

## CompuScholar, Inc.

### Alignment to the Arizona Computer Science Standards

#### 6th - 8th Grades

#### Arizona Standards Information:

CS Page	<a href="#">Arizona Computer Science Education Page</a>
Standards Link:	<a href="#">Arizona Computer Science Standards 6th - 8th Grade (2018)</a>

#### CompuScholar Courses in this Grade Band:

Course Title:	<b>Digital Savvy</b> , ISBN 978-0-9887070-8-5 <a href="#">Course Description and Syllabus</a>
Course Title:	<b>Web Design</b> , ISBN 978-0-9887070-3-0 <a href="#">Course Description and Syllabus</a>
Course Title:	<b>Python Programming</b> , ISBN 978-1-946113-00-9 <a href="#">Course Description and Syllabus</a>

Middle schools will normally use a combination of our "Digital Savvy", "Python Programming" and "Web Design" courses as desired to meet 6th - 8th grade requirements. Entire courses can be completed in sequential years or elements of selected courses can be combined in a single year.

#### Arizona Computer Science Standards (6th - 8th Grades)

Computing Systems	COMPUSCHOLAR ALIGNMENT
<b>Devices</b>	
6.CS.D.1 - Compare computing device designs based on how humans interact with them.	Our Web Design course contains lessons on user interface design, storyboarding, and improvement of the user experience.
7.CS.D.1 - Identify some advantages, disadvantages, and consequences with the design of computer devices based on an analysis of how users interact with devices.	
8.CS.D.1 - Improve the design of computing devices based on an analysis of how users interact them, and consider unintended consequences.	
<b>Hardware &amp; Software</b>	
6.CS.HS.1 - Explain how hardware and software can be used 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).
7.CS.HS.1 - Design projects that combine hardware and software to collect and exchange data.	
8.CS.HS.1 - Design and evaluate projects that combine hardware and software components to collect and exchange data	

<b>Troubleshooting</b>	
6.CS.T.1 - Identify problems that can occur in computing devices and their components within a system.	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.
7.CS.T.1 - Evaluate strategies to fix problems with computing devices and their components within a system.	
8.CS.T.1 - Systematically identify and develop strategies to fix problems with computing devices and their components.	

<b>Network &amp; The Internet</b>	<b>COMPUSCHOLAR ALIGNMENT</b>
<b>Network Communication &amp; Organization</b>	
6.NI.NCO.1 - Discuss how protocols are used in transmitting data across networks and the Internet.	Our courses describe relevant Internet protocols (HTTP/HTTPS, POP/IMAP, SMTP, FTP) and network topologies.
7.NI.NCO.1 - Compare and contrast models to understand the many protocols used for data transmission.	
8.NI.NCO.1 - Develop models to illustrate the role of protocols in transmitting data across networks and the Internet.	
<b>Cybersecurity</b>	
6.NI.C.1I - Identify multiple methods of encryption to secure the transmission of information	Our courses contain lessons on relevant security topics such as encryption (including SSL/TLS) and protection of online personal information.
7.NI.C.1 - Evaluate multiple methods of encryption for the secure transmission of information.	
8.NI.C.1 - Apply multiple methods of encryption to model the secure transmission of information.	
6.NI.C.2 - Identify different physical and digital security measures that protect electronic information.	Our courses contain relevant lessons on security topics, including physical and electronic threats and mitigation strategies.
7.NI.C.2 - Explain how physical and digital security measures protect electronic information.	
8.NI.C.2 - Evaluate how various physical and digital security measures protect electronic information and how a lack of such measures could lead to vulnerabilities	

<b>Data Analysis</b>	<b>COMPUSCHOLAR ALIGNMENT</b>
<b>Storage</b>	
6.DA.S.1 - Identify multiple encoding schemes used to represent data, including binary and ASCII.	Our courses cover numbering systems such as binary, decimal and hexadecimal. We also discuss encoding of data, including ASCII character and color representations.
7.DA.S.1 - Use multiple encoding schemes to represent data, including binary and ASCII.	
8.DA.S.1 - Represent data using multiple encoding schemes including binary and ASCII.	

<b>Collection, Visualization &amp; Transformation</b>	
6.DA.CVT.1 - Compare different computational tools used to collect, analyze and present data that is meaningful and useful.	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.
7.DA.CVT.1 - Collect and analyze data using computational tools to create models that are meaningful and useful.	
8.DA.CVT.1 - Collect data using computational tools and transform the data to make it more meaningful and useful.	
<b>Inference &amp; Models</b>	
6.DA.IM.1 - Discuss the validity of a computational model based on the reliability of the data.	Our courses contain lessons that allow students to experiment with input data and parameters to observe changed results.
7.DA.IM.1 - Use computational models and determine the reliability and validity of data they generate.	
8.DA.IM.1 - Design computational models and evaluate them based on the reliability and validity of the data they generate.	

<b>Algorithms &amp; Programming</b>	<b>COMPUSCHOLAR ALIGNMENT</b>
<b>Algorithms</b>	
6.AP.A.1 - Identify planning strategies such as flowcharts or pseudocode, to simulate algorithms that solve	Our courses describe how to use flowcharts to design algorithms to solve specific problems.
7.AP.A.1 - Use planning strategies, such as flowcharts or pseudocode, to develop algorithms to address complex problems.	
8.AP.A.1 - Develop planning strategies, such as flowcharts or pseudocode, to develop algorithms to address complex problems.	
<b>Variables</b>	
6.AP.V.1 - Identify 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.
7.AP.V.1 - Compare and contrast variables that represent different data types and perform operations on their values	
8.AP.V.1 - Create named variables that represent different data types and perform operations on their values.	

<b>Control</b>	
6.AP.C.1 - Design 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.
7.AP.C.1 - Design and develop programs that combine control structures, including nested loops and compound conditionals.	
8.AP.C.1 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	
<b>Modularity</b>	
6.AP.M.1 - Decompose problems 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.
7.AP.M.1 - Decompose problems into parts to facilitate the design, implementation, and review of programs.	
8.AP.M.1 - Decompose problems into parts to facilitate the design, implementation, and review of programs.	
6.AP.M.2 - Use procedures 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.
7.AP.M.2 - Use procedures with parameters to organize code and make it easier to reuse.	
8.AP.M.2 - Create procedures with parameters to organize code and make it easier to reuse.	
<b>Program Development</b>	
6.AP.PD.1 - 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.
7.AP.PD.1 - Seek and incorporate feedback from team members and users to refine a solution that meets user needs.	
8.AP.PD.1 - Seek and incorporate feedback from team members and users to refine a solution that meets user needs.	
6.AP.PD.2 - Incorporate existing code into programs and give attribution.	Our courses teach students how to use existing, language-specific libraries (e.g. Python modules) and leverage those features for new, creative programs. Our courses additionally teach respect for intellectual property and licensing considerations.
7.AP.PD.2 - Incorporate existing code and media into programs, and give attribution.	
8.AP.PD.2 - Incorporate existing code, media, and libraries into original programs, and give attribution.	
6.AP.PD.3 - Test programs using a range of inputs and identify expected outputs.	Our courses contain team projects that include a testing phase using a written test plan.
7.AP.PD.3 - Systematically test and refine programs using a range of possible inputs.	
8.AP.PD.3 - Systematically test and refine programs using a range of possible inputs.	

6.AP.PD.4 - Maintain a timeline with specific tasks while collaboratively developing computational artifacts.	Our courses describe team roles and project planning. Team projects incorporate real-world experience with teamwork and basic SDLC concepts.
7.AP.PD.4 - Distribute and execute tasks while maintaining a project timeline when collaboratively developing computational artifacts.	
8.AP.PD.4 - Distribute and execute tasks while maintaining a project timeline when collaboratively developing computational artifacts.	
6.AP.PD.5 - 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.
7.AP.PD.5 - Document programs to make them easier to follow, test, and debug.	
8.AP.PD.5 - Document programs to make them easier to follow, test, and debug.	

Impacts of Computing	COMPUSCHOLAR ALIGNMENT
<b>Culture</b>	
6.IC.C.1 - Identify some of the tradeoffs associated with computing technologies that can affect people's everyday activities and career options.	Our courses contain relevant lessons on the impact of computing on society, personal life and career choices.
7.IC.C.1 - Explain how some of the tradeoffs associated with computing technologies can affect people's everyday activities and career options.	
8.IC.C.1 - Compare and contrast tradeoffs associated with computing technologies that affect people's everyday activities and career options.	
6.IC.C.2 - Identify issues of bias and accessibility in the design of 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.
7.IC.C.2 - Discuss how bias and accessibility issues can impact the functionality of existing technologies.	
8.IC.C.2 - Develop a solution to address an issue of bias or accessibility in the design of existing technologies.	
<b>Social Interactions</b>	
6.IC.SI.1 - Identify the advantages of creating a computational product by collaborating with others using digital technologies.	Team projects allow students to work together to create programs or digital artifacts.
7.IC.SI.1 - Describe the process for creating a computational product by collaborating with others using digital technologies.	
8.IC.SI.1 - Collaborate with contributors by using digital technologies when creating a computational product.	

<b>Safety, Law &amp; Ethics</b>	
6.IC.SLE.1 - Describe how some digital information can be public or can be kept private and secure.	Our courses discuss the security of personal information in online environments.
7.IC.SLE.1 - Identify the benefits and risks associated with sharing information digitally.	
8.IC.SLE.1 - Evaluate the benefits and risks associated with sharing information digitally.	