Arizona CS Standards Alignment with CodeX Curriculum

	Unit 1	Unit 2	Unit 3
Computing Systems			
6.CS.D.1 Compare computing device designs based on how humans interact with them.			
6.CS.HS.1 Explain how hardware and software can be used to collect and exchange data.	[1]		
6.CS.T.1 Identify problems that can occur in computing devices and their components within a system.	[2]		
Networks and the Internet			
6.NI.C.1 Identify multiple methods of encryption to secure the transmission of information.			
6.NI.C.2 Identify different physical and digital security measures that protect electronic information.			
6.NI.NCO.1 Discuss how protocols are used in transmitting data across networks and the Internet.			
Data and Analysis			
6.DA.CVT.1 Compare different computational tools used to collect, analyze and present data that is meaningful and useful.			
6.DA.S.1 Identify multiple encoding schemes used to represent data, including binary and ASCII.			
6.DA.IM.1 Discuss the validity of a computational model based on the reliability of the data.			
Algorithms and Programming			
6.AP.A.1 Identify planning strategies such as flowcharts or pseudocode, to simulate algorithms that solve problems.	[3]		
6.AP.V.1 Identify variables that represent different data types and perform operations on their values.	[4]		
6.AP.C.1 Design programs that combine control structures, including nested loops and compound conditionals.	[5]		
6.AP.M.1 Decompose problems into parts to facilitate the design, implementation, and review of programs.	[6]		
6.AP.M.2 Use procedures to organize code and make it easier to reuse.	[7]		
6.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.	[8]		
6.AP.PD.3 Test programs using a range of inputs and identify expected outputs.			
6.AP.PD.4 Maintain a timeline with specific tasks while collaboratively developing computational artifacts.			
6.AP.PD.5 Document programs in order to make them easier to follow, test, and debug.	[9]		
Impacts of Computing			
6.IC.C.1 Identify some of the tradeoffs associated with computing technologies that can affect people's everyday activities and career options.			
6.IC.C.2 Identify issues of bias and accessibility in the design of existing technologies.			
6.IC.SI.1 Identify the advantages of creating a computational product by collaborating with others using digital technologies.			
6.IC.SLE.1 Describe how some digital information can be public or can be kept private and secure.			

Arizona CS Standards Alignment with CodeX Curriculum

	Unit 1	Unit 2	Unit 3
Computing Systems			
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.			
7.CS.HS.1 Design projects that combine hardware and software to collect and exchange data.			
7.CS.T.1 Evaluate strategies to fix problems with computing devices and their components within a system.	[10]		
Networks and the Internet			
7.NI.C.1 Evaluate multiple methods of encryption for the secure transmission of information.			
7.NI.C.2 Explain how physical and digital security measures protect electronic information.			
7.NI.NCO.1 Compare and contrast models to understand the many protocols used for data transmission.			
Data and Analysis			
7.DA.CVT.1 Collect and analyze data using computational tools to create models that are meaningful and useful.			
7.DA.S.1 Use multiple encoding schemes to represent data, including binary and ASCII.			
7.DA.IM.1 Use computational models and determine the reliability and validity of data they generate.			
Algorithms and Programming			
7.AP.A.1 Use planning strategies, such as flowcharts or pseudocode, to develop algorithms to address complex problems.	[11]		
7.AP.V.1 Compare and contrast variables that represent different data types and perform operations on their values.	[12]		
7.AP.C.1 Design and develop programs that combine control structures, including nested loops and compound conditionals.	[13]		
7.AP.M.1 Decompose problems into parts to facilitate the design, implementation, and review of programs.	[14]		
7.AP.M.2 Use procedures with parameters to organize code and make it easier to reuse.		[15]	
7.AP.PD.1 Seek and incorporate feedback from team members and users to refine a solution that meets user needs.			
7.AP.PD.2 Incorporate existing code and media into programs, and give attribution.	[16]		
7.AP.PD.3 Systematically test and refine programs using a range of possible inputs.			
7.AP.PD.4 Distribute and execute tasks while maintaining a project timeline when collaboratively developing computational artifacts.			
7.AP.PD.5 Document programs to make them easier to follow, test, and debug.	[17]		
Impacts of Computing			
7.IC.C.1 Explain how some of the tradeoffs associated with computing technologies can affect people's everyday activities and career options.			
7.IC.C.2 Discuss how bias and accessibility issues can impact the functionality of existing technologies.			
7.IC.SI.1 Describe the process for creating a computational product by collaborating with others using digital technologies.			
7.IC.SLE.1 Identify the benefits and risks associated with sharing information digitally.			

Arizona CS Standards Alignment with CodeX Curriculum

	Unit 1	Unit 2	Unit 3
Computing Systems			
8.CS.D.1 Improve the design of computing devices based on an analysis of how users interact them, and consider unintended consequences.			
8.CS.HS.1 Design and evaluate projects that combine hardware and software components to collect and exchange data.			
8.CS.T.1 Systematically identify and develop strategies to fix problems with computing devices and their components.	[18]		
Networks and the Internet			
8.NI.C.1 Apply multiple methods of encryption to model the secure transmission of 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.			
8.NI.NCO.1 Develop models to illustrate the role of protocols in transmitting data across networks and the Internet.			
Data and Analysis			
8.DA.CVT.1 Collect data using computational tools and transform the data to make it more meaningful and useful.			
8.DA.S.1 Represent data using multiple encoding schemes including binary and ASCII.			
8.DA.IM.1 Design computational models and evaluate them based on the reliability and validity of the data they generate.			
Algorithms and Programming			
8.AP.A.1 Develop planning strategies, such as flowcharts or pseudocode, to develop algorithms to address complex problems.	[19]		
8.AP.V.1 Create named variables that represent different data types and perform operations on their values.	[20]		
8.AP.C.1 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	[21]		
8.AP.M.1 Decompose problems into parts to facilitate the design, implementation, and review of programs.	[22]		
8.AP.M.2 Create procedures with parameters to organize code and make it easier to reuse.		[23]	
8.AP.PD.1 Seek and incorporate feedback from team members and users to refine a solution that meets user needs.			
8.AP.PD.2 Incorporate existing code, media, and libraries into original programs, and give attribution.	[24]		
8.AP.PD.3 Systematically test and refine programs using a range of possible inputs.			
8.AP.PD.4 Distribute and execute tasks while maintaining a project timeline when collaboratively developing computational artifacts.			
8.AP.PD.5 Document programs to make them easier to follow, test, and debug.	[25]		
Impacts of Computing			
8.IC.C.1 Compare and contrast tradeoffs associated with computing technologies that affect people's everyday activities and career options.			
8.IC.C.2 Develop a solution to address an issue of bias or accessibility in the design of existing technologies.			
8.IC.SI.1 Collaborate with contributors by using digital technologies when creating a computational product.			
8.IC.SLE.1 Evaluate the benefits and risks associated with sharing information digitally.			

Arizona CS Standards Alignment with CodeX Curriculum			
	Unit 1	Unit 2	Unit 3
Computing Systems			
HS.CS.D.1 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.			
HS.CS.HS.1 Describe levels of abstraction and interactions between application software, system software, and hardware layers.			
HS.CS.T.1 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.	[26]		
Networks and the Internet			
HS.NI.C.1 Describe how sensitive data can be affected by malware and other attacks.			
HS.NI.C.2 Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts.			
HS.NI.C.3 Compare various security measures, considering tradeoffs between the usability and security of a computing system.			
HS.NI.NCO.1 Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.			
Data and Analysis			
HS.DA.CVT.1 Create interactive data visualizations using software tools to help others better understand real-world phenomena.			
HS.DA.S.1 Translate between different bit representations of real-world phenomena, such as characters, numbers, and images.			
HS. DA.S.2 Evaluate the tradeoffs in how and where data is stored.			
HS.DA.IM.1 Analyze computational models to better understand real-world phenomena.			
Algorithms and Programming			
HS.AP.A.1 Create prototypes that use algorithms for practical intent, personal expression, or to address a societal issue.			
HS.AP.V.1 Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.		[27]	
HS.AP.C.1 Justify the selection of specific control structures and explain the benefits and drawbacks of choices made, when tradeoffs involve readability and program performance.			
HS.AP.C.2 Use events that initiate instructions to design and iteratively develop computational artifacts			
HS.AP.M.1 Decompose problems into smaller components using constructs such as procedures, modules, and/or objects.			
HS.AP.M.2 Use procedures within a program, combinations of data and procedures, or independent but interrelated programs to design and iteratively develop computational artifacts.			
HS.AP.PD.1 Evaluate and refine computational artifacts to make them more usable and accessible.			
HS.AP.PD.2 Use team roles and collaborative tools to design and iteratively develop computational artifacts.			
HS.AP.PD.3 Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.	[28]		
Impacts of Computing			
HS.IC.C.1 Evaluate the ways access to computing impacts personal, ethical, social, economic, and cultural practices.			
HS.IC.C.2 Test and refine computational artifacts to reduce bias and equity deficits.			
HS.IC.C.3 . Demonstrate ways a given algorithm applies to problems across disciplines.			
HS.IC.SI.1 Analyze the impact of collaborative tools and methods that increase social connectivity.			
HS.IC.SLE.1 Explain the beneficial and harmful effects that intellectual property laws can have on innovation.			
HS.IC.SLE.2 Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users.			
HS.IC.SLE.3 Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.			

[1] Mission 2 explains this

[2] Debugger introduced in Mission 2 as well as troubleshooting techniques

[3] Introduced in the teachers' manual

[4] 3.8 introduces the use of variables

[5] Mission 6 introduces the use of nested loops but does not discuss them
Mission 9 introduces compound conditionals

[6] Code Tracing Chart
Pseudocode
Flowcharts
All introduced in the teachers' manual

[7] Mission 5 discusses readability of code
7.6 discusses readability for maintenance reasons and reuse

[8] Libraries are explained and used in all lessons

[9] Code Tracing Charts are discussed in teachers' manual

[10] Debugger introduced in Mission 2 as well as troubleshooting techniques

[11] Introduced in the teachers' manual

[12] Mission 4 begins the use of data types and storing them in different variables

[13] Mission 6 introduces the use of nested loops but does not discuss them
Mission 9 introduces compound conditionals

[14] Code Tracing Chart
Pseudocode
Flowcharts
All introduced in the teachers' manual

[15] Mission 9 introduces the use of parameters and arguments

[16] Libraries are explained and used in all lessons

[17] Code Tracing Charts are discussed in teachers' manual

[18] Debugger introduced in Mission 2 as well as troubleshooting techniques

[19] Introduced in the teachers' manual

[20] Mission 4 begins the use of data types and storing them in different variables

[21] Mission 6 introduces the use of nested loops but does not discuss them
Mission 9 introduces compound conditionals

[22] Code Tracing Chart
Pseudocode
Flowcharts
All introduced in the teachers' manual

[23] Mission 9 introduces the use of parameters and arguments

[24] Libraries are explained and used in all lessons

[25] Code Tracing Charts are discussed in teachers' manual

[26] Utilize the Code Tracing Chart and share with others

[27] 7.5 introduces the use of lists

[28] Code Tracing Charts are discussed in teachers' manual