

CompuScholar, Inc.

Alignment to the California CTE "ICT" Software and Systems Development Pathway

California Standards Information:

CTE Page	California CTE Standards Page
Standards Link:	CTE ICT Curriculum Standards

CompuScholar Courses:

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

California's CTE "Information and Communication Technologies (ICT)" program defines a "Software and Systems Development Pathway" for students. Any of CompuScholar's programming courses can generally be used as a primary resource to meet these standards. Schools may also wish to add "Digital Savvy" and "Web Design" material to address specific requirements.

CTE / ICT Software and System Development Pathway Standards

C1.0 Identify and apply the systems development process.	COMPUSCHOLAR ALIGNMENTS
C1.1 Identify the phases of the systems development life cycle, including analysis, design, programming, testing, implementation, maintenance, and	Each programming course contains lessons on common SDLC approaches and opportunities for students to participate in full SDLC projects.
C1.2 Identify and describe models of systems development, systems development life cycle (SDLC), and agile computing.	Each programming course contains lessons on common SDLC approaches and opportunities for students to participate in full SDLC projects.
C1.3 Identify and describe how specifications and requirements are developed for new and existing software applications.	Each programming course describes relevant requirements and design documents and provides opportunities for students to create those

C1.4 Work as a member of, and within the scope and boundaries of, a development project team.	Each course contains team projects allowing students to work within a group.
C1.5 Track development project milestones using the concept of versions.	Each course contains lessons on project management and planning, plus opportunities to create and execute project plans.
C1.6 Diagram processes using flowcharts and the Unified Modeling Language.	Each course describes flowcharting for algorithm development.

C2.0 Define and analyze systems and software requirements.	COMPUSCHOLAR ALIGNMENTS
C2.1 Describe the major purposes and benefits of development, including automation, improving productivity, modeling and analysis, and entertainment.	Each course contains a variety of real-world examples that can be represented or solved by program development.
C2.2 Recognize and prevent unintended consequences of development work: programming errors, security issues, health and environmental risks, and privacy concerns.	Each course contains relevant lessons on testing, security, ethical and privacy considerations.
C2.3 Develop strategies that target the specific needs and desires of the customer.	Each course contains opportunities to create requirements documents based on customer needs.
C2.4 Analyze customers' needs for development.	Each course contains opportunities to create design documents based on customer requirements.
C2.5 Determine and document the requirements and alternative solutions to fulfill the customers' needs.	Each course contains opportunities to create requirements and design documents for creative projects.

C3.0 Create effective interfaces between humans and technology.	COMPUSCHOLAR ALIGNMENTS
C3.1 Describe and apply the basic process of input, processing, and output.	Each course covers user input, console and/or graphical output, and internal processing.
C3.2 Design effective and intuitive interfaces using knowledge of cognitive, physical, and social interactions.	Each course covers relevant console or GUI design considerations.
C3.3 Support methods of accessibility for all potential users, including users with disabilities and non-English-speaking users.	Courses cover accessibility standards, where relevant.

C4.0 Develop software using programming languages.	COMPUSCHOLAR ALIGNMENTS
C4.1 Identify and describe the abstraction level of programming languages from low-level, hardware-based languages to high-level, interpreted, Web-based languages.	Each course describes a variety of programming languages, their implementation models and intended uses.
C4.2 Describe the interaction and integration of programming languages and protocols such as how client-side programming can work with server-side programming to use a query language to access a	Courses contain descriptions of protocols and system interactions, where relevant.

C4.3 Identify and use different authoring tools and integrated development environments (IDEs).	Each course will use or describe an IDE relevant to the specific programming language.
C4.4 Identify and apply data types and encoding.	Each course covers data types and encoding concepts.
C4.5 Demonstrate awareness of various programming paradigms, including procedural, object oriented, event-driven, and multithreaded programming.	Each course teaches basic program flow, procedural and object-oriented programming.
C4.6 Use proper programming language syntax.	Each course teaches relevant language syntax and best coding practices.
C4.7 Use various data structures, arrays, objects, files, and databases.	Courses will teach and use relevant data structures, including arrays, lists, files and objects.
C4.8 Use object oriented programming concepts, properties, methods, and inheritance.	Each course teaches OOP concepts and allows students to create their own classes.
C4.9 Create programs using control structures, procedures, functions, parameters, variables, error recovery, and recursion.	Students will create many programs in each course, including these concepts as they are introduced.
C4.10 Create and know the comparative advantages of various queue, sorting, and searching algorithms.	Courses will discuss sorting and searching algorithms and relative performances.
C4.11 Document development work for various audiences, such as comments for other programmers, and manuals for users.	Courses describe both basic commenting practices and online reference materials for the specific language.

C5.0 Test, debug, and improve software development work.	COMPUSCHOLAR ALIGNMENTS
C5.1 Identify the characteristics of reliable, effective, and efficient products.	Each course contains a dedicated
C5.2 Describe the ways in which specification changes and technological advances can require the modification of programs.	Each course contains relevant lessons on the full SDLC, including maintenance of existing programs.
C5.3 Use strategies to optimize code for improved performance.	Courses contain analysis of code and choice of algorithms to optimize performance, where relevant.
C5.4 Test software and projects.	Our courses contain dedicated troubleshooting and debugging chapters and multiple opportunities to iteratively test software.
C5.5 Evaluate results against initial requirements.	Our courses contain opportunities to define and execute a test plan based on project requirements.
C5.6 Debug software as part of the quality assurance process.	Our courses contain dedicated troubleshooting and debugging strategies, including code analysis, tracing (logging) and debuggers.

C6.0 Integrate a variety of media into development projects.	COMPUSCHOLAR ALIGNMENTS
C6.1 Identify the basic design elements necessary to produce effective print, video, audio, and interactive media.	The Web Design course contains design strategies and incorporation of multi-media elements into effective user interfaces.

C6.2 Describe the various encoding methods of media and trade-offs: vector graphics vs. bitmaps, and bit depth.	The Web Design course describes the different image representations and optimization approaches.
C6.3 Use media design and editing software: keyframe animation, drawing software, image editors, and three-dimensional design.	The Digital Savvy course contains a chapter on image editing.
C6.4 Develop a presentation or other multimedia project: video, game, or interactive Web sites, from storyboard to production.	All courses contain opportunities for creative projects where students can apply relevant skills to create digital artifacts.
C6.5 Analyze the use of media to determine the appropriate file format and level of compression.	The Web Design course describes file formats for images, sounds and videos.
C6.6 Integrate media into a full project using appropriate tools.	All courses contain opportunities for creative projects where students can apply relevant skills to create digital artifacts.
C6.7 Create and/or capture professional-quality media, images, documents, audio, and video clips.	All courses contain opportunities for creative projects where students can insert their own data or media, when desired.

C7.0 Develop Web and online projects.	COMPUSCHOLAR ALIGNMENTS
C7.1 Identify the hardware (server) and software required for Web hosting and other services.	The Web Design course describes web servers and hosting requirements.
C7.2 Describe the full process of online content delivery, registering domain names, setting up hosting, and setting up e-mail addresses.	The Web Design course describes domain name registration and hosting selection.
C7.3 Attract Web-site visitors through search engine optimization using various strategies like keywords and meta-tags.	The Web Design course discusses meta-data within pages and introduces SEO concepts.
C7.4 Enable e-commerce capabilities to sell products, create a shopping cart, and handle credit card transactions.	Described but not implemented in Web Design.
C7.5 Create an online project, Web-based business, and e-portfolio.	The Web Design course offers multiple opportunities to create local websites.
C7.6 Optimize fast delivery and retrieval of online content such as Web pages.	The Web Design course recommends best practices for optimized content, where relevant.

C8.0 Develop databases.	COMPUSCHOLAR ALIGNMENTS
C8.1 Describe the critical function of databases in modern organizations.	The Digital Savvy course contains an introductory database chapter.
C8.2 Identify and use the basic structures of databases, fields, records, tables, and views.	Students will learn about database, fields, records and tables in Digital Savvy.

C8.3 Identify and explain the types of relationships between tables (one-to-one, one-to-many, many-to-many) and use methods to establish these relationships, including primary keys, foreign keys, and indexes.	Students will learn about basic SQL relationships in Digital Savvy.
C8.4 Use data modeling techniques to create databases based upon business needs.	N/A
C8.5 Use queries to extract and manipulate data (select queries, action queries).	Students will learn basic SQL queries in Digital Savvy.
C8.6 Develop databases that are properly normalized using appropriate schemas.	N/A
C8.7 Export and import data to and from other applications and a database recognizing the limitations and challenges inherent in the process.	Digital Savvy students will have opportunities to mix database content with other digital artifacts in creative projects.
C8.8 Analyze and display data to assist with decision making using methods like cross tabulations, graphs, and charts.	Digital Savvy students will have opportunities to incorporate data from databases in spreadsheets, charts and presentation programs.

C9.0 Develop software for a variety of devices, including robotics.	COMPUSCHOLAR ALIGNMENTS
C9.1 Demonstrate awareness of the applications of device development work, including personalized computing, robotics, and smart appliances.	N/A (CompuScholar courses avoid requiring hardware components due to the logistical and cost burdens that hardware places on schools).
C9.2 Install equipment, assemble hardware, and perform tests using appropriate tools and technology.	
C9.3 Use hardware to gain input, process information, and take action.	
C9.4 Apply the concepts of embedded programming, including digital logic, machine-level representation of data, and memory-system organization.	
C9.5 Program a micro-controller for a device or robot.	

C10.0 Develop intelligent computing.	COMPUSCHOLAR ALIGNMENTS
C10.1 Describe models of intelligent behavior and what distinguishes humans from machines.	Each course contains a discussion of relevant artificial intelligence concepts.
C10.2 Describe the major areas of intelligent computing, including perception, proximity, processing, and control.	Each course contains a discussion of example artificial intelligence applications.
C10.3 Know artificial intelligence methods such as neural networks, Bayesian inferences, fuzzy logic, and finite state machines.	N/A
C10.4 Implement artificial intelligent behavior through various methods: mathematical modeling, reinforcement learning, and probabilistic analysis.	N/A