CompuScholar, Inc.

Alignment to the 2023 Oklahoma Computer Science Standards

9th - 12th Grade Band

Oklahoma Standards Information:

CS Page	Oklahoma Computer Science Standards
Standards Link:	2023 Oklahoma Academic Standards for Computer Science

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
Course Title:	**Computer Science Foundations**, ISBN 978-1-946113-02-3 Course Description and Syllabus
Course Title:	**Java Programming**, ISBN 978-1-946113-99-3 Course Description and Syllabus
Course Title:	**C# Programming**, ISBN 978-1-946113-01-6 Course Description and Syllabus
Course Title:	**Unity Game Programming**, ISBN 978-0-9887070-7-8 Course Description and Syllabus

Oklahoma Academic Standards for Computer Science (High School)

Computing Systems	**COMPUSCHOLAR ALIGNMENT**
Devices	
L1.CS.D.01 Model how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.	Our courses contain lessons that describe how various hardware components (e.g. CPU, RAM, disk drives) encapsulate computing tasks such as processing, short-term storage and long-term
Hardware & Software	
L1.CS.HS.01 Analyze the levels of abstraction and interactions between application software, system software, and hardware.	Our courses describe the relationships between hardware, operating systems, device drivers, and a variety of end-user applications.
L2.CS.HS.01 Identify and categorize the roles of a variety of operating system software.	Our courses discuss relevant operating system features and tasks including basic file management.

Troubleshooting	
L1.CS.T.01 Develop and apply criteria for the systematic discovery of errors and systematic strategies for the correction of errors 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.
L2.CS.T.01 Illustrate how understanding the ways hardware components facilitate logic, input, output, and storage in computing systems will support troubleshooting.	Our courses contain lessons that describe hardware components including storage, processing and I/O peripherals.

Network & The Internet	**COMPUSCHOLAR ALIGNMENT**
Network Communication & Organization	
L1.NI.NCO.01 Evaluate the scalability and reliability of networks by identifying and illustrating the basic components of computer networks (e.g., routers, switches, servers, etc.) and network protocols (e.g., IP, DNS).	Our courses describe networking components and common network topology. IP addresses, MAC addresses and URLs are introduced for identification of devices and online resources.
L2.NI.NCO.01 Describe the issues that impact network functionality (e.g., bandwidth, load, latency, topology).	Network design considerations, including scalability, bandwidth, security and points of failure are addressed where relevant.
Cybersecurity	
L1.NI.CY.01 Compare physical and cybersecurity measures by evaluating trade-offs between the usability and security of a computing system and the risks of an attack.	Our courses contain chapters or lessons on security topics, including physical vs. electronic security mechanisms where relevant.
L2.NI.CY.01 Compare and refine ways in which software developers protect devices and information from unauthorized access.	Our courses contain chapters or lessons on relevant security topics such as encryption (including SSL/TLS) and establishing defenses against mal-ware and viruses.
L1.NI.CY.02 Recommend security measures to address various scenarios based on information security principles.	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.
L1.NI.CY.03 Explain trade-offs when selecting and implementing cybersecurity recommendations from multiple perspectives, such as the user, enterprise, and government.	N/A

Data Analysis	COMPUSCHOLAR ALIGNMENT
Storage	
L1.IC.SLE.03 Evaluate the social and economic consequences of how law and ethics interact with digital aspects of privacy, data, property, information, and identity.	Our courses cover numbering systems such as binary, decimal and hexadecimal. The encoding of data, including ASCII character and color representations is discussed where relevant.
L1.DA.S.02 Evaluate the trade-offs in how data is organized and stored digitally.	Our courses cover representation of data in multiple formats, from simple encoding of data types up through data structures and object definitions with properties. Choices of data type, data structure, or object definitions have pros and cons and may impact the desirable characteristics of the program or the maintainability of the underlying code.
Collection, Visualization & Transformation	
L1.DA.CVT.01 Use tools and techniques to locate, collect, and create visualizations of small and largescale data sets (e.g., paper surveys and online data sets).	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.
L2.DA.CVT.01 Use data analysis tools and techniques to identify patterns from complex real-world data.	Our team projects and other labs give students opportunities to visualize and explain data by spreadsheet, charts & graphs or programmatic display, where relevant.
L2.DA.CVT.02 Generate data sets that use a variety of data collection tools and analysis techniques to support a claim and/or communicate information.	Our team projects and other labs give students opportunities to research topics, obtain data sets and produce digital artifacts to explain or influence opinion about an issue.
Inference & Models	
L1.DA.IM.01 Illustrate and explain the relationships between collected data elements using computational models.	N/A
L2.DA.IM.01 Use models and simulations to help plan, conduct, and refine investigations.	Our courses contain opportunities for student to build models of real-world phenomena and simulate or predict results. Students are encouraged to alter or modify input data to observe the impact on resulting output and verify initial assumptions.

Algorithms & Programming	COMPUSCHOLAR ALIGNMENT
Algorithms	
L1.AP.A.01 Create a prototype that uses algorithms (e. g., searching, sorting, finding shortest distance) to provide a possible solution for a real- world problem.	Our courses describe how to use flowcharts to design algorithms to solve specific problems. Common sorting and searching algorithms or game-specific AI routines are illustrated.

L2.AP.A.01 Model and use appropriate terminology to describe how artificial intelligence algorithms drive many software and physical systems (e.g., autonomous robots, pattern recognition, text analysis).	Our courses contain context-appropriate descriptions of AI algorithms such as game AI and applications to real-world problems (e.g. self-driving cars).
L2.AP.A.02 Develop an artificial intelligence algorithm to play a game against a human opponent or solve a real-world problem.	Our courses contain context-appropriate opportunities for students to study or implement AI algorithms to perform specific tasks (e.g. game AI or other real-world problems).
L2.AP.A.03 Critically examine and trace classic algorithms (e.g., selection sort, insertion sort, binary search, linear search).	Our courses contain appropriate analysis and comparison of algorithms, including multiple sorting and searching approaches.
L2.AP.A.04 Evaluate algorithms (e.g., sorting, searching) in terms of their efficiency and clarity.	Our courses contain appropriate analysis and comparison of algorithms, including trade-offs in performance, coding complexity, and accuracy of results.
Variables	
L1.AP.V.01 Demonstrate the use of lists (e.g., arrays) to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.	Our programming courses cover simple data structures such as arrays and lists. Students will use each data structure in hands-on projects.
L2.AP.V.01 Compare and contrast data structures and their uses (e.g., lists, stacks, queues).	Our programming courses cover lists, stacks and queues and the trade-offs involved with each representation. Students will use each data structure in hands-on projects.
Control	
L1.AP.C.01 Justify the selection of specific control structures (e.g., sequence, conditionals, repetition, procedures) considering program efficiencies such as readability, performance, and memory usage.	Our courses cover traditional flow control structures (conditionals, loops, functions) and the trade-offs in design, including selecting between appropriate flow control logic.
L2.AP.C.01 Model the execution of repetition (e.g., loops, recursion) of an algorithm illustrating output and changes in values of named variables.	Our courses include lessons on iteration using all traditional looping structures, plus recursion. Careful study is made of loop or recursive method behavior and resulting changes in variables. Students are introduced to debugging techniques such as breakpoints and stepping through live code at run-time.
Modularity	
L1.AP.M.01 Decompose problems into procedures using systematic analysis and design.	Our courses cover Object-Oriented Programming (OOP), modular programming with functions, and functional decomposition of complex tasks down to manageable logical blocks.
L2.AP.M.01 Construct solutions to problems using student-created components (e.g., procedures, modules, objects).	Our courses include opportunities for students to build their own functions/methods and objects.

L1.AP.M.02 Create computational artifacts by systematically organizing, manipulating and/or processing data.	Team projects and other labs give students the opportunity to incorporate real-world data in the creation of their own digital artifacts and programs.
L2.AP.M.02 Design or redesign a solution to a large-scale computational problem by identifying generalizable patterns.	Our courses compare and contrast standard algorithms for large scale sorting and searching. They additionally cover relevant AI algorithms to implement specific tasks or features.
L2.AP.M.03 Create programming solutions by reusing existing code (e.g., libraries, Application Programming Interface (APIs), code repositories).	Our courses explain and use relevant, language-specific libraries such as the Java class library, the .NET framework or Python modules.
Program Development	
L1.AP.PD.01 Create software that will provide solutions to a variety of users using a software development process.	Our courses contain team projects that allow students to define, design, build and test a unique project using standard SDLC stages and traditional requirements, design and test documentation. Accessibility concerns are addressed as needed.
L2.AP.PD.01 Create software that will provide solutions to a variety of users using multiple software development processes.	Our courses contain team projects in which students will experience standard software engineering roles and development life-cycles. Students will write project plans, requirements and design documents, and follow systematic test plans.
L1.AP.PD.02 Evaluate a variety of software licensing schemes (e.g., open source, freeware, commercial) and discuss the advantages and disadvantages of each scheme in software development.	Our courses contain lessons describing the types of software licensing, pros and cons of each and intellectual property considerations.
L2.AP.PD.02 Design software in a project team environment using integrated development environments (IDEs), versioning systems, and collaboration systems.	Our courses contain team projects that provide opportunities for collaboration on unique projects. Students will use a language-specific IDE to complete the coding.
L1.AP.PD.03 While working in a team, develop, test, and refine event-based programs that solve practical problems or allow self-expression.	Our courses contain team projects that allow students to define, design, build and test a unique project using standard SDLC stages. The resulting programs can GUI/ event-based as appropriate.
L2.AP.PD.03 Develop programs for multiple computing platforms.	Our courses teach multiple languages, and those with easy cross-platform support (e.g. Java or HTML or Unity) are clearly defined as such, so student projects can be run on computers with different operating systems.
L1.AP.PD.04 Using visual aids and documentation, illustrate the design elements and data flow (e.g., flowcharts, pseudocode) of the development of a complex program.	Our courses teach flowcharts as a way to help design and visualize the detailed steps in an algorithm.

L2.AP.PD.04 Systematically examine code for correctness, usability, readability, efficiency, portability, and scalability through peer review.	Our courses contain team projects that include a testing phase using a written test plan. Students will receive feedback from peers and incorporate that feedback into the final project.
L1.AP.PD.05 Evaluate and refine computational artifacts to make them more user-friendly, efficient and/or accessible.	Our courses describe multiple algorithms (e.g. sorting or searching or AI) and the assorted design & efficiency considerations that go into algorithm selection and development.
L2.AP.PD.05 Develop and use a series of test cases to verify that a program performs according to its design specifications.	Our courses contain team projects that include a testing phase using a written test plan. Students will receive feedback from peers and incorporate that feedback into the final project.
L2.AP.PD.06 Explain security issues that might lead to compromised computer programs.	Our courses contain chapters or lessons on relevant security topics such as SSL/TLS, mal-ware and viruses, and protection of business data.
L2.AP.PD.07 Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality).	Our courses contain multiple projects where starting code is provided and students will expand or complete the initial project. Regression testing is described in chapters of debugging and testing.

Impacts of Computing	**COMPUSCHOLAR ALIGNMENT**
Culture	
L1.IC.CU.01 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.	Our courses contain relevant lessons on the global impact of computing, ethical computing concepts, intellectual property and licensing, careers in computing, etc.
L2.IC.CU.01 Evaluate the beneficial and harmful effects that computational artifacts and innovations have on society.	Our courses contain relevant lessons on the global impact of computing. Students learn about their personal digital footprint and understand that digital identities and online actions have long-term or permanent consequences.
L1.IC.CU.02 Test and refine computational artifacts to ensure access to a variety of user audiences.	Students are presented with examples and projects that are free from bias and inequity, and are similarly encouraged to follow ethical practices in their own development.
L2.IC.CU.02 Evaluate the impact of location and user audience on the distribution of computing resources in a global society.	Our courses contain lessons on the global impact of computing and digital accessibility.
L1.IC.CU.03 Demonstrate ways a given algorithm can help solve computational problems across disciplines.	Our courses cover a variety of algorithms that can be applied to real-world situations in mathematics, physics, economics, etc.

L2.IC.CU.03 Design and implement a study that evaluates or predicts how creating, testing, and refining computational artifacts has revolutionized an aspect of our culture and how it might evolve (e.g., education, healthcare, art/entertainment, energy).	Our courses include lessons on trends in future development, as well as opportunities to study how Artificial Intelligence might be applied in a variety of situations.
Social Interactions	
L1.IC.SI.01 Demonstrate and debate how computing increases and decreases connectivity and communication among people of various cultures.	Our courses contain relevant lessons on the global impact of computing. Students may additionally learn about Internet communication tools and social media.
Safety, Law & Ethics	
L1.IC.SLE.01 Describe the beneficial and harmful effects that intellectual property laws can have on innovation.	Our courses cover intellectual property laws, copyright considerations and various types of software licensing.
L2.IC.SLE.01 Debate laws and regulations that impact the development and use of software.	Our courses cover intellectual property laws, copyright considerations and various types of software licensing.
L1.IC.SLE.02 Describe and discuss the privacy concerns related to the large-scale collection and analysis of information about individuals (e.g., how websites collect and uses data) that may not be evident to users.	Students are taught to minimize their personal information shared online. They will learn about their personal digital footprint and understand that digital identities and online actions have long-term or permanent consequences.
L1.IC.SLE.03 Evaluate the social and economic consequences of how law and ethics interact with digital aspects of privacy, data, property, information, and identity.	Our courses cover computing ethics, copyrights, intellectual property, personal privacy and computer security. Relevant laws are discussed in each lesson.