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:	Tech Essentials, ISBN 978-1-946113-03-0
	Course Description and Syllabus
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.

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.

CSTA Standards, 6 - 8th Grade

	Apply multiple methods of encryption to model	Our courses contain discussions of
2-NI-06	the secure transmission of information.	relevant encryption and data
		transmission technologies.
	Represent data using multiple encoding schemes.	Our courses contain relevant
2-DA-07		discussions of data encoding and
		representation.
	Collect data using computational tools and	Our courses discussion how to obtain or
2-DA-08	transform the data to make it more useful and	generate data for analysis, where
	reliable.	relevant.
	Refine computational models based on the data	Our courses contain discussion on
2-DA-09	they have generated.	simulations and step-wise refinement,
		where relevant.
	Use flowcharts and/or pseudocode to address	Our courses describe flowcharts as an
2-AP-10	complex problems as algorithms.	aid to algorithm development and
		program coding.
	Create clearly named variables that represent	Our courses describe best practices for
2-AP-11	different data types and perform operations on	variable naming conventions and data
	their values.	type selection.
	Design and iteratively develop programs that	Our courses contain many opportunities
2-AP-12	combine control structures, including nested	to code a variety of control structures
	loops and compound conditionals.	and conditional logic.
	Decompose problems and subproblems into	Our courses describe how to subdivide
2-AP-13	parts to facilitate the design, implementation,	problems into smaller tasks,
	and review of programs.	encapsulate data and concepts, etc.
	Create procedures with parameters to organize	Our courses teach students how to
2-AP-14	code and make it easier to reuse.	create functions with parameters and
		return data.
	Seek and incorporate feedback from team	Our courses have team projects with
2-AP-15	members and users to refine a solution that	opportunities for peer review, feedback
	meets user needs.	and refinement.
	Incorporate existing code, media, and libraries	Our courses make use of relevant
2-AP-16	into original programs, and give attribution.	libraries and SDKs to complete
		programs.
	Systematically test and refine programs using a	Our courses contain specific lessons on
2-AP-17	range of test cases.	debugging and program testing.
	Distribute tasks and maintain a project timeline	Our courses contain team projects with
2-AP-18	when collaboratively developing computational	a SDLC, timeline and collaborating team
	artifacts.	roles.
	Document programs in order to make them	Our courses contain relevant discussion
2-AP-19	easier to follow, test, and debug.	of requirements, design docs and test
		plans.
	Compare tradeoffs associated with computing	Our courses contain lessons on the
2-IC-20	technologies that affect people's everyday	impact of computing in today's society.
	activities and career options.	

2-IC-21	Discuss issues of bias and accessibility in the	Our courses contain lessons on
	design of existing technologies.	equitable computing access, where
		relevant.
2-IC-22	Collaborate with many contributors through	Our courses offer opportunities for
	strategies such as crowdsourcing or surveys	students to gather data as part of team
	when creating a computational artifact.	projects using any desired approach.
2-IC-23	Describe tradeoffs between allowing information	Our courses contain discussions of
	to be public and keeping information private and	online safety, digital footprints, and
	secure.	information security practices.