CompuScholar, Inc.

Alignment to the K-12 Computer Science Framework

6th - 8th Grade Band

K-12 Computer Science Framework Details:

Standards Link:	K–12-Computer-Science-Framework.pdf

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

Note 1: Citation(s) listed may represent a subset of the instances where objectives are met throughout the course.

Note 2: Citation(s) for a "Lesson" refer to the "Lesson Text" elements and associated "Activities" within the course, unless otherwise noted. The "Instructional Video" components are supplements designed to introduce or re-enforce the main lesson concepts, and the Lesson Text contains full details.

K-12 CS Concepts, 6 - 8 Grade Band

K-12 CS SHORTHAND	K-12 CS STATEMENT	COMPUSCHOLAR ALIGNMENT
6–8.Computing Systems.Devices	The interaction between humans and computing devices presents advantages, disadvantages, and unintended consequences. The study of human–computer interaction can improve the design of devices and extend the abilities of humans.	Our courses contain introductory chapters on computer hardware and relevant software. Additional chapters cover optimal relevant design, digital accessibility, global computing issues, impacts of artificial intelligence, and technology-specific pros, cons, risks and rewards.
6–8.Computing Systems.Hardware and Software	Hardware and software determine a computing system's capability to store and process information. The design or selection of a computing system involves multiple considerations and potential tradeoffs, such as functionality, cost, size, speed, accessibility, and aesthetics.	Our courses contain introductory chapters on computer hardware and operating system considerations. Additional chapters or lessons discuss trade-offs in purchasing decisions for computing systems, web hosting providers, or other relevant technology.

K-12 CS SHORTHAND	K-12 CS STATEMENT	COMPUSCHOLAR ALIGNMENT
6–8.Computing Systems. Troubleshooting	Comprehensive troubleshooting requires knowledge of how computing devices and components work and interact. A systematic process will identify the source of a problem, whether within a device or in a larger system of connected devices.	Our courses contain dedicated troubleshooting and debugging information for relevant technology. The courses combine to cover diagnosis of common computing problems, network connectivity, code issues, and specific debugging tools (e.g. breakpoints) where appropriate.
6–8.Networks and the Internet.Network Communication and Organization	Computers send and receive information based on a set of rules called protocols. Protocols define how messages between computers are structured and sent. Considerations of security, speed, and reliability are used to determine the best path to send and receive data.	Our courses contain introductory networking concepts such as network hardware components, topology, relevant protocols, addressing, and design trade-offs where appropriate.
6–8.Networks and the Internet.Cybersecurity	The information sent and received across networks can be protected from unauthorized access and modification in a variety of ways, such as encryption to maintain its confidentiality and restricted access to maintain its integrity. Security measures to safeguard online information proactively address the threat of breaches to personal and private data.	Our courses contain chapters or lessons on relevant security topics such as encryption (including SSL/TLS), online safety and privacy, protecting personal information online, and establishing defenses against mal-ware and viruses.
6–8.Data and Analysis.Collection	People design algorithms and tools to automate the collection of data by computers. When data collection is automated, data is sampled and converted into a form that a computer can process. For example, data from an analog sensor must be converted into a digital form. The method used to automate data collection is influenced by the availability of tools and the intended use of the data.	Our courses contain lessons on algorithms at a design (e.g. flowchart) and practical implementation level, where relevant. We additionally cover the encoding of data, numeric representation and data types on computers, and explore conversions between numbering systems (e.g. decimal, binary, hex).

K-12 CS SHORTHAND	K-12 CS STATEMENT	COMPUSCHOLAR ALIGNMENT
6–8.Data and Analysis.Storage	Applications store data as a representation. Representations occur at multiple levels, from the arrangement of information into organized formats (such as tables in software) to the physical storage of bits. The software tools used to access information translate the low-level representation of bits into a form understandable by people.	Our courses cover the encoding of data, numeric representation and data types on computers, and explore conversions between numbering systems (e.g. decimal, binary, hex). Topic-specific examples are given (e.g. RGB colors in hex, floating point data types and data structures) where appropriate.
6–8.Data and Analysis.Visualization and Transformation	Data can be transformed to remove errors, highlight or expose relationships, and/or make it easier for computers to process.	Our courses describe the relationships between data structures and human concepts in a technology-specific manner. For example, we discuss how new HTML5 elements bring additional semantic meaning to web pages or how relational data fits into tables. Relevant lessons describe how computers process input data to produce the desired results.
6–8.Data and Analysis.Inference and Models	Computer models can be used to simulate events, examine theories and inferences, or make predictions with either few or millions of data points. Computer models are abstractions that represent phenomena and use data and algorithms to emphasize key features and relationships within a system. As more data is automatically collected, models can be refined.	Our courses describe how to create relevant algorithms, model and predict real-world systems and generate output data given specific input parameters.

K-12 CS Practices (All Grades)

K-12 CS SHORTHAND	K-12 CS STATEMENT	COMPUSCHOLAR ALIGNMENT
P1.Fostering an Inclusive Computing Culture.1	Include the unique perspectives of others and reflect on one's own perspectives when designing and developing computational products.	Our courses contain lessons on the global impact of computing and digital accessibility. Team projects allow students to collaborate and contribute unique viewpoints to create digital artifacts.
P1.Fostering an Inclusive Computing Culture.2	Address the needs of diverse end users during the design process to produce artifacts with broad accessibility and usability	Our courses contain lessons on digital accessibility and relevant standards. Individual and group projects, where relevant, are required to follow accessibility guidelines.

K-12 CS SHORTHAND	K-12 CS STATEMENT	COMPUSCHOLAR ALIGNMENT
P1.Fostering an Inclusive Computing Culture.3	Employ self- and peer-advocacy to address bias in interactions, product design, and development methods.	The courses contain guidance on ethical computing, professional behavior and impacts of a global digital environment. Multiple opportunities are available for students to work in teams and provide feedback within the team and to other teams.
P2.Collaborating Around Computing.1	Cultivate working relationships with individuals possessing diverse perspectives, skills, and personalities.	Our courses contain team projects and classroom discussion opportunities that allow diverse individuals to work together to debate, form opinions and produce digital artifacts.
P2.Collaborating Around Computing.2	Create team norms, expectations, and equitable workloads to increase efficiency and effectiveness.	Team projects and related lessons instruct students on creating an appropriate scope, timeline, and executing projects in phases. Team members will work efficiently together or in parallel to meet self-directed project milestones.
P2.Collaborating Around Computing.3	Solicit and incorporate feedback from, and provide constructive feedback to, team members and other stakeholders.	Team projects contain testing, evaluation and feedback phases. Peer and professional or teacher review is used to generate actionable suggestions for improvement of digital artifacts.
P2.Collaborating Around Computing.4	Evaluate and select technological tools that can be used to collaborate on a project.	Students are exposed to a variety of productivity tools and will use collaborative services and technology as appropriate during team projects.
P3.Recognizing and Defining Computational Problems.1	Identify complex, interdisciplinary, real-world problems that can be solved computationally.	Courses, where appropriate, contain lessons on modeling real-world problems in a variety of subject areas (e.g. physics, space exploration, monitoring and interpretation of website behavior).
P3.Recognizing and Defining Computational Problems.2	Decompose complex real-world problems into manageable subproblems that could integrate existing solutions or procedures	Courses, where appropriate, contain lessons on algorithms, decomposition of tasks into smaller functions, and the application of artificial intelligence to complement and extend human functionality.
P3.Recognizing and Defining Computational Problems.3	Evaluate whether it is appropriate and feasible to solve a problem computationally.	Courses, where appropriate, contain lessons on the limitations and implementation challenges of algorithms. This includes discussion of whether parts of a complex problem are best handled by an algorithm or AI instead of a human.

K-12 CS SHORTHAND	K-12 CS STATEMENT	COMPUSCHOLAR ALIGNMENT
P4.Developing and Using Abstractions.1	Extract common features from a set of interrelated processes or complex phenomena.	Courses, where appropriate, evaluate situations to identify algorithmic applications, functions, and repeatable patterns to represent part or all of a task.
P4.Developing and Using Abstractions.2	Evaluate existing technological functionalities and incorporate them into new designs.	In team projects, students will research selected topics and select appropriate technical tools (e.g. word processors, spreadsheets, databases, websites, apps) to visualize and present information to new audiences.
P4.Developing and Using Abstractions.3	Create modules and develop points of interaction that can apply to multiple situations and reduce complexity.	Courses, where appropriate, introduce functions and decomposition of complex problems into more manageable tasks.
P4.Developing and Using Abstractions.4	Model phenomena and processes and simulate systems to understand and evaluate potential outcomes.	Courses, where appropriate, introduce modeling of physical and other phenomena and simulations to predict outcomes across a variety of subject areas.
P5.Creating Computational Artifacts.1	Plan the development of a computational artifact using an iterative process that includes reflection on and modification of the plan, taking into account key features, time and resource constraints, and user expectations.	Students will learn about project management tools such as project plans, requirements & design documents, development lifecycles (e.g. classic waterfall), and apply those tools and skills in group or individual projects.
P5.Creating Computational Artifacts.2	Create a computational artifact for practical intent, personal expression, or to address a societal issue.	Courses include team or individual projects where students will select a topic or issue of interest, research and gather data, and create one or more digital artifacts to summarize or interpret the results.
P5.Creating Computational Artifacts.3	Modify an existing artifact to improve or customize it.	Students are frequently given "starter" projects as the basis for completing more complex activities. They are also given opportunities to examine output from other teams, provide feedback, and incorporate feedback to make improvements.
P6.Testing and Refining Computational Artifacts.1	Systematically test computational artifacts by considering all scenarios and using test cases.	Courses contain dedicated chapters and lessons on debugging, test plans and troubleshooting skills. Team projects include iterative testing and fixing phases to address bug reports and feedback.

K-12 CS SHORTHAND	K-12 CS STATEMENT	COMPUSCHOLAR ALIGNMENT
P6.Testing and Refining Computational Artifacts.2	Identify and fix errors using a systematic process.	Courses contain dedicated chapters and lessons that describe appropriate procedures for finding and fixing problems or bugs. Test plans and checklists are introduced as useful tools to systematically evaluate a digital artifact.
P6.Testing and Refining Computational Artifacts.3	Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability, and accessibility.	Team projects include iterative test and feedback phases. Students will incorporate test reports and feedback from peers and professionals to improve their digital artifacts.
P7.Communicating About Computing.1	Select, organize, and interpret large data sets from multiple sources to support a claim.	Team projects ask students to research a topic of interest, obtain data, incorporate that data into an appropriate digital expression, and interpret the results for a wider audience.
P7.Communicating About Computing.2	Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose.	The courses describe how to create requirements and design documents, storyboards, flowcharts, end user documents with appropriate languages as
P7.Communicating About Computing.3	Articulate ideas responsibly by observing intellectual property rights and giving appropriate attribution.	The courses contain dedicated lessons on computing ethics, intellectual property, copyrights, and proper citation. Students are required to adhere to these standards throughout each course.