

## CompuScholar, Inc.

# Alignment to the Oklahoma Academic Standards (OAS) for Computer Science (9th - 12th Grade) Java Programming

### Oklahoma Standards Information:

CS Page	<a href="#">Oklahoma Computer Science Standards</a>
Standards Link:	<a href="#">2023 Oklahoma Academic Standards for Computer Science</a>

### CompuScholar Course Details:

Course Title:	<a href="#">Java Programming</a>
Course ISBN:	978-1-946113-99-3
Course Year:	2024

### Course Description

CompuScholar's *Java Programming* curriculum is commonly used for **Computer Science I** and **Computer Science II** courses in many states. It is also **endorsed by the College Board** as fully aligned with **AP CS A**. The course covers introductory coding concepts, decision-making, iteration, data structures, algorithms (including searching and sorting), OOP, recursion, and other classic CS topics.

### Oklahoma Subject Codes

This course is best used as a primary resource for the following subjects:

- 2531 - Computer Science I**
- 2532 - Computer Science II**
- 2535 - AP Computer Science A (College-Board Endorsed)**

### Oklahoma Academic Standards (OAS) for Computer Science (High School)

Level 1 Requirements in White
Level 2 Requirements in Blue

**Note 1:** Citation(s) for a "Lesson" refer to the "**Lesson Text**" page where instruction of concepts is found. Additional hands-on practice can be found in the nearby "**Chapter Homework**" and "**Chapter Activity**" pages within that chapter.

**Note 2:** The "Instructional Video" components are optional supplements designed to introduce or reinforce the main lesson concepts and are not cited as standards-bearing content.

**Note 3:** Citation(s) to "Supplemental" or "Suppl." Chapters refer to Supplemental Chapters found at the end of the course.

<b>Computing Systems</b>	<b>CITATIONS</b>
<b>Devices</b>	
L1.CS.D.01 Model how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.	Chapter 5, Lesson 1 Chapter 14, Lesson 1 Chapter 15, Lesson 3 Chapter 23, Lesson 1
<b>Hardware &amp; Software</b>	
L1.CS.HS.01 Analyze the levels of abstraction and interactions between application software, system software, and hardware.	Chapter 1, Lessons 2, 3
L2.CS.HS.01 Identify and categorize the roles of a variety of operating system software.	Chapter 1, Lesson 3
<b>Troubleshooting</b>	
L1.CS.T.01 Develop and apply criteria for the systematic discovery of errors and systematic strategies for the correction of errors in computing systems.	Chapter 11, Lessons 1, 2 Chapter 27, Lesson 3
L2.CS.T.01 Illustrate how understanding the ways hardware components facilitate logic, input, output, and storage in computing systems will support troubleshooting.	Chapter 1, Lesson 2

<b>Network &amp; The Internet</b>	<b>CITATIONS</b>
<b>Network Communication &amp; Organization</b>	
L1.NI.NCO.01 Evaluate the scalability and reliability of networks by identifying and illustrating the basic components of computer networks (e.g., routers, switches, servers, etc.) and network protocols (e.g., IP, DNS).	Supplemental Chapter 4, Lessons 1 - 5
L2.NI.NCO.01 Describe the issues that impact network functionality (e.g., bandwidth, load, latency, topology).	Supplemental Chapter 4, Lessons 2, 4
<b>Cybersecurity</b>	
L1.NI.CY.01 Compare physical and cybersecurity measures by evaluating trade-offs between the usability and security of a computing system and the risks of an attack.	Chapter 1, Lesson 5 Supplemental Chapter 3, Lesson 1
L2.NI.CY.01 Compare and refine ways in which software developers protect devices and information from unauthorized access.	Chapter 1, Lesson 5 Supplemental Chapter 3, Lesson 1
L1.NI.CY.02 Recommend security measures to address various scenarios based on information security principles.	Chapter 1, Lesson 5 Supplemental Chapter 3, Lesson 1
L1.NI.CY.03 Explain trade-offs when selecting and implementing cybersecurity recommendations from multiple perspectives, such as the user, enterprise, and government.	N/A

<b>Data Analysis</b>	<b>CITATIONS</b>
<b>Storage</b>	
L1.DA.S.01 Convert and compare different bit representations of data types, such as characters, numbers, and images	Chapter 3, Lesson 1 Chapter 4, Lesson 3 Chapter 7, Lessons 2, 4 Supplemental Chapter 1, Lesson 1
L1.DA.S.02 Evaluate the trade-offs in how data is organized and stored digitally.	Chapter 5, Lessons 2, 4 Chapters 18, 19, 21, 25 Supplemental Chapter 1, Lesson 5
<b>Collection, Visualization &amp; Transformation</b>	
L1.DA.CVT.01 Use tools and techniques to locate, collect, and create visualizations of small and largescale data sets (e.g., paper surveys and online data sets).	Chapter 27, Activities 1, 2 Supplemental Chapter 1, Lesson 4
L2.DA.CVT.01 Use data analysis tools and techniques to identify patterns from complex real-world data.	Chapter 27, Activities 1, 2 Supplemental Chapter 1, Lesson 4
L2.DA.CVT.02 Generate data sets that use a variety of data collection tools and analysis techniques to support a claim and/or communicate information.	Chapter 27, Activities 1, 2 Supplemental Chapter 1, Lesson 4
<b>Inference &amp; Models</b>	
L1.DA.IM.01 Illustrate and explain the relationships between collected data elements using computational models.	Chapter 27, Activities 1, 2 Supplemental Chapter 1, Lesson 4
L2.DA.IM.01 Use models and simulations to help plan, conduct, and refine investigations.	Chapter 27, Activities 1, 2 Supplemental Chapter 1, Lesson 4

<b>Algorithms &amp; Programming</b>	<b>CITATIONS</b>
<b>Algorithms</b>	
L1.AP.A.01 Create a prototype that uses algorithms (e. g., searching, sorting, finding shortest distance) to provide a possible solution for a real- world problem.	Chapters 13, 20, 24
L2.AP.A.01 Model and use appropriate terminology to describe how artificial intelligence algorithms drive many software and physical systems (e.g., autonomous robots, pattern recognition, text analysis).	Supplemental Chapter 3, Lesson 3
L2.AP.A.02 Develop an artificial intelligence algorithm to play a game against a human opponent or solve a real-world problem.	Chapter 27, Activities 1, 2
L2.AP.A.03 Critically examine and trace classic algorithms (e.g., selection sort, insertion sort, binary search, linear search).	Chapter 20
L2.AP.A.04 Evaluate algorithms (e.g., sorting, searching) in terms of their efficiency and clarity.	Chapters 20, 33
<b>Variables</b>	
L1.AP.V.01 Demonstrate the use of lists (e.g., arrays) to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.	Chapters 18, 19

L2.AP.V.01 Compare and contrast data structures and their uses (e.g., lists, stacks, queues).	Chapters 18, 19, 21 Supplemental Chapter 1, Lesson 5
<b>Control</b>	
L1.AP.C.01 Justify the selection of specific control structures (e.g., sequence, conditionals, repetition, procedures) considering program efficiencies such as readability, performance, and memory usage.	Chapters 8, 12, 15, 24
L2.AP.C.01 Model the execution of repetition (e.g., loops, recursion) of an algorithm illustrating output and changes in values of named variables.	Chapters 12, 13, 24
<b>Modularity</b>	
L1.AP.M.01 Decompose problems into procedures using systematic analysis and design.	Chapter 13, Lessons 1, 2 Chapter 14, Lesson 5 Chapter 15, Lesson 1 Chapter 27, Lesson 2
L2.AP.M.01 Construct solutions to problems using student-created components (e.g., procedures, modules, objects).	Chapters 14, 15, 16, 17, 22, 23, 27
L1.AP.M.02 Create computational artifacts by systematically organizing, manipulating and/or processing data.	Chapters 18, 19, 21, 22, 23, 25, 27
L2.AP.M.02 Design or redesign a solution to a large-scale computational problem by identifying generalizable patterns.	Chapters 20, 24, 27 Supplemental Chapter 3, Lesson 3
L2.AP.M.03 Create programming solutions by reusing existing code (e.g., libraries, Application Programming Interface (APIs), code repositories).	Chapter 5, Lesson 1 Chapter 7, Lesson 3 Chapter 19, Lessons 1, 2
<b>Program Development</b>	
L1.AP.PD.01 Create software that will provide solutions to a variety of users using a software development process.	Chapter 27
L2.AP.PD.01 Create software that will provide solutions to a variety of users using multiple software development processes.	Chapter 27
L1.AP.PD.02 Evaluate a variety of software licensing schemes (e.g., open source, freeware, commercial) and discuss the advantages and disadvantages of each scheme in software development.	Chapter 1, Lesson 4 Supplemental Chapter 3, Lesson 5
L2.AP.PD.02 Design software in a project team environment using integrated development environments (IDEs), versioning systems, and collaboration systems.	Chapter 27, 28, 29 (excluding versioning and collaboration systems)
L1.AP.PD.03 While working in a team, develop, test, and refine event-based programs that solve practical problems or allow self-expression.	Chapter 27
L2.AP.PD.03 Develop programs for multiple computing platforms.	Chapter 2, Lessons 1, 2 (Java is cross-platform and all student work runs on multiple platforms)
L1.AP.PD.04 Using visual aids and documentation, illustrate the design elements and data flow (e.g., flowcharts, pseudocode) of the development of a complex program.	Chapter 13, Lessons 1, 2 Chapter 27, Lesson 2 Supplemental Chapter 1, Lesson 6
L2.AP.PD.04 Systematically examine code for correctness, usability, readability, efficiency, portability, and scalability through peer review.	Chapter 11, Lesson 1 Chapter 27, Lesson 3 Chapter 27, Activities 2, 3

L1.AP.PD.05 Evaluate and refine computational artifacts to make them more user-friendly, efficient and/or accessible.	Chapter 13, Lesson 3 Chapter 22, Lessons 1, 2, 3
L2.AP.PD.05 Develop and use a series of test cases to verify that a program performs according to its design specifications.	Chapter 27, Lesson 3
L2.AP.PD.06 Explain security issues that might lead to compromised computer programs.	Chapter 1, Lesson 5 Supplemental Chapter 3, Lesson 1
L2.AP.PD.07 Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality).	Chapter 17 (incremental additions to a program) Chapter 27, Lesson 3 (Iterative testing)

<b>Impacts of Computing</b>	<b>CITATIONS</b>
<b>Culture</b>	
L1.IC.CU.01 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.	Chapter 1, Lesson 4 Supplemental Chapter 3, Lessons 2 - 4
L2.IC.CU.01 Evaluate the beneficial and harmful effects that computational artifacts and innovations have on society.	Chapter 1, Lesson 4 Supplemental Chapter 3, Lessons 2 - 4
L1.IC.CU.02 Test and refine computational artifacts to ensure access to a variety of user audiences.	N/A
L2.IC.CU.02 Evaluate the impact of location and user audience on the distribution of computing resources in a global society.	N/A
L1.IC.CU.03 Demonstrate ways a given algorithm can help solve computational problems across disciplines.	Chapter 13, Lessons 3, 4 Chapters 20, 24
L2.IC.CU.03 Design and implement a study that evaluates or predicts how creating, testing, and refining computational artifacts has revolutionized an aspect of our culture and how it might evolve (e.g., education, healthcare, art/entertainment, energy).	Supplemental Chapter 3, Lessons 2, 3
<b>Social Interactions</b>	
L1.IC.SI.01 Demonstrate and debate how computing increases and decreases connectivity and communication among people of various cultures.	Supplemental Chapter 3, Lesson 2
<b>Safety, Law &amp; Ethics</b>	
L1.IC.SLE.01 Describe the beneficial and harmful effects that intellectual property laws can have on innovation.	Chapter 1, Lesson 4 Supplemental Chapter 3, Lesson 5
L2.IC.SLE.01 Debate laws and regulations that impact the development and use of software.	Chapter 1, Lessons 4, 5
L1.IC.SLE.02 Describe and discuss the privacy concerns related to the large-scale collection and analysis of information about individuals (e.g., how websites collect and uses data) that may not be evident to users.	Chapter 1, Lesson 5 Supplemental Chapter 3, Lesson 1
L1.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.	Chapter 1, Lessons 4, 5 Supplemental Chapter 3, Lesson 1