STEAM 8 Curriculum Map

Standards	**Content**	**Skills/Practices**	**Materials/ Resources**	**Assessments (All)** Daily/Weekly/ Benchmarks	**Timeline** (Months/ Weeks/Days)
CSTA K-12 Computer Science Standards (2017) **AP - Algorithms & Programming** 1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate. 2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms. 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. 2-AP-15 Seek and incorporate feedback from team members and users to refine a solution that meets user needs. 1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation and review stages of program development.	Problem Solving and Computing The Problem Solving and Computing unit is a highly interactive and collaborative introduction to the field of computer science, as framed within the broader pursuit of solving problems. Through a series of puzzles, challenges, and real world scenarios, students are introduced to a problem solving process that they will return to repeatedly throughout the course. Students then learn how computers input, output, store, and process information to help humans solve problems within the context of apps. The unit concludes with students designing an app that helps solve a problem of their choosing. **Guiding Questions** Chapter 1 - The Problem Solving Process • What strategies and processes can I use to become a more effective problem	By the end of the unit, students should be able to identify the defined characteristics of a computer and how it is used to solve information problems. They should be able to use a structured problem solving process to address problems and design solutions that use computing technology. The unit also serves to build a collaborative classroom environment where students view computer science as relevant, fun, and empowering.	Chromebook Additional Resources can be found at the Problem Solving and Computing Resources site	**Daily:** • Activity Guides • Wrap Up Questions **Project** Lesson 8: Project - Propose an App To conclude their study of the problem solving process and the input/output/store/process model of a computer, students will propose an app designed to solve a problem of their choosing. To learn more about this project, check out the description in this unit's lesson progression.	3 Weeks

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. **CS - Computing Systems** 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. **IC - Impacts of Computing** 2-IC-20 Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.	solver? <u>Chapter 2 - Computers and Problem Solving</u> ● How do computers help people to solve problems? ● How do people and computers approach problems differently? ● What does a computer need from people in order to solve problems effectively?				
IC - Impacts of Computing 2-IC-21 Discuss issues of bias and accessibility in the design of existing technologies. **AP - Algorithms & Programming** 2-AP-10 Use flowcharts and/or pseudocode to	<u>Interactive Animations and Games</u> In the Interactive Animations and Games unit, students build on their coding experience as they create programmatic images, animations, interactive art, and games. Starting off with simple, primitive shapes and building up to more	By the end of the unit, students should be able to create an interactive animation or game that includes basic programming concepts such as control structures, variables, user input, and randomness. They should manage this	Chromebook Additional Resources can be found at the <u>Interactive Animations and Games Resources</u>	**Daily:** ● Activity Guides ● Wrap Up Questions **Project** Lesson 17: Project - Interactive Card: students plan for and develop an	7 Weeks

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-14 - Create procedures with parameters to organize code and make it easier to reuse. 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	sophisticated sprite-based games, students become familiar with the programming concepts and the design process computer scientists use daily. They then learn how these simpler constructs can be combined to create more complex programs. In the final project, students develop a personalized, interactive program. Along the way, they practice design, testing, and iteration, as they come to see that failure and debugging are an expected and valuable part of the programming process. **Guiding Questions** <u>Chapter 1 - Images and Animations</u> ● What is a computer program? ● What are the core features of most programming languages? ● How does programming enable creativity and individual expression? ● What practices and strategies will help me as I write programs? <u>Chapter 2 - Building Games</u>	task by working with others to break it down using objects (sprites) and functions. Throughout the process, they should give and respond constructively to peer feedback, and work with their teammates to complete a project. Students should leave this unit viewing themselves as computer programmers, and see programming as a fun and creative form of expression.		interactive greeting card using all of the programming techniques they've learned to this point. **Project** Lesson 27: Project - Design a Game: Students plan and build original games using the project guide from the previous two lessons. Working individually or in pairs, they plan, develop, and give feedback on the games. After incorporating the peer feedback, students share out their completed games.	

developing computational artifacts. 2-AP-19 Document programs in order to make them easier to follow, test, and debug.	• How do software developers manage complexity and scale? • How can programs be organized so that common problems only need to be solved once? • How can I build on previous solutions to create even more complex behavior?				