

CompuScholar, Inc.

Alignment to the Kentucky Computer Science Standards

6th - 8th Grade

Kentucky Standards Information:

CS Page	Kentucky Computer Science Education Page
Standards Link:	KY Academic 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

Middle schools will normally use the "Python Programming" course as a basis for coding concepts and incorporate elements from Digital Savvy and/or Web Design, as desired, to meet additional requirements.

Kentucky Computer Science Standards (Middle School)

Network & The Internet	COMPUSCHOLAR ALIGNMENT
M-NI-01: Model how different sets of rules (protocols) are used to transmit different types of data across	Our courses describe relevant Internet protocols and network topologies.
M-NI-02: Model how information is disguised using different methods of encryption to secure it during transmission from one point to another.	Our courses contain lessons on the purpose and types of encryption, encoding of data, and secure transmission protocols.
M-NI-03: Explain how physical and digital security practices and measures proactively address the threat of breaches to personal and private data.	Our courses contain relevant lessons on security topics, including physical and electronic threats and mitigation strategies.

Data Analysis	COMPUSCHOLAR ALIGNMENT
M-DA-01: Store data using multiple encoding methods.	Our courses cover numbering systems such as binary, decimal and hexadecimal. We also discuss encoding of data, including ASCII character and color representations.
M-DA-02: Collect data using computational tools and transform the data to make it more useful and reliable.	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.

M-DA-03: Refine computational models based on the data they have generated.	Our courses contain lessons that allow students to experiment with input data and parameters to observe changed results.
---	--

Algorithms & Programming	COMPUSCHOLAR ALIGNMENT
M-AP-01: Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.	Our courses describe team roles and project planning. Team projects incorporate real-world experience with teamwork and basic SDLC concepts.
M-AP-02: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	Our courses cover Object-Oriented Programming (OOP), modular programming with functions, and breaking complex tasks down to manageable logical blocks.
M-AP-03: Seek and incorporate feedback from team members and users to refine a solution that meets user needs.	Our courses contain team projects that include a refinement phase. Students will receive feedback from peers and incorporate that feedback into the final project.
M-AP-04: Create flowcharts and/or pseudocode to address complex problems as algorithms.	Our courses describe how to use flowcharts to design algorithms to solve specific problems.
M-AP-05: Create clearly named variables that represent different data types and perform operations on their values.	Our courses carefully describe how to create, initialize, update and use variable data.
M-AP-06: Create procedures with parameters to organize code and make it easier to reuse.	Our courses include opportunities for students to create their own functions/methods and organize related code into objects for reuse.
M-AP-07: Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	Our courses cover traditional flow control structures (conditionals, loops, functions) and the trade-offs in design, including selecting between appropriate flow control logic.
M-AP-08: Incorporate existing code, media, and libraries into original programs, and give attribution.	Our courses teach students how to use existing libraries (e.g. Python modules) and leverage those features for new, creative programs.
M-AP-09: Systematically test and refine programs using a range of test cases.	Our courses contain team projects that include a testing phase using a written test plan.
M-AP-10: Document programs in order to make them easier to follow, test, and debug.	Our courses give students the opportunity to create requirements, design and test documents in addition to well-commented code.
M-AP-11: Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries.	Our courses describe licensing models such as open source, freeware, shareware, and discuss intellectual property rights.
M-AP-12: Develop a process creating a computational artifact that leads to a minimum viable product followed by reflection, analysis, and iteration.	Our courses cover basic SDLC models and include project opportunities to complete simple waterfall steps, including feedback and improvement.

Impacts of Computing	COMPUSCHOLAR ALIGNMENT
M-IC-02: Compare the positive & negative effects of computing technologies on society.	Our courses contain relevant lessons on the impact of computing on society, personal life and career choices.
M-IC-01: Discuss issues of bias and accessibility in existing technologies.	Our courses contain lessons on the global impact of computing, bias and the digital divide. We also describe accessibility for equitable program use.
M-IC-03: Collaborate with others using appropriate tools at the local, national, and/or international levels.	Team projects allow students to work together to create programs or digital artifacts.
M-IC-04: Discuss the benefits and consequences of making information either public or private.	Our courses discuss the security of personal information in online environments.

Computing Systems	COMPUSCHOLAR ALIGNMENT
M-CS-01: Recommend improvements to the design of computing devices based on an analysis of how users interact with the devices.	Some courses (e.g. Web Design) contain lessons on user interface design, storyboarding, and improvement of the user experience.
M-CS-02: Design projects that combine hardware and software components to collect and exchange data.	N/A (CompuScholar courses avoid requiring hardware components due to the logistical and cost burdens that hardware places on schools).
M-CS-03: Identify and fix problems with computing devices and their components systematically.	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.