



PENCIL CODE

Pencil Code is an exciting apprenticeship where students learn how to create simple computer programs--via Pencil Code--along with the beginnings of CSS and HTML. Students explore coding from the perspective of an artist, engineer, and designer. Each lesson is workshop-based, where students complete tasks outlined on cards, building understanding along the way of how to use, manipulate, and master various programming commands. Students create outlined projects and design unique innovations to the projects along the way. Each student creates a WOW! Portfolio of his or her work. Each portfolio features a project demonstrating various skills and showcases a final project of the student's design that highlights his/ her skills, interests, and creativity.

Unit Standards and Objectives

Innovation Standard #1: Citizen Schools students will generate an idea or product that suits a practical or artistic purpose

Lesson Objectives:

- SWBAT generate a list of ideas for innovations, categorize the ideas based on feasibility, advantages and limitations, and select the best to complete based on interest.
- SWBAT think of an innovation to one or more of the Pencil Code Cards to fill a specific purpose for the user.

Innovation Standard #2: Citizen Schools students will use a design process to create ideas or products

Lesson Objectives:

- SWBAT use the programming design process to create a unique scene (underwater or city).
- SWBAT test and revise their programming language to make at least two Pencil Code Cards with simple coding errors function properly.
- SWBAT become familiar with three ways professionals use programming in their careers.

Innovation Standard #3: Citizen Schools students will realize a product or idea that suits a given purpose

Lesson Objectives

- SWBAT describe the WOW! Portfolio each student will create.
- SWBAT design and create a program that combines concepts from two or more weeks (i.e., loops and buttons or function and images).
- SWBAT complete a WOW! Portfolio web page with the purpose of showing his/ her work to an audience.
- SWBAT design a final project, the purpose of which is to showcase their innovation skills (to be included in their WOW! Portfolio).
- SWBAT complete an innovative final project to showcase their skills within their WOW! Portfolio.
- SWBAT complete a WOW! Portfolio, including a showcase final project, along with four projects from the semester featuring specific skills (basic commands, buttons/ events, functions, and animation/images).
- SWBAT celebrate all that they have learned and accomplished.

Content Standard #4: Implement problem solutions using a programming language, including: looping behavior, conditional statements, logic, expressions, variables, and functions.

Lesson Objectives

- SWBAT use Pencil Code to create a program that draws or moves the turtle around the screen using at least five different basic commands.
- SWBAT to use loops and indented code to create drawings that use repetition.
- SWBAT use and switch between text code and block code language.
- SWBAT use Pencil Code to create at least one button, key down, or click to trigger actions including sounds, movement making extra turtles appear.
- SWBAT use Pencil Code to create one-three unique functions in his/ her scene.
- SWBAT find and correct "bugs" on at least two Pencil Code Cards.
- SWBAT use Pencil Code to create multiple sprites and animate Google images.
- SWBAT use HTML to add a header and link to at least two of his/ her WOW! Portfolio projects.
- SWBAT use CSS to add at least two of the following to his/ her WOW! Portfolio sites: special font, background, hover and/ or iFrame borders.
- SWBAT use HTML to link all four of their WOW! Portfolio projects to his/ her WOW! Portfolio site.
- SWBAT begin a final project for his/ her WOW! Portfolio that showcases his/ her programming skills, including at least one of the following: looping behavior, conditional statements, logic, expressions, variables, and/ or functions.



Essential Questions

- Do you need to be creative to be good at computer programming?
- How can I use computer coding to design new ideas/ products?

Performance Task Assessment (WOW!)

For the Pencil Code Performance Task, each student will create a portfolio of his/ her work by both selecting a piece of work from each week of the unit to showcase his/ her growth and creating a final piece of work demonstrating a combination of skills learned.

Goal: Students will create a portfolio of their work that showcases growth and demonstrates computer programming skills in Pencil Code (Coffeescript), HTML, and CSS in order to apply for a design/ programming job at a tech company.

Role: Students are applicants applying for a programming position at a tech company.

Audience: The audience for the task are artists, engineers, and designers from a local programming firm along with family, friends, and teachers from the students' community who take on the persona of design principals at a tech company deciding whether to hire the applicant (student) as a designer to work at their firm.

Situation: Typically, when programmers apply for coding positions as artists, engineers, or designers, they are asked to submit a portfolio of exemplary work. The Citizen Schools students' WOW! Portfolios reflect such a portfolio.

Product: Students will create a portfolio of their work. The portfolio will be a website that includes HTML and CSS programming of headers, design, etc. and links to four projects that demonstrate general skills from the apprenticeship and one project that the student created as a feature showcase.

Standards: Students' portfolios will be evaluated using the Citizen Schools 21st Century Skill [Innovation Rubric](#). To create their final project for their WOW! Portfolio, students will use the process of innovation to first brainstorm ideas, then categorize and select the best idea based on the advantages, limitations and feasibility of the idea, follow the Programming Design Process, and finally complete the innovation successfully. Additionally, portfolios will be evaluated against the computer programming skills (basic movement/ drawing, loops, buttons/ events, functions, debugging, conditionals, animation, images, headers in HTML, formatting in CSS) taught throughout the apprenticeship to ensure students have mastered each of the coding techniques.

Lesson Plans At-A-Glance

Lesson Plans are available [here](#)

Week	Lesson Objectives	Agenda	Outcomes & Work Products
1	<ul style="list-style-type: none"> ● SWBAT explain expectations for behavior, computer use, and professionalism in this apprenticeship. ● SWBAT describe the WOW! Portfolio each student will create. ● SWBAT use Pencil Code to create a program that draws or moves the turtle around the screen using at least five different basic commands. 	<ul style="list-style-type: none"> ● Hook: Welcome to Pencil Code! ● Introduction to New Material: What is Computer Programming? ● Activity 1: Expectations ● Activity 2: Human Pencil Code ● Activity 3: Drawing & Cards ● Assessment: Work Product 	<ul style="list-style-type: none"> ● Basic Pencil Code commands ● Knowledge of what their WOW! Portfolios will be
2	<ul style="list-style-type: none"> ● SWBAT generate a list of ideas for innovations, categorize the ideas based on feasibility, advantages and limitations, and select the best to complete based on interest. 	<ul style="list-style-type: none"> ● Hook: Feature Project ● Introduction to New Material: Loops & Text Code ● Activity 1: Cards ● Activity 2: Game OR Share Projects ● Activity 3: Innovations 	<ul style="list-style-type: none"> ● Loops using the "For" command and indented code ● How to use both block and text code ● Stage 1 of the innovation process: Generating Ideas ● Create Pjt. #1 for their WOW! Portfolio



Core Apprenticeship Library
Apprenticeship Sector: Coding & Gaming
Unit Guide: Pencil Code



	<ul style="list-style-type: none"> ● SWBAT use loops and indented code to create drawings that use repetition. ● SWBAT use and switch between text code and block code language. 	<ul style="list-style-type: none"> ● Assessment: Work Product 	
3	<ul style="list-style-type: none"> ● SWBAT use Pencil Code to create at least one button, key down, or click to trigger actions including sounds, or movement making extra turtles appear. ● SWBAT think of an innovation to one or more of the Pencil Code Cards to fill a specific purpose for the user. 	<ul style="list-style-type: none"> ● Hook: Feature Project ● Introduction to New Material: Buttons, Events and Sound ● Activity 1: Cards ● Activity 2: Game OR Share Projects ● Activity 3: Innovations ● Assessment: Work Product 	<ul style="list-style-type: none"> ● How to code for buttons, key downs, and clicks ● How to add sound, movement, and extra turtles ● Innovation for purpose ● Create Pjt. #2 for their WOW! Portfolio
4	<ul style="list-style-type: none"> ● SWBAT use the programming design process to create a unique scene (underwater or city). ● SWBAT use Pencil Code to create one-three unique functions in his/ her scene. 	<ul style="list-style-type: none"> ● Hook: Feature Project ● Introduction to New Material: Functions ● Activity 1: Cards ● Activity 2: Game OR Share Projects ● Activity 3: Innovations ● Assessment: Work Product 	<ul style="list-style-type: none"> ● How to create functions ● Stage 2 of the innovation process: Programming Design Process ● Create Pjt. #3 for their WOW! Portfolio
5	<ul style="list-style-type: none"> ● SWBAT test and revise their programming language to make at least two Pencil Code Cards with simple coding errors function properly. ● SWBAT become familiar with three ways professionals use programming in their careers. ● SWBAT find and correct “bugs” on at least two Pencil Code Cards. 	<ul style="list-style-type: none"> ● Hook: Feature Project ● Introduction to New Material: Debugging ● Activity 1: Cards ● Activity 2: Game OR Share Projects ● Activity 3: Innovations ● Assessment: Work Product 	<ul style="list-style-type: none"> ● How to catch and fix bugs ● Q&A with computer programming professionals
6	<ul style="list-style-type: none"> ● SWBAT design and create a program that combines concepts from two or more weeks (i.e., loops and buttons or function and images). ● SWBAT use Pencil Code to create multiple sprites and animate Google images. 	<ul style="list-style-type: none"> ● Hook: Feature Project ● Introduction to New Material: Animation & Images ● Activity 1: Cards ● Activity 2: Game OR Share Projects ● Activity 3: Look Back at Innovations ● Assessment: Work Product 	<ul style="list-style-type: none"> ● How to create sprites and animate images ● Stage 3 of the innovation process: Creating the idea ● Create Pjt. #4 for their WOW! Portfolio
7	<ul style="list-style-type: none"> ● SWBAT complete ~35% of his/ her WOW! Portfolio web page with the purpose of showing his/ her work to an audience. ● SWBAT use HTML to add a header and link to at least two of his/ her WOW! Portfolio projects. 	<ul style="list-style-type: none"> ● Hook: Feature Project ● Introduction to New Material: Computer Languages ● Activity 1: Cards ● Activity 2: Game OR Share Projects ● Introduction to New Material: WOW! Portfolios ● Activity 3: WOW! Portfolio Work Time ● Assessment: Work Product 	<ul style="list-style-type: none"> ● How to use HTML to add content to a webpage ● Begin creating the websites to house their WOW! Portfolios
8	<ul style="list-style-type: none"> ● SWBAT complete ~70% of his/ her WOW! Portfolio web page with the purpose of showing his/ her work to an audience. ● SWBAT use CSS to add at least two of the following to his/ her WOW! Portfolio sites: special font, background, hover and/ or iFrame borders. ● SWBAT use HTML to link all four of their WOW! Portfolio projects to his/ her WOW! Portfolio site. 	<ul style="list-style-type: none"> ● Hook: Feature Project ● Introduction to New Material: CSS ● Activity 1: Cards ● Activity 2: Share Projects ● Activity 3: WOW! Portfolio Work Time ● Assessment: Work Product 	<ul style="list-style-type: none"> ● How to use CSS to add style to a webpage ● Finish linking all four initial projects to their WOW! Portfolios
9	<ul style="list-style-type: none"> ● SWBAT design a final project, the purpose of which is to showcase their innovation skills (to be included in their WOW! Portfolio). ● SWBAT begin a final project for his/ her WOW! Portfolio that showcases his/ her programming skills including at least one of the following: looping behavior, conditional statements, logic, expressions, variables, and/ or functions. 	<ul style="list-style-type: none"> ● Hook: Feature Project ● Introduction to New Material: WOW! Portfolio Final Project ● Activity 1: Final Project Work Time #1 ● Activity 2: Share Projects ● Activity 3: Final Project Work Time #2 ● Assessment: Work Product 	<ul style="list-style-type: none"> ● Begin their final project--a showcase piece for their WOW! Portfolios



Core Apprenticeship Library
Apprenticeship Sector: Coding & Gaming
Unit Guide: Pencil Code



10	<ul style="list-style-type: none"> ● SWBAT complete an innovative final project to showcase their skills within their WOW! Portfolio. ● SWBAT complete a WOW! Portfolio, including a showcase final project along with four projects from the semester featuring specific skills (basic commands, buttons/ events, functions, and animation/ images). ● SWBAT celebrate all that they have learned and accomplished. 	<ul style="list-style-type: none"> ● Hook: Feature Project ● Introduction to New Material: WOW! Portfolios ● Activity 1: Final Project Work Time ● Activity 2: Share Projects ● Activity 3: Celebration ● Assessment: Wrap Up 	<ul style="list-style-type: none"> ● Complete the final project for the WOW! Portfolios ● Finalize the WOW! Portfolios
----	---	---	--

Lesson Elements	
Hook <small>Opening ritual used each week to build excitement</small>	Each week, plan to feature an exemplary student project from the week before. Use this as an opportunity to highlight each student's work. Ideally, feature a different student's work each week. Select a student whose work from the previous week might further push other students' innovations, thinking, and creativity.
Assessment <small>How you will measure student learning (i.e., exit tickets, student writing, student presentations, etc.)</small>	The weekly assessment for every week Lessons 1-9 is a review of the student's programming work. In weeks 1-6 students will select a program to save as his/ her assessment. To accompany the saved work product, students will also complete a short written component within the Student Guide. Additionally, on several weeks, collect the Innovation Handouts to further measure student learning and mastery.
Structures <small>Learning structures, tools, or student grouping strategies</small>	These lessons are all workshop-based. Students receive a short mini-lesson on the programming topics of the day and then spend a majority of the class programming and innovating on programs by completing and modifying Pencil Code Cards.
Procedures <small>Special procedures used each class (i.e. handing out folders, rearranging seating, etc.)</small>	<p>It is strongly recommended that you create clear procedures for many of the repeated tasks in your classroom. Lesson 1 includes a graphic organizer to help you plan for the top recommended procedures for this apprenticeship, namely: assigned seats, get attention, student folders, file naming and saving, getting a new card, wrapping up, assessments, and shutting down. See Lesson 1 for more information and resources and save time for CTs and TL to plan these collaboratively.</p> <p>Plan to reteach and practice all procedures you taught in Week 1 in each lesson thereafter. Students will naturally forget from week to week, not because they are maliciously misbehaving, but simply because a lot happens from week to week and they easily forget these details unless they are matter-of-factly reinforced.</p>

Implementation Notes	
Supplies <small>Materials, tools, technology</small>	<ul style="list-style-type: none"> ● Folders (one per student): Recommended to use folders with brads so that you could put the student guide in the brads as a "book" in the folder. ● Computers (one per student), with internet access ● Teacher computer connected to a projector (1 per apprenticeship) ● File box or binder with sheet protectors to organize Pencil Code Cards, week by week ● Paper and copies for the handouts ● Thank you for the guest speakers in Week 5
Budget	<ul style="list-style-type: none"> ● Folders - ~\$7 on Amazon for 25 folders ● File Box - ~\$10 on Amazon for open top portable file storage
	<p>The computers are is the most important material of this apprenticeship. Plan to hold the entire apprenticeship in the lab or have access to a laptop cart each session.</p> <p>Throughout the apprenticeship, students will complete and innovate off of Pencil Code Cards.</p> <p>Samples of all of the Pencil Code Cards are available at citizenschools.pencilcode.net. Username: citizenschools, Password: Citizen. Each card is saved in accordance with the file-saving procedure</p>



Core Apprenticeship Library
Apprenticeship Sector: Coding & Gaming
Unit Guide: Pencil Code



<p>Supporting Materials & Resources</p> <p>handouts, books, materials</p>	<p>taught in this apprenticeship (week#/cardname).</p> <p>Although a teacher view of the students' Pencil Code accounts is not available, it is highly recommended that the teacher create a page that links to each student's account. That way, sharing programs during Activity 2 of Lessons 1-9 will be easy. See a sample mock up here. Additionally, you might want to set up a page to link each feature project for the Hook and each WOW! Portfolio after Lesson 7. This will make it easier to review progress towards completion on each portfolio and share the WOW! Portfolios.</p> <p>Student Guides: a copy of the full guide per student should be printed and put in student folders in preparation for Lesson 1:</p> <ul style="list-style-type: none"> ● In it, students will fill out information about their daily work product assessment. ● The students' actual work products for assessment will be saved in their Pencil Code files and the teacher can access them from the site they create that links to each student's page and files. Then, once in there, the teacher will just click and open the assessment for that week. <p>Pencil Code Cards: a copy of each card per student should be printed and prepared for each lesson. Cards provide students with guided instruction to complete coding activities aligned to topics of the week. It is recommended that you use a file box or a binder with sheet protectors to store cards from previous weeks. See Lesson 1 for full details on organization and procedures with respect to the Pencil Code Cards.</p>
<p>Location</p> <p>Tables/ desks, or classroom, gym, kitchen, outside, etc.</p>	<p>Each lesson uses a class set of computers in a normal classroom at desks/ tables or a computer lab.</p> <p>Additionally:</p> <ul style="list-style-type: none"> ● In Lesson 1, some outdoor space is needed for one activity. ● In Lesson 10, you may want to hold a small celebration of all that was accomplished in the apprenticeship. If you decide to have snacks available during that time, then you may need to do this in a room without computers. Check with your school to find out what the campus policies are for food and computers.
<p>Choice and Voice</p> <p>Key decisions students make</p>	<p>Students will independently decide how to decorate and personalize their WOW! Portfolios using HTML and CSS code. Additionally, students will independently design their final projects. Their choices will determine their final products.</p>
<p>Modifications for Student Needs</p> <p>Supports and changes to help meet the needs of all learners</p>	<p>Because this apprenticeship is workshop-based, students work at their own pace and learn as they complete Pencil Code Cards. One student might complete one card and another four. Thus some level of modification is built-in naturally. Additionally, consider limiting the number of blanks on the Cards for students that would benefit from that modification.</p> <p>For the WOW! Portfolios, consider modifying the expectations for the student's specific portfolio or final project as needed. Review Lesson 7 for expectations for the four projects that students will include in their portfolio from Lessons 1-6, and review Lesson 9 for expectations for the final project. Both of these sets of expectations could be modified as needed without embarrassment or singling out of the students with different needs.</p>
<p>Student Background Knowledge and Skills Needed</p> <p>Academic skills, social emotional skills or developmental milestones needed</p>	<p>Students may struggle if they do not have a solid grasp of the English language. All the Pencil Code Cards are only in English and Pencil Code itself is basically abbreviated English. It may be difficult for an English language learner to memorize simple commands (fd or rt) when they do not automatically think of the root word of what they are trying to tell the computer to do (go forward, right turn). An easy reference guide is included in the Student Guide that may lessen this gap.</p> <p>Additionally, some math is used. For example, locating coordinates, calculating how far to go in one direction, and degrees of an angle to turn. Students do not have to have this background knowledge, but it would be helpful. Notes are included in the relevant lessons in case you need to allocate more time to teaching these skills, and an easy reference guide for angles and shapes is included in the Student Guide.</p> <p>In programming, detail matters. Students who are not attentive to detail may be excessively frustrated by programing. One capitalized letter or a missing comma will make a program not function. Block coding will help but is not always an answer. Teach students to switch back and forth between text and block coding frequently.</p>
<p>College and Career</p>	<p>This apprenticeship connects easily to College and Career Readiness, as it teaches skills that a</p>








Core Apprenticeship Library
Apprenticeship Sector: Coding & Gaming
 Unit Guide: Pencil Code



<p>Readiness</p> <p>Connections to college and career</p>	<p>computer programmer might use. Review these resources to get more ideas of the types of careers and fields that a computer programmer might pursue.</p> <ul style="list-style-type: none"> ● Computing Careers ● Careers Brochure <p>Additionally, in Lesson 5, it is recommended that you coordinate three guest speakers with a background in computer programming for a panel discussion on computer programming as a career pathway. If possible, try to find people who use computer programming in very different ways in their jobs, e.g. an IT specialist, a mobile game designer, and a programmer for healthcare software. Try searching for “your city name, type of software, software” to find companies in your area that specialize in different types of programming. For example, a search for “Durham Healthcare Software,” returns several companies that, if in the Durham area, you could email and see if they’d be interested in sending a representative to speak to your students. If you do not have access to people in your area, consider bringing in one or more of the speakers virtually.</p> <p>Most schools do not offer computer programming classes until high school at the earliest. Review your district’s class schedules to share information about available classes. Also review local classes that might be available at the library or community center. There is time set aside in Lesson 10 to share this information, but students may be interested earlier.</p>
<p>Co-Teaching Roles</p> <p>Recommendations for co-teaching and planning</p>	<p>This apprenticeship needs to have at least two co-teachers but ideally would have one co-teacher for every four-five students. The apprenticeship is taught in workshop style and frequent check-ins with students as they work will increase the effectiveness of the instruction. Specific recommendations are included throughout the lessons to maximize the use of co-teachers.</p>
<p>Special Resources</p> <p>Field trips, excursions, guest speakers</p>	<p>A panel of guest speakers is recommended in Lesson 5. See that lesson, as well as the section on College and Career Readiness above, for more resources.</p> <p>If your apprenticeship schedule has an extra week, it is recommended that you plan a field trip to a local company that specializes in some type of computer programming.</p>
<p>Road Map to WOW!</p> <p>Visual overview for students of their 10 week apprenticeship</p>	<p>Note to CT/TL: Create a poster-sized visual of the information listed below, display and reference weekly in your classroom.</p> <p>Visual overview for students of their 10-week apprenticeship:</p> <ul style="list-style-type: none"> Week 1: Welcome to Pencil Code! Week 2: Loops & Text Code Week 3: Buttons & Events Week 4: Functions Week 5: Debugging Week 6: Animation & Images Week 7: HTML Week 8: CSS Week 9: Final Projects Week 10: WOW! Portfolios WOW!



Co-Teaching Structures Guide

Teaching Model	Description	Why should we use it?	When should we use it?
<p>Parallel Teaching</p> 	<p>Class is split into two (or more) small teams. <u>Same</u> content is taught to each team.</p>	<ul style="list-style-type: none"> ·Low student-teacher ratio ·Greater proximity to high-risk students ·Co-teachers have equal presence and responsibility in the classroom 	<ul style="list-style-type: none"> ·When we can plan effectively together to ensure we teach the same content to each group well. ·Classroom's physical structure permits it. ·For lessons with heavy independent work ·Need to provide a lot of individual attention
<p>Station Teaching</p> 	<p>Class is split into two (or more) small teams. <u>Different</u> material taught to each group simultaneously and then teams switch or teachers switch.</p>	<ul style="list-style-type: none"> ·Low student-teacher ratio ·Co-teachers have equal presence and responsibility in the classroom. ·More variety in teaching methods for teachers and students 	<ul style="list-style-type: none"> ·When a lesson can be split into two mutually exclusive and equally timed parts (e.g. using a camera/critiquing a photo, chopping vegetables/measuring ingredients) ·When the classroom's physical structure permits it ·For lessons with a lot of knowledge or skill-building
<p>Team Teaching</p> 	<p>Both teachers actively teach the material taking turns during the lesson to lead teach. While one teacher is lead teaching the other goes around to groups or individual students.</p>	<ul style="list-style-type: none"> ·One teacher can pay attention to high-risk students while one teacher leads the full class. ·Co-teachers have equal presence and responsibility in the classroom. 	<ul style="list-style-type: none"> ·When it's difficult to effectively split a lesson into two stations ·When a lesson has lectures and independent practice time ·If most SPED students can follow whole-group instruction ·Best used with well-developed co-teaching relationship ·For lessons with a lot of group work
<p>Alternative Teaching</p> 	<p>One teacher remediates a small group of students (pre-teach, re-teach, supplement, or enrich) and catches them up for the main lesson being taught by the other teacher.</p>	<ul style="list-style-type: none"> ·Low student-teacher ratio. ·To remediate in class for a small group of students. ·To catch students up who may not have understood/missed previous lessons 	<ul style="list-style-type: none"> ·When the benefits from a few minutes of remediation/ pre-teaching will pre-empt greater misunderstandings for the lesson. ·When the classroom's physical structure permits small group in one part of the room. (CTs should not be left alone in the classroom with students.)
<p>One Teach, One Assist</p> 	<p>One teacher lead teaches the whole lesson and the other teacher works with individual students.</p>	<ul style="list-style-type: none"> ·To redirect behavior from an especially low functioning student. ·To pay greater attention to a student who needs one-on-one interaction in order to keep up 	<ul style="list-style-type: none"> ·If there is a particularly high-needs student(s) in the classroom that need specific support. ·During direct-teach sections of the lesson



The Pitch

Preparation beforehand:

- Ensure you can show the video (projector, computer, speakers, etc.)
- Cue up the video: [Anybody Can Learn](#) by code.org
- Cue up the Pencil Code program you are going to use. The one described in the pitch can be found here: [ApprenticeshipFairPitchTurtleRace](#) (username: citizenschools, password: Citizen) Or you can create a more exciting one!
- Print and prepare images that you would like to display from the [code.org "inspire" page](#)--see step #3 for the posters. Pick two-four that you think will be most relevant to your students, if you have space to display them during your pitch.

Make your pitch:

- **Introduce yourself and say:** Computers are everywhere. Careers in computers are also everywhere, and increasingly so as your generation grows up and joins the workforce. And what's even better? Anyone can learn how to code: you just have to take the right classes and practice! This apprenticeship is an introduction to computer programming.
 - **Show:** a blank screen in Pencil Code.
 - **Say:** See the turtle? I can use code to program it to do almost anything I want! What should I tell the turtle to do?
 - **Take student answers.** Encourage silly and serious suggestions. Write the suggestions on a chart paper.
 - **Demonstrate** moving forward, backward, and making a nesting loop.
 - **Say:** One thing that you'll be able to do in this apprenticeship is create simple games like this one.
 - **Play program:** [Apprenticeship Fair Pitch Turtle Race](#)
 - **Say:** Look at the code compared to the visual. That's what you'll learn how to write (point at the code), so that you can create that (point at the turtles) in this apprenticeship.
- **Say:** Besides it being cool, creative, and fun, there are lots of good reasons to want to learn how to program. Let's hear from some of the experts in this video.
- **Play the video:** [Anybody Can Learn](#) by code.org (1 minute).
 - **Say:** It's amazing, isn't it? We all use computers all the time, but very, few people know how to code in a sophisticated way.
 - **Say:** I'm not a billionaire like the people featured in this video, but I love computer programming. For me it's an outlet to be creative and bring my ideas to life. It also pays me a great salary to support my family and live comfortably.
 - **Share your story.** Sample: I took my first computer programming class when I was in college. I was fascinated with computers and how they work. I wanted to program games and did for a while.
- **Say:** Think about it. Have you ever been to the doctor and seen the special equipment they use to diagnose and treat illnesses? In the future, you could be the computer programmer behind it. Do you use social media? What if in the future you could work on the next Facebook, Snapchat, or Twitter? Have you ever thought it would be cool to design video games? In the future, you could be programming the next Angry Birds.
- **Say:** At the root of everything we do on our smartphones, our tablets, and our computers is a creative computer programmer. See if computer programming is right for you for a career by taking this apprenticeship!
- **Say:** In this apprenticeship, we'll create all sorts of computer programs AND you'll innovate off of these programs to create your own ideas.

Materials Needed for Pitch Day

1. Computer, projector, and speakers to show the video and programs
2. Pencil Code program to show. You can use the one described in the Pitch ([found here](#)) or you can create your own.
3. Chart paper or white board and markers
4. Images that you'd like to display from the [code.org "inspire" page](#)--see step #3 for the posters



Apprenticeship in Action

Citizen Teachers and Teaching Fellows who have taught this apprenticeship had the following notes, stories and suggestions to share:

"One of my favorite memories is from the first class when one of the kids did something he thought was a bug, he said 'I think that's a Gllitch!' And then when we told him, that's the way the program is written, it's supposed to work that way, he said, 'I made a Gllitch!' and then the funny way he said glitch became a running joke in the class, both for describing bugs and for surprising fireworks. In a way, it was our class's way of celebrating accidental discoveries rather than fearing mistakes.

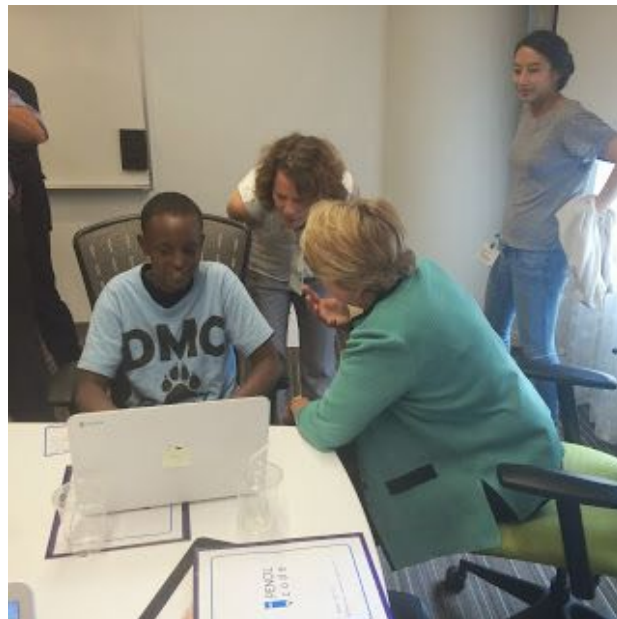
Another favorite moment was when one of our students was hooking up the "remote control" program (with keyboard control to move a sprite) to move the image of a smartphone on the screen. But after it went off the screen, it was hard to navigate it back on again. She decided to add an 'x' button to make it go back home again, and when it worked, she was amazed at herself, beaming for the rest of the session (and over the weeks). Even for small things, when you realize you can put the same pieces together to make something new, it's really empowering."

- David Bau, Pencil Code developer and Citizen Teacher of Pencil Code since 2011:

A Tip for the Pitch: "In your pitch, show students what coding is about. Show both the blocks and the text code. Coding can be hard, but you can do hard things!"

"Last fall, I worked with a student *Xiomara* in her Google Apprenticeship, "Pencil Code." Each week she brought a positive, can-do attitude to class, and constantly challenged herself to grow in her computer coding. During the Google WOW! she really shined as an expert computer coder, and proudly described our class website to anyone who was stopping by our station. During our reflection week, she wrote that her favorite part of the Google WOW! was that she felt like she was in a movie, "trying to get a job at the company." She truly is a real super star."

-Teaching Fellow at Edwards Middle School, Boston, MA 2016



CEO of AmeriCorps, Wendy Spencer, visits the DMC Google Pencil Code Apprenticeship (2015)



A Pencil Code Apprentice shares her project with guests at a WOW! Showcase Event in Massachusetts (2015)

Apprenticeship Description for WOW! Communications

Pencil Code is an exciting apprenticeship where students learn how to create simple computer programs--via Pencil Code--along with the beginnings of CSS and HTML. Students explore programming from the perspective of an artist, engineer, and designer. Students create outlined projects and design unique innovations to the projects along the way. Each student creates a WOW! Portfolio of his or her work and each portfolio features projects demonstrating various programming skills and showcases a final project of the student's design that highlights his/ her skills, interests, and creativity.

Apprenticeship Acknowledgements

"Pencil Code" was written and edited by Mandy Haeuser Gandin and Amy Hoffmaster, with support from Matthew Dawson, and the developer of Pencil Code, David Bau. Matthew, David, a growing cohort of Google volunteer Citizen Teachers--including Yana Malysheva and Ethan Apter-- and additional volunteers from John Hancock and Cognizant have taught this apprenticeship with students from several campuses in Boston and Chelsea Public Schools. We'd also like to thank Bob Cassels for his ongoing support and wealth of information.

Cindy Yang and Ji-Sun Ham developed an initial version of the curriculum, including the scope of concepts to be taught and the Pencil Code Cards, as part of coursework in the Technology, Innovation, and Education program at Harvard Graduate School of Education (2015).

We'd also like to thank Teaching Fellow Mallory Haeflich and Citizen Teachers from collaborative teams of John Hancock and Cognizant employees (Kevin James Egan, Joy Basu, Latisha Silvera, Rama Krushna Arugolla, Uma Dutta, and Nikhil Ribeiro) for their feedback on the initial curriculum and resources.