CompuScholar, Inc.

Support for **CSTA Computer Science** Standards

CSTA Standards:

Name:	Computer Science Teachers Association (2017). CSTA K-12 Computer Science Standards, Revised 2017.
Grade Level:	6 - 8
Standards Link:	http://www.csteachers.org/standards

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

Note 1: The CSTA is not currently conducting crosswalks for specific courses against CSTA standards. Therefore, per CSTA terms of use, CompuScholar does not publish alignments to specific courses, but has broad support for CSTA goals.

CSTA Standards, 6 - 8th Grade

IDENTIFIER	STANDARD	COMPUSCHOLAR STATEMENT OF SUPPORT
2-CS-01	Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices.	Our courses contain discussion of relevant design and usability features.
2-CS-02	Design projects that combine hardware and software components to collect and exchange data.	Our courses rely on locally provided hardware if needed to produce data for student projects.
2-CS-03	Systematically identify and fix problems with computing devices and their components.	Our courses contain relevant discussions of troubleshooting and debugging techniques.
2-NI-04	Model the role of protocols in transmitting data across networks and the Internet.	Our courses contain discussions of relevant network protocols.
2-NI-05	Explain how physical and digital security measures protect electronic information.	Our courses contain discussion of relevant security features for hardware and software.
2-NI-06	Apply multiple methods of encryption to model the secure transmission of information.	Our courses contain discussions of relevant encryption and data transmission technologies.

2-DA-07	Represent data using multiple encoding schemes.	Our courses contain relevant discussions of data encoding and representation.
2-DA-08	Collect data using computational tools and transform the data to make it more useful and reliable.	Our courses discussion how to obtain or generate data for analysis, where relevant.
2-DA-09	Refine computational models based on the data they have generated.	Our courses contain discussion on simulations and step-wise refinement, where relevant.
2-AP-10	Use flowcharts and/or pseudocode to address complex problems as algorithms.	Our courses describe flowcharts as an aid to algorithm development and program coding.
2-AP-11	Create clearly named variables that represent different data types and perform operations on their values.	Our courses describe best practices for variable naming conventions and data type selection.
2-AP-12	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	Our courses contain many opportunities to code a variety of control structures and conditional logic.
2-AP-13	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	Our courses describe how to subdivide problems into smaller tasks, encapsulate data and concepts, etc.
2-AP-14	Create procedures with parameters to organize code and make it easier to reuse.	Our courses teach students how to create functions with parameters and return data.
2-AP-15	Seek and incorporate feedback from team members and users to refine a solution that meets user needs.	Our courses have team projects with opportunities for peer review, feedback and refinement.
2-AP-16	Incorporate existing code, media, and libraries into original programs, and give attribution.	Our courses make use of relevant libraries and SDKs to complete programs.
2-AP-17	Systematically test and refine programs using a range of test cases.	Our courses contain specific lessons on debugging and program testing.
2-AP-18	Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.	Our courses contain team projects with a SDLC, timeline and collaborating team roles.
2-AP-19	Document programs in order to make them easier to follow, test, and debug.	Our courses contain relevant discussion of requirements, design docs and test plans.
2-IC-20	Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.	Our courses contain lessons on the impact of computing in today's society.
2-IC-21	Discuss issues of bias and accessibility in the design of existing technologies.	Our courses contain lessons on equitable computing access, where relevant.

2-IC-22	Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.	Our courses offer opportunities for students to gather data as part of team projects using any desired approach.
2-IC-23	Describe tradeoffs between allowing information to be public and keeping information private and secure.	Our courses contain discussions of online safety, digital footprints, and information security practices.