CompuScholar, Inc.

Alignment to the California Computer Science Standards

6th - 8th 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

Middle schools will normally use the "Python Programming" course as a basis for coding concepts and incorporate elements from Digital Savvy and/or Web Design, as desired, to meet additional requirements.

California Computer Science Standards (Middle School)

Computing Systems	COMPUSCHOLAR ALIGNMENT
Devices	
6-8.CS.1 Design modifications to computing devices in order to improve the ways users interact with the devices.	Some courses (e.g. Web Design) contain lessons on user interface design, storyboarding, and improvement of the user experience.
Hardware & Software	
6-8.CS.2 Design a project that combines hardware and software components to collect and exchange data.	N/A (CompuScholar courses avoid requiring hardware components due to the logistical and cost burdens that hardware places on schools).
Troubleshooting	
6-8.CS.3 Systematically apply troubleshooting strategies to identify and resolve hardware and software problems in computing systems.	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	
6-8.NI.4 Model the role of protocols in transmitting data across networks and the Internet.	Our courses describe relevant Internet protocols and network topologies.
Cybersecurity	
6-8.NI.5 Explain potential security threats and security measures to mitigate threats.	Our courses contain relevant lessons on security topics, including physical and electronic threats and mitigation strategies.
6-8.NI.6 Apply multiple methods of information protection to model the secure transmission of information.	Our courses contain lessons on relevant security topics such as encryption (including SSL/TLS) and protection of online personal information.

Data Analysis	COMPUSCHOLAR ALIGNMENT
Storage	
6-8.DA.7 Represent data in multiple ways.	Our courses cover numbering systems such as binary, decimal and hexadecimal. We also discuss encoding of data, including ASCII character and color representations.
Collection, Visualization & Transformation	
6-8.DA.8 Collect data using computational tools and transform the data to make it more useful.	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.
Inference & Models	
6-8.DA.9 Test and analyze the effects of changing variables while using computational models.	Our courses contain lessons that allow students to experiment with input data and parameters to observe changed results.

Algorithms & Programming	COMPUSCHOLAR ALIGNMENT
Algorithms	
6-8.AP.10 Use flowcharts and/or pseudocode to design and illustrate algorithms that solve complex problems.	Our courses describe how to use flowcharts to design algorithms to solve specific problems.
Variables	
6-8.AP.11 Create clearly named variables that store data, and perform operations on their contents.	Our courses carefully describe how to create, initialize, update and use variable data.
Control	
6-8.AP.12 Design and iteratively develop programs that combine control structures and use compound conditions.	Our courses cover traditional flow control structures (conditionals, loops, functions) and the trade-offs in design, including selecting between appropriate flow control logic.

Modularity	
6-8.AP.13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	Our courses cover Object-Oriented Programming (OOP), modular programming with functions, and breaking complex tasks down to manageable logical blocks.
6-8.AP.14 Create procedures with parameters to organize code and make it easier to reuse.	Our courses include opportunities for students to create their own functions/methods and organize related code into objects for reuse.
Program Development	
6-8.AP.15 Seek and incorporate feedback from team members and users to refine a solution that meets user needs.	Our courses contain team projects that include a refinement phase. Students will receive feedback from peers and incorporate that feedback into the final project.
6-8.AP.16 Incorporate existing code, media, and libraries into original programs, and give attribution.	Our courses teach students how to use existing libraries (e.g. Python modules) and leverage those features for new, creative programs.
6-8.AP.17 Systematically test and refine programs using a range of test cases.	Our courses contain team projects that include a testing phase using a written test plan.
6-8.AP.18 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.	Our courses describe team roles and project planning. Team projects incorporate real-world experience with teamwork and basic SDLC concepts.
6-8.AP.19 Document programs in order to make them easier to use, read, test, and debug.	Our courses give students the opportunity to create requirements, design and test documents in addition to well-commented code.

Impacts of Computing	**COMPUSCHOLAR ALIGNMENT**
Culture	
6-8.IC.20 Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.	Our courses contain relevant lessons on the impact of computing on society, personal life and career choices.
6-8.IC.21 Discuss issues of bias and accessibility in the design of existing technologies.	Our courses contain lessons on the global impact of computing, bias and the digital divide. We also describe accessibility for equitable program use.
Social Interactions	
6-8.IC.22 Collaborate with many contributors when creating a computational artifact.	Team projects allow students to work together to create programs or digital artifacts.
Safety, Law & Ethics	
6-8.IC.23 Compare tradeoffs associated with licenses for computational artifacts to balance the protection of the creators' rights and the ability for others to use and modify the artifacts.	Our courses cover intellectual property laws, copyright considerations and various types of software licensing.
6-8.IC.24 Compare tradeoffs between allowing information to be public and keeping information private and secure.	Our courses discuss the security of personal information in online environments.