CompuScholar, Inc.

Alignment to the California Computer Science Standards

9th - 12th Grade

California Standards Information:

CS Page	California Computer Science Education Page
Standards Link:	K-12 Computer Science Standards (XLSX)

CompuScholar Courses in this Grade Band:

Course Title:	**Digital Savvy**, ISBN 978-0-9887070-8-5 Course Description and Syllabus
Course Title:	**Web Design**, ISBN 978-0-9887070-3-0 Course Description and Syllabus
Course Title:	**Python Programming**, ISBN 978-1-946113-00-9 Course Description and Syllabus
Course Title:	**Java Programming (Abridged)**, ISBN 978-0-9887070-4-7 Course Description and Syllabus
Course Title:	**Java Programming (AP)**, ISBN 978-0-9887070-2-3 Course Description and Syllabus
Course Title:	**Windows Programming with C#**, ISBN 978-0-9887070-0-9 Course Description and Syllabus
Course Title:	**Unity Game Programming**, ISBN 978-0-9887070-7-8 Course Description and Syllabus

California's Computer Science standards are broken into "Core" and "Specialty" categories. The "Core" elements represent foundational skills expected to be taught in any introductory Computer Science course, while the "Specialty" skills may be deferred to more advanced courses.

Any CompuScholar "Programming" course can be used to meet either "Core" or "Specialty" requirements. Material from Digital Savvy and Web Design may be included as desired for relevant topics.

California "Core" Computer Science Standards (High School)

Computing Systems	**COMPUSCHOLAR ALIGNMENT**
Devices	
9-12.CS.1 Describe ways in which abstractions hide the underlying implementation details of computing systems to simplify user experiences.	Our courses contain lessons that describe how various hardware components (e.g. CPU, RAM, disk drives) encapsulate computing tasks such as processing, short-term storage and long-term

Hardware & Software	
9-12.CS.2 Compare levels of abstraction and interactions between application software, system software, and hardware.	Our courses describe the relationships between hardware, operating systems, device drivers, and a variety of end-user applications.
Troubleshooting	
9-12.CS.3 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.	Our courses contain dedicated troubleshooting and debugging information for relevant technology. The programming courses describe how to use a variety of debugging approaches, including code analysis, tracing (logging) and setting breakpoints in a debugger. Best practices and common troubleshooting tips are provided as needed.

Network & The Internet	**COMPUSCHOLAR ALIGNMENT**
Network Communication & Organization	
9-12.NI.4 Describe issues that impact network functionality.	Network design considerations, including scalability, bandwidth, security and points of failure are addressed where relevant.
9-12.NI.5 Describe the design characteristics of the Internet.	Our courses contain relevant descriptions of major Internet components, including clients, servers, protocols, ISPs and addressing.
Cybersecurity	
9-12.NI.6 Compare and contrast security measures to address various security threats.	Our courses contain chapters or lessons on security topics, including physical vs. electronic security mechanisms where relevant.
9-12.NI.7 Compare and contrast cryptographic techniques to model the secure transmission of information.	Our courses contain chapters or lessons on relevant security topics such as encryption (including SSL/TLS) and establishing defenses against mal-ware and viruses.

Data Analysis	**COMPUSCHOLAR ALIGNMENT**
Storage	
9-12.DA.8 Translate between different representations of data abstractions of real-world phenomena, such as characters, numbers, and images.	Our courses cover numbering systems such as binary, decimal and hexadecimal. The encoding of data, including ASCII character and color representations is discussed where relevant.
9-12.DA.9 Describe tradeoffs associated with how data elements are organized and stored.	Our courses cover representation of data in multiple formats, from simple encoding of data types up through data structures and object definitions with properties. Choices of data type, data structure, or object definitions have pros and cons and may impact the desirable characteristics of the program or the maintainability of the underlying code.

Collection, Visualization & Transformation	
9-12.DA.10 Create data visualizations to help others better understand real-world phenomena.	Our course contain opportunities to explore and represent real-world data in the form of equations, charts and graphs and similar tools.
Inference & Models	
9-12.DA.11 Refine computational models to better represent the relationships among different elements of data collected from a phenomenon or process.	Student-driven projects give opportunities for analysis and representation of real-world data.

Algorithms & Programming	**COMPUSCHOLAR ALIGNMENT**
Algorithms	
9-12.AP.12 Design algorithms to solve computational problems using a combination of original and existing algorithms.	Our courses describe how to use flowcharts to design algorithms to solve specific problems. Common sorting and searching algorithms or game-specific AI
Variables	
9-12.AP.13 Create more generalized computational solutions using collections instead of repeatedly using simple variables.	Our programming courses cover simple data structures such as arrays and lists. Students will use each data structure in hands-on projects.
Control	
9-12.AP.14 Justify the selection of specific control structures by identifying tradeoffs associated with implementation, readability, and performance.	Our courses cover traditional flow control structures (conditionals, loops, functions) and the trade-offs in design, including selecting between appropriate flow control logic.
9-12.AP.15 Iteratively design and develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.	Our courses contain opportunities for students to participate in simple SDLC stages to iteratively design and develop digital or computational artifacts of interest.
Modularity	
9-12.AP.16 Decompose problems into smaller subproblems through systematic analysis, using constructs such as procedures, modules, and/or classes.	Our courses cover Object-Oriented Programming (OOP), modular programming with functions, and functional decomposition of complex tasks down to manageable logical blocks.
9-12.AP.17 Create computational artifacts using modular design.	Our courses include opportunities for students to build their own functions/methods and objects.
9-12.AP.18 Systematically design programs for broad audiences by incorporating feedback from users.	Our courses contain opportunities for students to participate in simple SDLC stages, including incorporation of feedback from users to improve the project.
9-12.AP.19 Explain the limitations of licenses that restrict use of computational artifacts when using resources such as libraries.	Our courses explain and use relevant, language-specific libraries such as the Java class library, the .NET framework or Python modules, with licensing considerations discussed as needed.

Program Development	
9-12.AP.20 Iteratively evaluate and refine a computational artifact to enhance its performance, reliability, usability, and accessibility.	Our courses contain team projects that allow students to define, design, build and test a unique project using standard SDLC stages and traditional requirements, design and test documentation. Accessibility concerns are addressed as needed.
9-12.AP.21 Design and develop computational artifacts working in team roles using collaborative tools.	Our courses contain team projects in which students will experience standard software engineering roles and development life-cycles. Students will write project plans, requirements and design documents, and follow systematic test plans.
9-12.AP.22 Document decisions made during the design process using text, graphics, presentations, and/or demonstrations in the development of complex	Our courses contain team projects with specific design stages and documentation, including visualization tools where appropriate.

Impacts of Computing	**COMPUSCHOLAR ALIGNMENT**
Culture	
9-12.IC.23 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.	Our courses contain relevant lessons on the global impact of computing, ethical computing concepts, intellectual property and licensing, careers in computing, etc.
9-12.IC.24 Identify impacts of bias and equity deficit on design and implementation of computational artifacts and apply appropriate processes for evaluating issues of bias.	Students are presented with examples and projects that are free from bias and inequity, and are similarly encouraged to follow ethical practices in their own development.
9-12.IC.25 Demonstrate ways a given algorithm applies to problems across disciplines.	Our courses cover a variety of algorithms that can be applied to real-world situations in mathematics, physics, economics, etc.
9-12.IC.26 Study, discuss, and think critically about the potential impacts and implications of emerging technologies on larger social, economic, and political structures, with evidence from credible sources.	Our courses include lessons on trends in future development, as well as opportunities to study how Artificial Intelligence might be applied in a variety of situations.
Social Interactions	
9-12.IC.27 Use collaboration tools and methods to increase connectivity with people of different cultures and careers.	Our courses contain relevant lessons on the global impact of computing. Students may additionally learn about Internet communication tools and social
Safety, Law & Ethics	
9-12.IC.28 Explain the beneficial and harmful effects that intellectual property laws can have on innovation.	Our courses cover intellectual property laws, copyright considerations and various types of software licensing.

9-12.IC.29 Explain the privacy concerns related to the collection and generation of data through automated processes.	Students are taught to minimize their personal information shared online. They will learn about their personal digital footprint and understand that digital identities and online actions have long-term or permanent consequences.
9-12.IC.30 Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.	Our courses cover computing ethics, copyrights, intellectual property, personal privacy and computer security. Relevant laws are discussed in each lesson.

California "Specialty" Computer Science Standards (High School)

Computing Systems	**COMPUSCHOLAR ALIGNMENT**
Hardware & Software	
9-12S.CS.1 Illustrate ways computing systems implement logic through hardware components.	Our courses contain appropriate discussions of the major hardware components of a computing system.
9-12S.CS.2 Categorize and describe the different functions of operating system software.	Our courses discuss relevant operating system features and tasks including basic file and application management.

Network & The Internet	**COMPUSCHOLAR ALIGNMENT**
Network Communication & Organization	
9-12S.NI.3 Examine the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.	Our courses describe networking components and common network topology. IP addresses, MAC addresses and URLs are introduced for identification of devices and online resources.
9-12S.NI.4 Explain how the characteristics of the Internet influence the systems developed on it.	Our courses, where relevant, describe Internet protocols, DNS servers, domain registration and similar systems.
Cybersecurity	
9-12S.NI.5 Develop solutions to security threats.	Our courses contain lessons on relevant strategies to protect computers, business data and online communications.
9-12S.NI.6 Analyze cryptographic techniques to model the secure transmission of information.	Our courses contain chapters or lessons on relevant security topics such as encryption (including SSL/TLS), online safety and privacy and protecting personal information online.

Data Analysis	**COMPUSCHOLAR ALIGNMENT**
Collection, Visualization & Transformation	
9-12S.DA.7 Select and use data collection tools and techniques to generate data sets.	Our team projects and other labs give students opportunities to research topics, obtain data sets, and produce digital artifacts or apps to visualize and explain the aggregated information.

9-12S.DA.8 Use data analysis tools and techniques to identify patterns in data representing complex systems.	Our team projects and other labs give students opportunities to visualize and explain data by spreadsheet, charts & graphs or programmatic display, where relevant.
Inference & Models	
9-12S.DA.9 Evaluate the ability of models and simulations to test and support the refinement of hypotheses.	Our courses contain opportunities for student to build models of real-world phenomena and simulate or predict results. Students are encouraged to alter or modify input data to observe the impact on resulting output and verify initial assumptions.

Algorithms & Programming	**COMPUSCHOLAR ALIGNMENT**
Algorithms	
9-12S.AP.10 Describe how artificial intelligence drives many software and physical systems.	Our courses contain context-appropriate descriptions of AI algorithms such as game AI and applications to real-world problems (e.g. self-driving cars).
9-12S.AP.11 Implement an algorithm that uses artificial intelligence to overcome a simple challenge.	Our courses contain context-appropriate opportunities for students to study or implement AI algorithms to perform specific tasks (e.g. game AI or other real-world problems).
9-12S.AP.12 Implement searching and sorting algorithms to solve computational problems.	Our courses contain appropriate analysis and comparison of algorithms, including multiple sorting and searching approaches.
9-12S.AP.13 Evaluate algorithms in terms of their efficiency.	Our courses contain appropriate analysis and comparison of algorithms, including trade-offs in performance, coding complexity, and accuracy of results.
Variables	
9-12S.AP.14 Compare and contrast fundamental data structures and their uses.	Our programming courses cover lists, stacks and queues and the trade-offs involved with each representation. Students will use each data structure in hands-on projects.
Control	
9-12S.AP.15 Demonstrate the flow of execution of a recursive algorithm.	Our coding courses describe and demonstrate recursive algorithms and execution flow, where relevant.
Modularity	
9-12S.AP.16 Analyze a large-scale computational problem and identify generalizable patterns or problem components that can be applied to a solution.	Our courses compare and contrast standard algorithms for large scale sorting and searching. They additionally cover relevant AI algorithms to implement specific tasks or features.

Program Development	
9-12S.AP.17 Construct solutions to problems using student-created components, such as procedures, modules, and/or objects.	Our courses contain multiple opportunities for students to create their own functions and objects.
9-12S.AP.18 Demonstrate code reuse by creating programming solutions using libraries and APIs.	Our coding courses use a variety of libraries and APIs appropriate for the language (e.g. Java Class Library, .NET Framework, Unity SDK).
9-12S.AP.19 Plan and develop programs for broad audiences using a specific software life cycle process.	Our courses contain team projects that allow students to define, design, build and test a unique project using standard SDLC stages.
9-12S.AP.20 Develop programs for multiple computing platforms.	Our courses teach multiple languages, and those with easy cross-platform support (e.g. Java or HTML or Unity) are clearly defined as such, so student projects can be run on computers with different operating systems.
9-12S.AP.21 Identify and fix security issues that might compromise computer programs.	Our courses contain chapters or lessons on relevant security topics such as SSL/TLS, mal-ware and viruses, and protection of business data.
9-12S.AP.22 Develop and use a series of test cases to verify that a program performs according to its design specifications.	Our courses contain team projects that include a testing phase using a written test plan. Students will receive feedback from peers and incorporate that feedback into the final project.
9-12S.AP.23 Modify an existing program to add additional functionality and discuss intended and unintended implications.	Our courses contain multiple projects where starting code is provided and students will expand or complete the initial project. Regression testing is described in chapters of debugging and testing.
9-12S.AP.24 Evaluate key qualities of a program through a process such as a code review.	Each coding course contains a debugging chapter that includes code review as a methodical process for evaluating a program.
9-12S.AP.25 Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (e.g., code documentation) while developing software within a group.	Students will use IDEs and relevant collaborative tools and practices to develop numerous projects throughout each course. Team projects are included for group work.
9-12S.AP.26 Compare multiple programming languages, and discuss how their features make them suitable for solving different types of problems.	Each coding course discusses common programming languages as well as specialty languages intended for specific problems.

Impacts of Computing	**COMPUSCHOLAR ALIGNMENT**
Culture	
9-12S.IC.27 Evaluate computational artifacts with regard to improving their beneficial effects and reducing harmful effects on society.	Our courses contain relevant lessons on the global impact of computing. Students learn about their personal digital footprint and understand that digital identities and online actions have long-term or permanent consequences.

Safety, Law & Ethics	
9-12S.IC.28 Evaluate how computational innovations that have revolutionized aspects of our culture might evolve.	Each course contains lessons on new or evolving aspects of relevant technology as well as impacts of technology on society.
9-12S.IC.29 Evaluate the impact of equity, access, and influence on the distribution of computing resources in a global society.	Our courses contain lessons on the global impact of computing and digital accessibility.
9-12S.IC.30 Debate laws and regulations that impact the development and use of software.	Our courses cover intellectual property laws, copyright considerations and various types of software licensing.