1st Quarter (44 Days)			
Resources: code.org			
Week	**Unit/Lesson**	**Learning Objectives**	**TEKs**
Week 1	**Intro to Problem Solving.** **The Problem Solving Process.** **Exploring Problem Solving.**	1B-AP-08 - Compare and refine multiple algorithms for the same task and determine which is the most appropriate. 1B-AP-11 - Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. 1B-AP-16 - Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation and review stages of program development.	1D, 2A, 2B, 2C, 4D
Week 2	**What is a Computer?** **Input and Output.** **Processing.**	1B-CS-01 - Describe how internal and external parts of computing devices function to form a system. 1B-CS-02 - Model how computer hardware and software work together as a system to accomplish tasks.	6A, 6B, 6C, 6D, 6E, 6F
Week 3	**Apps and Storage,** **Project: Propose an App.**	2-AP-10 - Use flowcharts and/or pseudocode to address complex problems as algorithms. 2-IC-20 - Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options. 2-AP-15 - Seek and incorporate feedback from team members and users to refine a solution that meets user needs. 2-AP-18 - Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts. 2-CS-02 - Design projects that combine hardware and software components to collect and exchange data.	1A, 1F, 5A, 5B, 5C, 5D, 5E, 5F
Week 4	**Exploring Websites.** **Websites for Expression.** **Intro to HTML.**	2-IC-20 - Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 1B-IC-18 - Discuss computing technologies that have changed the world and express how those technologies influence, and are influenced by, cultural practices. 1B-AP-11 - Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.	
Week 5	**Headings.** **Digital Footprint.** **Lists.** **Intellectual Property and Images.** **Clean Code and Debugging.**	1B-AP-11 - Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. 1B-AP-15 - Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. 2-IC-20 - Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options. 2-IC-23 - Describe tradeoffs between allowing information to be public and keeping information private and secure.	

		1B-NI-05 - Discuss real-world cybersecurity problems and how personal information can be protected. 1B-AP-12 - Modify, remix or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution. 3A-AP-20 - Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries. 1B-IC-21 - Use public domain or creative commons media and refrain from copying or using material created by others without permission. 2-IC-23 - Describe tradeoffs between allowing information to be public and keeping information private and secure. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	
Week 6	**Project: Multi-Page Website**	2-AP-15 - Seek and incorporate feedback from team members and users to refine a solution that meets user needs. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug. 1B-IC-21 - Use public domain or creative commons media and refrain from copying or using material created by others without permission.	2a, 2b, 2c
Week 7	**Styling Text with CSS** **Styling Elements with CSS** **Sources and Search Engines** **RGB Colors and Classes**	2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug. 2-IC-20 - Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options. 2-IC-21 - Discuss issues of bias and accessibility in the design of existing technologies. 2-IC-23 - Describe tradeoffs between allowing information to be public and keeping information private and secure. 2-AP-17 - Systematically test and refine programs using a range of test cases.	
Week 8	**Review and Assessment**	**1st Benchmark**	
Week 9	**Project: Personal Portfolio Website**	2-AP-15 - Seek and incorporate feedback from team members and users to refine a solution that meets user needs. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-18 - Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	1E, 1F, 1G
Week 10			

2nd Quarter (43 Days)			
Resources:			
Week	**Unit/Lesson**	**Learning Objectives**	**TEKs**
Week 1	**Programming for Entertainment** **Plotting Shapes** **Drawing in Game Lab** **Shapes and Randomization**	2-IC-21 - Discuss issues of bias and accessibility in the design of existing technologies. 2-AP-10 - Use flowcharts and/or pseudocode to address complex problems as algorithms. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	6A, 6B, 6C, 6D, 6E, 6F
Week 2	**Variables** **Sprites** **The Draw Loop** **Counter Pattern Unplugged**	2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-12 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-10 - Use flowcharts and/or pseudocode to address complex problems as algorithms.	G
Week 3	**Sprite Movement** **Booleans Unplugged** **Booleans and Conditionals** **Conditionals and User Input** **Other Forms of Input**	2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug. 2-AP-10 - Use flowcharts and/or pseudocode to address complex problems as algorithms.	6A, 6B, 6C, 6D, 6E, 6F
Week 4	**Project: Interactive Card**	2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-15 - Seek and incorporate feedback from team members and users to refine a solution that meets user needs. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution.	4I, 4J, 4K, 4L

		2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-18 - Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug. **IOWA/ITBS Complete Battery Gr 3-8**	
Week 5	**Velocity** **Collision Detection** **Complex Sprite Movement** **Collisions**	2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	1H, 1I, 1J
Week 6	**Functions** **The Game Design Process** **Using the Game Design Process**	2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-14 - Create procedures with parameters to organize code and make it easier to reuse. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	1A, 1D, 1H, 1K, 1L
Week 7	**Review and Assessment**	**2nd Benchmark**	
Week 8 **Week 9**	**Project: Design a Game**	2-AP-10 - Use flowcharts and/or pseudocode to address complex problems as algorithms. 2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-15 - Seek and incorporate feedback from team members and users to refine a solution that meets user needs. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-17 - Systematically test and refine programs using a range of test cases.	4K, 4L, 4N, 4R, 4S, 4T

		2-AP-18 - Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	

3rd Quarter (43 Days)			
Resources:			
Week	**Unit/Lesson**	**Learning Objectives**	**TEKs**
Week 1	**Analysis of Design** **Understanding Your User** **User-Centered Design** **Micro Activity**	2-CS-01 - Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices. 2-CS-02 - Design projects that combine hardware and software components to collect and exchange data. 2-IC-20 - Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options. 2-IC-21 - Discuss issues of bias and accessibility in the design of existing technologies.	4A, 4F, 4G, 6E
Week 2	**User Interfaces** **Feedback and Testing** **Identifying User Needs**	2-AP-10 - Use flowcharts and/or pseudocode to address complex problems as algorithms. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-CS-01 - Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices. 2-AP-10 - Use flowcharts and/or pseudocode to address complex problems as algorithms. 2-AP-15 - Seek and incorporate feedback from team members and users to refine a solution that meets user needs. 2-IC-21 - Discuss issues of bias and accessibility in the design of existing technologies. 2-IC-22 - Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.	1I, 2B, 2C
Week 3	**Project: Paper Prototype**	2-AP-10 - Use flowcharts and/or pseudocode to address complex problems as algorithms. 2-AP-15 - Seek and incorporate feedback from team members and users to refine a solution that meets user needs. 2-AP-17 - Systematically test and refine programs using a range of test cases.	4K,4L, 4N 2D, 4D
Week 4	**Designing Apps for Good** **Market Research** **Paper Prototypes**	2-IC-20 - Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options. 2-IC-21 - Discuss issues of bias and accessibility in the design of existing technologies. 2-AP-10 - Use flowcharts and/or pseudocode to address complex problems as algorithms. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	4N, 4H, 4J

Week 5	**Prototype Testing** **Digital Design** **Linking Screens**	2-AP-10 - Use flowcharts and/or pseudocode to address complex problems as algorithms. 2-AP-15 - Seek and incorporate feedback from team members and users to refine a solution that meets user needs. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-CS-01 - Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices. 2-DA-08 - Collect data using computational tools and transform the data to make it more useful and reliable. 2-DA-09 - Refine computational models based on the data they have generated. IC - Impacts of Computing 2-IC-22 - Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-18 - Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-14 - Create procedures with parameters to organize code and make it easier to reuse. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-18 - Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	4A, 4B, 4C, 4D, 4E, 4F, 4G, 4H, 4I, 4J, 4K, 4L, 4M, 4N, 4O, 4P, 4Q, 4R, 4S, 4T, 4U, 4V, 4W
Week 6	**Testing the App** **Improving and Iterating**	2-AP-15 - Seek and incorporate feedback from team members and users to refine a solution that meets user needs. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-18 - Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts. 2-CS-01 - Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices. 2-DA-08 - Collect data using computational tools and transform the data to make it more useful and reliable. 2-IC-22 - Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-14 - Create procedures with parameters to organize code and make it easier to reuse. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution.	

		2-AP-19 - Document programs in order to make them easier to follow, test, and debug. 2-DA-09 - Refine computational models based on the data they have generated.	
Week 7	**Review and Assessment**	**3rd Benchmark**	
Week 8 **Week 9**	**Project: App Completion and Presentation**	2-AP-19 - Document programs in order to make them easier to follow, test, and debug. 2-CS-01 - Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices. 2-DA-09 - Refine computational models based on the data they have generated. IC - Impacts of Computing 2-IC-20 - Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options. 2-IC-21 - Discuss issues of bias and accessibility in the design of existing technologies. 2-IC-22 - Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.	2E, 4S

4th Quarter (46 Days)			
Resources:			
Week	**Unit/Lesson**	**Learning Objectives**	**TEKs**
Week 1	**Representation Matters** **Patterns and Representation** **ASCII and Binary Representation** **Representing Images**	2-DA-07 - Represent data using multiple encoding schemes.	6D, 6M
Week 2	**Representing Numbers** **Keeping Data Secret** **Combining Representations** **Create a Representation**	2-DA-07 - Represent data using multiple encoding schemes. 2-NI-05 - Explain how physical and digital security measures protect electronic information. 2-NI-06 - Apply multiple methods of encryption to model the secure transmission of information. 2-AP-10 - Use flowcharts and/or pseudocode to address complex problems as algorithms. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-DA-07 - Represent data using multiple encoding schemes.	

Week 3	**Problem Solving and Data** **Problem Solving with Big Data** **Structuring Data**	2-DA-08 - Collect data using computational tools and transform the data to make it more useful and reliable. 2-IC-20 - Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options. 2-IC-23 - Describe tradeoffs between allowing information to be public and keeping information private and secure.	3c, 3D, 3G, 3H, 3K 6E
Week 4	**Making Decisions with Data** **Interpreting Data** **Automating Data Decisions**	**April 9: STAAR- Writing** 2-DA-08 - Collect data using computational tools and transform the data to make it more useful and reliable. 2-AP-10 - Use flowcharts and/or pseudocode to address complex problems as algorithms. 2-IC-22 - Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.	
Week 5	**Innovations in Computing** **Designing Screens with Code** **The Circuit Playground**	2-IC-20 - Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options. 2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug. 2-CS-02 - Design projects that combine hardware and software components to collect and exchange data. 2-CS-03 - Systematically identify and fix problems with computing devices and their components.	
Week 6	**Input Unplugged** **Board Events** **Getting Properties** **Analog Input**	2-AP-10 - Use flowcharts and/or pseudocode to address complex problems as algorithms. 2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug. 2-CS-01 - Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices. 2-CS-02 - Design projects that combine hardware and software components to collect and exchange data. 2-CS-03 - Systematically identify and fix problems with computing devices and their components. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug.	1C,2B, 4E

Week 7	**The Program Design Process** **Project: Make a Game**	2-AP-14 - Create procedures with parameters to organize code and make it easier to reuse. 3A-AP-16 - Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions. 2-CS-03 - Systematically identify and fix problems with computing devices and their components. 2-AP-10 - Use flowcharts and/or pseudocode to address complex problems as algorithms. 2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-15 - Seek and incorporate feedback from team members and users to refine a solution that meets user needs. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-18 - Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug. 2-CS-01 - Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices. 2-CS-02 - Design projects that combine hardware and software components to collect and exchange data..	3D, 4A 6H, 6I, 6J 6P, 6Q
Week 8	**Arrays and Color LEDs** **Making Music** **Arrays and For Loops** **Accelerometer**	2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug. 2-CS-02 - Design projects that combine hardware and software components to collect and exchange data. 2-AP-12 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-14 - Create procedures with parameters to organize code and make it easier to reuse. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-CS-01 - Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices. 2-CS-03 - Systematically identify and fix problems with computing devices and their components. **May 13: STAAR- Math May 14: STAAR- Reading**	
Week 9	**Review and assessment**	**Final Benchmark**	

Week 10	**Project: Prototype an Innovation**	2-AP-10 - Use flowcharts and/or pseudocode to address complex problems as algorithms. 2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 - Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 2-AP-13 - Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. 2-AP-15 - Seek and incorporate feedback from team members and users to refine a solution that meets user needs. 2-AP-16 - Incorporate existing code, media, and libraries into original programs, and give attribution. 2-AP-17 - Systematically test and refine programs using a range of test cases. 2-AP-18 - Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts. 2-AP-19 - Document programs in order to make them easier to follow, test, and debug. 2-CS-01 - Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices. 2-CS-02 - Design projects that combine hardware and software components to collect and exchange data. 2-CS-03 - Systematically identify and fix problems with computing devices and their components.	3D, 4A 6H, 6I, 6J 6P, 6Q