesson 2 Suppl. Chapter 1, Lessons 2, 6 Suppl. Chapter 2, Lesson 1
3. Articulate ideas responsibly by observing intellectual property rights and giving appropriate attribution.	Chapter 1, Lesson 4

## Advanced\* Concepts

\* These Advanced Computer Science Standards are considered to be higher level concepts that may be used by your advanced students, incorporated into upper level courses, and/or used in Career and Technical Education (CTE) programs.

<b>Algorithms and Programming</b>	
<b>Indicator and Standard</b>	<b>Citation(s)</b>
A9-12.AP.A.1 - Describe how artificial intelligence drives many software and physical systems.	Suppl. Chapter 3, Lesson 3
A9-12.AP.A.2 - Implement an artificial intelligence algorithm to play a game against a human opponent or solve a problem.	Chapter 18 Activity Chapter 27 (student-led project) Suppl. Chapter 3, Lesson 3
A9-12.AP.A.3 - Use and adapt classic algorithms to solve computational problems.	Chapter 13 Chapter 18, Lessons 4,5
A9-12.AP.A.4 - Evaluate algorithms in terms of their efficiency, correctness, and clarity.	Chapters 13, 20, 33
A9-12.AP.V.1 - Compare and contrast fundamental data structures and their uses.	Chapters 18, 19, 21 Suppl. Chapter 1, Lesson 5
A9-12.AP.C.1 - Illustrate the flow of execution of a recursive algorithm.	Chapter 24, Lesson 1
A9-12.AP.M.1 - Construct solutions to problems using student-created components, such as procedures, modules and/or objects.	Chapters 5, 14, 15, 22, 23
A9-12.AP.M.2 - Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.	Chapters 20, 24
A9-12.AP.M.3 - Demonstrate code reuse by creating programming solutions using libraries and APIs.	Java class library used throughout the course
A9-12.AP.PD.1 - Plan and develop programs for broad audiences using a software life cycle process.	Chapter 27 Suppl. Chapter 2, Lesson 1
A9-12.AP.PD.2 - Explain security issues that might lead to compromised computer programs.	Chapter 1, Lesson 5
A9-12.AP.PD.3 - Develop programs for multiple computing platforms.	Java inherently runs on multiple 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.	Chapter 27
A9-12.AP.PD.5 - Develop and use a series of test cases to verify that a program performs according to its design specifications.	Chapter 27, Lesson 3 Suppl. Chapter 2, Lesson 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).	Chapter 27, Lesson 3 (project includes regression testing)
A9-12.AP.PD.7 - Evaluate key qualities of a program through a process such as a code review.	Chapter 11, Lesson 1

A9-12.AP.PD.8 - Compare multiple programming languages and discuss how their features make them suitable for solving different types of problems.	Chapter 2, Lesson 1
---	---------------------

<b>Computing Systems</b>	
<b>Indicator and Standard</b>	<b>Citation(s)</b>
A9-12.CS.HS.1 - Categorize the roles of operating system software.	Chapter 1, Lesson 3
A9-12.CS.T.1 - Illustrate ways computing systems implement logic, input, and output through hardware components.	Chapter 1, Lesson 2

<b>Data and Analysis</b>	
<b>Indicator and Standard</b>	<b>Citation(s)</b>
A9-12.DA.CVT.1 - Use data analysis tools and techniques to identify patterns in data representing complex systems.	Suppl. Chapter 1, Lesson 4
A9-12.DA.CVT.2 - Select data collection tools and techniques to generate data sets that support a claim or communicate information.	Suppl. Chapter 1, Lesson 4
A9-12.DA.IM.1 - Evaluate the ability of models and simulations to test and support the refinement of hypotheses.	Suppl. Chapter 1, Lesson 4

<b>Impacts of Computing</b>	
<b>Indicator and Standard</b>	<b>Citation(s)</b>
A9-12.IC.C.1 - Evaluate computational artifacts to maximize their beneficial effects and minimize harmful effects on society.	Suppl. Chapter 3, Lessons 1 - 2
A9-12.IC.C.2 - Evaluate the impact of equity, access, and influence on the distribution of computing resources in a global society.	N/A (see Digital Savvy)
A9-12.IC.C.3 - Predict how computational innovations that have revolutionized aspects of our culture might evolve.	Suppl. Chapter 3, Lessons 2 - 3
A9-12.IC.SLE.1 - Debate laws and regulations that impact the development and use of software.	Chapter 1, Lessons 4 - 5

<b>Networks and the Internet</b>	
<b>Indicator and Standard</b>	<b>Citation(s)</b>
A9-12.NI.NCO.1 - Describe the issues that impact network functionality (e.g., bandwidth, load, delay, and topology).	Suppl. Chapter 4, Lessons 1 - 4
A9-12.NI.C.1 - Compare ways software developers protect devices and information from unauthorized access.	Chapter 1, Lesson 5