CompuScholar, Inc.

Alignment to the Missouri Computer Science Performance Standards

11th - 12th Grade

Missouri Standards Information:

CS Page	Missouri Computer Science Education Page
Standards Link:	Computer Science Performance 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
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
Course Title:	**Unity Game Programming**, ISBN 978-0-9887070-7-8 Course Description and Syllabus

High schools can use any desired combination of CompuScholar courses to meet performance standards. Entire courses can be completed in sequential years or elements of selected courses can be combined in a single year.

Missouri Computer Science Performance Standards (11th - 12th Grade)

Computing Systems	COMPUSCHOLAR ALIGNMENT
Devices	
11-12.CS.D.01 Illustrate ways computing systems implement logic through hardware components.	N/A
Hardware & Software	
11-12.CS.HS.01 Describe and categorize roles of an operating system.	Our courses contain relevant descriptions of operating system options, features and interaction with applications.
Troubleshooting	
11-12.CS.T.01 Describe how hardware components facilitate logic, input, output and storage in computing systems.	Our courses contain overviews of the major hardware components (CPU, RAM, ALU, storage, etc.) that make up a computing system.

Network & The Internet	COMPUSCHOLAR ALIGNMENT
Network Communication & Organization	
11-12.NI.NCO.01 Analyze the relationship between routers, switches, servers, topology and addressing.	Our courses describe networking components and common network topology. IP addresses, MAC addresses and URLs are introduced for identification of devices and online resources.
11-12.NI.NCO.02 Describe key protocols and underlying processes of internet-based services (e.g., http/https and Simple Mail Transfer Protocol (SMTP)/internet Message Access Protocol (IMAP), routing protocols).	Our courses describe the structure of the Internet and relevant protocols (HTTP/S, FTP, SMTP, POP/IMAP).
11-12.NI.NCO.03 Explain how the characteristics of the internet influence the systems developed on it.	Our courses contain a relevant history of internet development and the resulting types of systems (e.g. web sites and web servers) that are deployed online.
Cybersecurity	
11-12.NI.C.01 Compare and refine ways in which software developers protect devices and information from unauthorized access.	Our courses contain lessons on relevant security topics, including authentication and validation of user input.
11-12.NI.C.02 Analyze cryptographic techniques to model the secure transmission of information.	Our courses describe general encryption concepts and highlight SSL/TLS as a mechanism for secure online data transmission.

Data Analysis	COMPUSCHOLAR ALIGNMENT
Storage	
11-12.DA.S.01 Compare different bit representations of data types, such as characters, Booleans and numbers while recognizing when using each data type is appropriate.	Our courses cover numbering systems such as binary, decimal and hexadecimal. They also describe specific programming data types and storage of relevant values.
Collection, Visualization & Transformation	
11-12.DA.CVT.01 Generate data sets that use a variety of data collection tools and analysis techniques to support a claim and/or communicate information.	Our courses contain opportunities to gather real-world data to manipulate, visualize and communicate to a target audience.
Inference & Models	
11-12.DA.IM.01 Evaluate the ability of models and simulations to test and support the refinement of hypotheses.	Our courses contain opportunities to study real-world phenomena and evaluate the impact of input data on resulting outputs.

Algorithms & Programming	COMPUSCHOLAR ALIGNMENT
Algorithms	
11-12.AP.A.01 Critically examine and trace classic algorithms (e.g., selection sort, insertion sort, binary search, linear search).	Our courses detail sorting algorithms (bubble, insertion, merge, selection) and sorting techniques (linear and binary search).

11-12.AP.A.02 Implement an artificial intelligence algorithm to interact with a human or solve a problem.	Our courses discuss AI and give students opportunities to study or develop relevant algorithms (e.g. game AI).
11-12.AP.A.03 Describe how artificial intelligence algorithms drive many software and physical systems (e.g., autonomous robots, computer vision, pattern recognition, text analysis).	Our courses contain lessons on the impact and spread of AI, including a variety of applications, legal and ethical concerns.
11-12.AP.A.04 Evaluate algorithms (e.g., sorting, searching) in terms of their efficiency and clarity.	Our courses describe "Big-O" performance notation and compare sorting and searching algorithms for performance and clarity.
Variables	
11-12.AP.V.01 Create problem solutions that utilize data structures (e.g., lists, arrays, ArrayLists).	Our programming courses cover simple data structures such as arrays and lists. Students will use each data structure in hands-on projects.
Control	
11-12.AP.C.01 Trace the execution of iteration (e.g., loops, recursion), illustrating output and changes in values of named variables.	Our courses contain chapters on debugging and program analysis, including using program tracing and debuggers to visualize program flow and track data changes.
Modularity	
11-12.AP.M.01 Construct solutions to problems using student-created components (e.g., procedures, modules, objects).	Our courses cover Object-Oriented Programming (OOP), modular programming with functions, and functional decomposition of complex tasks.
11-12.AP.M.02 Create programming solutions by reusing existing code (e.g., libraries, Application Programming Interface (APIs), code repositories).	Our courses teach students to use relevant libraries and APIs such as .NET, the Java Class Library, Python modules and the Unity framework.
11-12.AP.M.03 Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.	Our courses compare and contrast standard algorithms for large scale sorting and searching. They additionally cover AI applied specific tasks (e.g. self-driving cars).
Program Development	
11-12.AP.PD.01 Use integrated development environments (IDEs) and collaborative tools and practices (code documentation) in a software project.	Our courses teach students to use industry-standard IDEs such as Visual Studio, Eclipse and Unity.
11-12.AP.PD.02 Plan and develop programs using a development process (e.g., waterfall, iterative, spiral, rapid application development, agile).	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.
11-12.AP.PD.03 Identify and compare features of various programming languages that make them useful for solving problems and developing systems.	Our courses contain a description and comparison of programming languages, including purpose and suitability for specific tasks.
11-12.AP.PD.04 Design software using version control.	N/A

11-12.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 with a dedicated testing phase that asks students to develop and execute a test plan to verify program adherence to specifications.
11-12.AP.PD.06 Explain security issues that might lead to compromised computer programs.	Our courses contain lessons on relevant security topics and the need to verify user input.
11-12.AP.PD.07 Evaluate key qualities of a program through a process such as a code review.	Our courses show students how to analyze code and perform a systematic code review.

Impacts of Computing	**COMPUSCHOLAR ALIGNMENT**
Culture	
11-12.IC.C.01 Evaluate the impact of equity, access and influence on the distribution of computing resources in a global society.	Our courses contain lessons on the global impact of computing and equitable access of computing resources.
Safety, Law & Ethics	
11-12.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.