

Prenda Curriculum and CSTA Standards

Prenda Course Units	CSTA Standard
<p>Javascript</p> <p>Learn sequences, variables, conditionals, loops, functions, and objects as you create a tool to help people do their chores.</p>	<p>1B-AP-09 Create programs that use variables to store and modify data.</p>
	<p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p>
	<p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p>
	<p>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.</p>
	<p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.</p>
	<p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p>
	<p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p>
	<p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>
	<p>1B-IC-20 Seek diverse perspectives for the purpose of improving computational artifacts.</p>
	<p>1B-IC-21 Use public domain or creative commons media, and refrain from copying or using material created by others without permission.</p>
	<p>2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.</p>
	<p>2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.</p>
	<p>2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.</p>
	<p>2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.</p>
	<p>2-AP-15 Seek and incorporate feedback from team members and users to refine a solution that meets user needs.</p>
	<p>2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p>

	<p>2-AP-17 Systematically test and refine programs using a range of test cases.</p> <p>2-AP-19 Document programs in order to make them easier to follow, test, and debug.</p> <p>3A-AP-14 Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.</p> <p>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.</p> <p>3A-AP-21 Evaluate and refine computational artifacts to make them more usable and accessible.</p>
<p>Python</p> <p>Learn sequences, variables, conditionals, loops, functions, and objects as you create a hangman game.</p>	<p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>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.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p>1B-IC-20 Seek diverse perspectives for the purpose of improving computational artifacts.</p> <p>1B-IC-21 Use public domain or creative commons media, and refrain from copying or using material created by others without permission.</p> <p>2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.</p> <p>2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.</p> <p>2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.</p>

	<p>2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.</p> <p>2-AP-15 Seek and incorporate feedback from team members and users to refine a solution that meets user needs.</p> <p>2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p> <p>2-AP-17 Systematically test and refine programs using a range of test cases.</p> <p>2-AP-19 Document programs in order to make them easier to follow, test, and debug.</p> <p>3A-AP-14 Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.</p> <p>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.</p> <p>3A-AP-21 Evaluate and refine computational artifacts to make them more usable and accessible.</p>
<p>Scratch</p> <p>Learn sequences, variables, loops, conditionals and events as you create a platformer game. Also practice remixing existing projects.</p>	<p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>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.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p>1B-IC-20 Seek diverse perspectives for the purpose of improving computational artifacts.</p> <p>1B-IC-21 Use public domain or creative commons media, and refrain from copying or using material created by others without permission.</p>

	<p>2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.</p> <p>2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.</p> <p>2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.</p> <p>2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.</p> <p>2-AP-15 Seek and incorporate feedback from team members and users to refine a solution that meets user needs.</p> <p>2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p> <p>2-AP-17 Systematically test and refine programs using a range of test cases.</p> <p>2-AP-19 Document programs in order to make them easier to follow, test, and debug.</p> <p>3A-AP-14 Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.</p> <p>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.</p> <p>3A-AP-21 Evaluate and refine computational artifacts to make them more usable and accessible.</p>
<p>Sphero</p> <p>Learn basic programming concepts like loops, conditionals and functions. Also learn how hardware interacts with software.</p>	<p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>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.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.</p>

	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.
<p>Ozobot</p> <p>Learn basic programming concepts like loops, conditionals and functions. Also learn how hardware interacts with software.</p>	1B-AP-09 Create programs that use variables to store and modify data.
	1B-AP-10 Create programs that include sequences, events, loops, and conditionals.
	1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.
	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.
	1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.
	1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.
	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-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.
<p>JS Web</p> <p>Learn how to access the contents of a web page with Javascript through the DOM API. Create interactive events and animations as you create a website project.</p>	1B-AP-09 Create programs that use variables to store and modify data.
	1B-AP-10 Create programs that include sequences, events, loops, and conditionals.
	1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.
	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.
	1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.

	<p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p>1B-IC-20 Seek diverse perspectives for the purpose of improving computational artifacts.</p> <p>1B-IC-21 Use public domain or creative commons media, and refrain from copying or using material created by others without permission.</p> <p>2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.</p> <p>2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.</p> <p>2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.</p> <p>2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.</p> <p>2-AP-15 Seek and incorporate feedback from team members and users to refine a solution that meets user needs.</p> <p>2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p> <p>2-AP-17 Systematically test and refine programs using a range of test cases.</p> <p>2-AP-19 Document programs in order to make them easier to follow, test, and debug.</p> <p>3A-AP-14 Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.</p> <p>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.</p> <p>3A-AP-21 Evaluate and refine computational artifacts to make them more usable and accessible.</p> <p>3B-AP-15 Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.</p>
<p>Animation</p> <p>Learn basic web animations using Javascript as you create a marketing website.</p>	<p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p>

	<p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p>
	<p>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.</p>
	<p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.</p>
	<p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p>
	<p>2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.</p>
	<p>2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.</p>
	<p>2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.</p>
	<p>2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.</p>
	<p>2-AP-17 Systematically test and refine programs using a range of test cases.</p>
	<p>2-AP-19 Document programs in order to make them easier to follow, test, and debug.</p>
	<p>3A-AP-14 Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.</p>
	<p>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.</p>
	<p>3B-AP-15 Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.</p>
<p>Hero Scratch</p> <p>Use variables, loops, functions, conditionals, and events as you create three different games in Scratch.</p>	<p>1B-AP-09 Create programs that use variables to store and modify data.</p>
	<p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p>
	<p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p>
	<p>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.</p>
	<p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.</p>

	<p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p>1B-IC-20 Seek diverse perspectives for the purpose of improving computational artifacts.</p> <p>1B-IC-21 Use public domain or creative commons media, and refrain from copying or using material created by others without permission.</p> <p>2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.</p> <p>2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.</p> <p>2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.</p> <p>2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.</p> <p>2-AP-15 Seek and incorporate feedback from team members and users to refine a solution that meets user needs.</p> <p>2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p> <p>2-AP-17 Systematically test and refine programs using a range of test cases.</p> <p>2-AP-19 Document programs in order to make them easier to follow, test, and debug.</p> <p>3A-AP-14 Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.</p> <p>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.</p> <p>3A-AP-21 Evaluate and refine computational artifacts to make them more usable and accessible.</p> <p>3B-AP-09 Implement an artificial intelligence algorithm to play a game against a human opponent or solve a problem.</p>
<p>Drawing</p> <p>Learn how to use the HTML Canvas to draw on a webpage with Javascript, and animate</p>	<p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p>

<p>your drawings as you create an artistic webpage.</p>	<p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>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.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.</p> <p>2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.</p> <p>2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.</p> <p>2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.</p> <p>2-AP-17 Systematically test and refine programs using a range of test cases.</p> <p>2-AP-19 Document programs in order to make them easier to follow, test, and debug.</p> <p>3A-AP-14 Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.</p> <p>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.</p> <p>3B-AP-15 Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.</p>
<p>JS Games</p> <p>Use variables, loops, functions, conditionals, methods, and debugging skills as you build four classic arcade games.</p>	<p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>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.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.</p>

	<p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p>1B-IC-20 Seek diverse perspectives for the purpose of improving computational artifacts.</p> <p>1B-IC-21 Use public domain or creative commons media, and refrain from copying or using material created by others without permission.</p> <p>2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.</p> <p>2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.</p> <p>2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.</p> <p>2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.</p> <p>2-AP-15 Seek and incorporate feedback from team members and users to refine a solution that meets user needs.</p> <p>2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p> <p>2-AP-17 Systematically test and refine programs using a range of test cases.</p> <p>2-AP-19 Document programs in order to make them easier to follow, test, and debug.</p> <p>3A-AP-14 Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.</p> <p>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.</p> <p>3A-AP-21 Evaluate and refine computational artifacts to make them more usable and accessible.</p> <p>3B-AP-09 Implement an artificial intelligence algorithm to play a game against a human opponent or solve a problem.</p>
<p>SQL</p> <p>Learn about databases, create and edit tables, and run queries as you gain the skills</p>	<p>1B-DA-06 Organize and present collected data visually to highlight relationships and support a claim.</p> <p>1B-DA-07 Use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate an idea.</p>

<p>needed to isolate a single piece of data from a large data set.</p>	<p>2-DA-07 Represent data using multiple encoding schemes.</p> <p>2-DA-08 Collect data using computational tools and transform the data to make it more useful and reliable.</p> <p>2-DA-09 Refine computational models based on the data they have generated.</p>
<p>Makey Makey</p> <p>Learn about computer systems and how hardware interacts with software. Use basic programming skills, like sequences and events, to program the Makey Makey.</p>	<p>1B-CS-02 Model how computer hardware and software work together as a system to accomplish tasks.</p> <p>1B-CS-03 Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies.</p> <p>2-CS-02 Design projects that combine hardware and software components to collect and exchange data.</p> <p>2-CS-03 Systematically identifies and fixes problems with computing devices and their components.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-IC-21 Use public domain or creative commons media, and refrain from copying or using material created by others without permission.</p>
<p>Raspberry Pi</p> <p>Learn about computer systems and how hardware interacts with software through sensors, cameras, and more.</p>	<p>1B-CS-02 Model how computer hardware and software work together as a system to accomplish tasks.</p> <p>1B-CS-03 Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies.</p> <p>2-CS-02 Design projects that combine hardware and software components to collect and exchange data.</p> <p>2-CS-03 Systematically identify and fix problems with computing devices and their components.</p> <p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>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.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p>

	<p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p>1B-IC-20 Seek diverse perspectives for the purpose of improving computational artifacts.</p> <p>1B-IC-21 Use public domain or creative commons media, and refrain from copying or using material created by others without permission.</p> <p>2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.</p> <p>2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.</p> <p>2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.</p> <p>2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.</p> <p>2-AP-15 Seek and incorporate feedback from team members and users to refine a solution that meets user needs.</p> <p>2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p> <p>2-AP-17 Systematically test and refine programs using a range of test cases.</p> <p>2-AP-19 Document programs in order to make them easier to follow, test, and debug.</p> <p>3A-AP-14 Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.</p> <p>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.</p> <p>3A-AP-21 Evaluate and refine computational artifacts to make them more usable and accessible.</p>
<p>HTML</p> <p>Learn basic HTML syntax and the most popular tags as you create an online job posting.</p>	<p>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.</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-IC-21 Use public domain or creative commons media, and refrain from copying or using material created by others without permission.</p>

	<p>2-AP-15 Seek and incorporate feedback from team members and users to refine a solution that meets user needs.</p>
	<p>2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p>
<p>CSS</p> <p>Learn the basic syntax of CSS and common properties as well as advanced concepts such as positions, grid, and flexbox as you create a library digital display.</p>	<p>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.</p>
	<p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p>
	<p>1B-IC-21 Use public domain or creative commons media, and refrain from copying or using material created by others without permission.</p>
	<p>2-AP-15 Seek and incorporate feedback from team members and users to refine a solution that meets user needs.</p>
	<p>2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p>
<p>Design</p> <p>Learn basic design principles such as a design process, and specific web design principles as you create a website for a fictional town.</p>	<p>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.</p>
	<p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p>
	<p>1B-IC-21 Use public domain or creative commons media, and refrain from copying or using material created by others without permission.</p>
	<p>2-AP-15 Seek and incorporate feedback from team members and users to refine a solution that meets user needs.</p>
	<p>2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p>
<p>Prenda Missions</p> <p>In Prenda, we have over 50 projects for students to work on. These vary by coding discipline (web, games, scripting), and are opportunities for students to use their own creativity to complete the project.</p>	<p>1B-AP-09 Create programs that use variables to store and modify data.</p>
	<p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p>
	<p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p>
	<p>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.</p>
	<p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.</p>
	<p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p>

	<p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p>
	<p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>
	<p>1B-IC-20 Seek diverse perspectives for the purpose of improving computational artifacts.</p>
	<p>1B-IC-21 Use public domain or creative commons media, and refrain from copying or using material created by others without permission.</p>
	<p>2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.</p>
	<p>2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.</p>
	<p>2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.</p>
	<p>2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.</p>
	<p>2-AP-15 Seek and incorporate feedback from team members and users to refine a solution that meets user needs.</p>
	<p>2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p>
	<p>2-AP-17 Systematically test and refine programs using a range of test cases.</p>
	<p>2-AP-19 Document programs in order to make them easier to follow, test, and debug.</p>
	<p>3A-AP-14 Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.</p>
	<p>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.</p>
	<p>3A-AP-21 Evaluate and refine computational artifacts to make them more usable and accessible.</p>
	<p>3B-AP-09 Implement an artificial intelligence algorithm to play a game against a human opponent or solve a problem.</p>
	<p>3A-AP-15 Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.</p>
	<p>3A-AP-19 Systematically design and develop programs for broad audiences by incorporating feedback from users.</p>

3A-AP-21 Evaluate and refine computational artifacts to make them more usable and accessible.