

CompuScholar, Inc.

Alignment to the Missouri Computer Science Performance Standards

9th - 10th 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 (9th - 10th Grade)

Computing Systems	COMPUSCHOLAR ALIGNMENT
Devices	
9-10.CS.D.01 Explain 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 storage.
Hardware & Software	
9-10.CS.HS.01 Explain the 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.

Troubleshooting	
9-10.CS.T.01 Develop, communicate and apply systematic troubleshooting strategies for 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.

Network & The Internet	COMPUSCHOLAR ALIGNMENT
Network Communication & Organization	
9-10.NI.NCO.02 Describe the issues that impact network functionality (e.g., bandwidth, load, delay, topology).	Network design considerations, including scalability, bandwidth, security and points of failure are addressed where relevant.
9-10.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) 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.
Cybersecurity	
9-10.NI.C.01 Compare physical and cybersecurity measures by evaluating trade-offs between the usability and security of a computing system.	Our courses contain chapters or lessons on security topics, including physical vs. electronic security mechanisms where relevant.
9-10.NI.C.02 Illustrate how sensitive data can be affected by attacks.	Our courses contain chapters or lessons on security topics, including physical vs. electronic security mechanisms where relevant.
9-10.NI.C.03 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.
9-10.NI.C.04 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	
9-10.DA.S.01 Translate and compare different bit representations of data types, such as characters, numbers and images.	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.

9-10.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	
9-10.DA.CVT.01 Create data visualizations to help others better understand real-world phenomena.	Our course contain opportunities to explore and represent real-world data in the form of equations, charts and graphs and similar tools.
9-10.DA.CVT.02 Explain the insights and knowledge gained from digitally processed data by using appropriate visualizations, notions and precise language.	Our team projects allow students to gather, modify and present data using relevant technologies and languages.
9-10.DA.CVT.03 Evaluate and refine computational artifacts to make them more usable and accessible.	Our courses contain relevant lessons on usability and accessibility when discussing human interaction with computing systems, programs or web sites.
Inference & Models	
9-10.DA.IM.01 Show the relationships between collected data elements using computational models.	N/A
9-10.DA.IM.02 Refine computational models to better represent the relationships among different elements of data collected from a phenomenon or process.	Student-driven projects give opportunities for analysis and representation of real-world data.

Algorithms & Programming	COMPUSCHOLAR ALIGNMENT
Algorithms	
9-10.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
Variables	
9-10.AP.V.01 Create problem solutions that utilize primitive variables (e.g., strings, ints, Booleans, doubles).	Our courses carefully describe how to create, initialize, update and use all types of primitive variables.
9-10.AP.V.02 Demonstrate the use of advanced variables (e.g., lists, arrays, objects) to simplify solutions, generalizing computational problems instead of repeatedly using primitive variables.	Our programming courses cover simple data structures such as arrays and lists. Students will use each data structure in hands-on projects.
Control	
9-10.AP.C.01 Apply the concepts 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.

Modularity	
9-10.AP.M.01 Break down a solution into procedures using systematic analysis and design utilizing functional abstraction.	Our courses cover Object-Oriented Programming (OOP), modular programming with functions, and functional decomposition of complex tasks down to manageable logical blocks.
9-10.AP.M.02 Create computational artifacts (file, graphic, video, audio) 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.
Program Development	
9-10.AP.PD.01 Using visual aids and documentation, illustrate the design elements and data flow (e.g., flowcharts, pseudocode) of the development of a program.	Our courses teach flowcharts as a way to help design and visualize the detailed steps in an algorithm.
9-10.AP.PD.04 Evaluate and refine computational artifacts to make them more user-friendly, efficient and/or accessible.	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.
9-10.AP.PD.02 Create a program by analyzing a problem and/or process, developing and documenting a solution, testing outcomes, debugging errors and adapting the program for a variety of users.	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.
9-10.AP.PD.03 While collaborating in a team, develop, test and refine 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.

Impacts of Computing	COMPUSCHOLAR ALIGNMENT
Culture	
9-10.IC.C.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.
9-10.IC.C.02 Test and refine computational artifacts to reduce bias and equity deficits.	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.
9-10.IC.C.03 Demonstrate how a given algorithm applies to problems across disciplines.	Our courses cover a variety of algorithms that can be applied to real-world situations in mathematics, physics, economics, etc.
Social Interactions	
9-10.IC.SI.01 Demonstrate through collaboration on a project how computing increases connectivity among people of various cultures.	Our courses contain team projects with opportunities for collaboration using relevant tools and self or teacher-directed topics.

9-10.IC.SI.02 Explain how the degrees of communication afforded by computing have impacted the nature and content of career fields.	Our courses contain career exploration exercises, focusing on relevant computing or software engineering opportunities.
Safety, Law & Ethics	
9-10.IC.SLE.01 Explain 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.
9-10.IC.SLE.02 Explain the privacy concerns related to the collection and analysis of information about individuals 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.
9-10.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.
9-10.IC.SLE.04 Define and classify 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.
9-10.IC.SLE.05 Identify and explain the potential impacts and implications of emerging technologies on larger social economic and political structures with evidence from credible sources.	Our courses contain relevant exploration of emerging technologies (e.g. AI), including discussion of relevant impact on social structures.