



Core Apprenticeship Library

Apprenticeship Sector: Coding & Gaming

Unit Guide: Pencil Code

Lesson #1: Welcome to Pencil Code!



Essential Question

- How can I use computer coding to design new ideas/ products?

Lesson Overview

Students will meet their teachers and be introduced to Pencil Code. After agreeing upon expectations for behavior, computer use, and professionalism in this apprenticeship, students will start coding! First they will learn basic drawing commands and then have the chance to tackle their first Pencil Code Card. Each week students will have the opportunity to select and work through multiple Pencil Code Cards. These cards provide little projects and challenges that reinforce the programming skill being taught and learned in that lesson. In this first lesson, the skills are to use basic drawing commands in block code in Pencil Code. Throughout this first lesson students will learn and practice various procedures--like how to pick a Pencil Code Card and how to save their work--that they will use throughout the apprenticeship.

Lesson Objectives

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

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

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

Lesson Objectives

- 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 5 different basic commands.

Lesson Agenda

10 Minutes	Hook: Welcome to Pencil Code!
5 Minutes	Introduction to New Material: What is Computer Programming?
20 Minutes	Activity 1: Expectations
20 Minutes	Activity 2: Human Coding
30 Minutes	Activity 3: Drawing & Cards
5 Minutes	Assessment: Work Product

Lesson Preparation

- Preparation:
 - Watch Tutorial 1 video.
 - Create a Pencil Code account. Play with Pencil Code and create the cards as practice.
 - 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



Core Apprenticeship Library

Apprenticeship Sector: Coding & Gaming

Unit Guide: Pencil Code

Lesson #1: Welcome to Pencil Code!



file-saving procedure taught in this apprenticeship (week#/cardname).

- Create your Word Wall. In each lesson you will add words to a word wall to build students' vocabulary. It is recommended that you create this Word Wall on a trifold board that you can bring to the apprenticeship each week. Post each word along with its definition. Review the previous week's words in context as needed. For this week, prepare the Word Wall board along with the words for this week (included in the Lesson Resources section at the end of this lesson).
- Prepare student folders to include:
 - Pencil Code Student Guide
 - Several blank pieces of paper for notes, doodles, and brainstorming
 - Ensure the folders have a pocket or add an empty sheet protector for the students to store their completed Pencil Code Cards
 - Consider using the labels included in this lesson to label, brand, and professionalize your student folders.
- Activity 2: Human Coding
 - Decide where to facilitate Human Coding. Ideally, an outdoor open area or a large indoor space would be used. Obtain permissions to use this space and take the apprenticeship there as needed.
 - Align Human Coding Cards to your space. See Lesson Resources. Four unique cards are included but likely will need modifications to tailor the "code" to your space. Ensure that the treasures will be hidden from obvious sight. You also may need more than four cards or pathways depending on your predicted class size.
 - Plan to have students find their student folders for the apprenticeship as the "treasure." Alternatively, prepare "treasures" that the students will find by following the code correctly. Suggestions could include pencils, fun school supplies, or company swag.
- Decide on and prepare your procedures for this class. Work closely with your TL or Citizen Schools staff co-teacher to plan procedures. Use the handout included in the Lesson Resources to help. Guidelines and recommendations are included for these required procedures.
 - Activity 3 is where you will teach the majority of your procedures for creating file names, saving versions of programs, getting and using cards, student folder use, and getting attention from the class during work time. Plan ahead not only for what you will expect students to do for each of these procedures, but also for how you will teach and reinforce each of these procedures.
- If you are not a computer programmer by trade or training, practice explaining what computer programming is as part of your preparation for this class. While simple, it can feel complex and is an explanation that needs to be practiced. Consider reviewing resources and explanations online like this [Intro to Computer Programming](#).
 - Key messages include:
 - Computer programming involves writing lists of instructions in a way a computer can understand--writing the code--to tell the computer to complete specific tasks, manipulate numbers, draw a picture, etc.
 - The instructions have to be specific and basic--each step has to be broken down to its simplest components.
 - There are many languages used to program computers (see Lesson 7). We are using Pencil Code, based on the language Coffeescript, for Lessons 1-6.
- Co-teaching plan:
 - It is recommended to use Team Teaching or the One Teach, One Assist method and tradeoff who is the lead for a majority of this lesson. This will introduce students to both the CT(s) and the TL and set the stage for a true shared teaching experience. Plan ahead for which teacher will take the lead on which activities.
 - In Activity 2, if you have co-teachers, consider using Parallel Teaching to facilitate this activity in smaller groups. You will need multiple spaces, but students will have more space to work and try out their human coding.



Core Apprenticeship Library
Apprenticeship Sector: Coding & Gaming
Unit Guide: Pencil Code
Lesson #1: Welcome to Pencil Code!



Materials

Every Class:

- Computer with internet access (one per student)
- Computer with projector for teacher use (one per class)
- Student folders (one per student)
 - Student Guide*
 - Storage compartment for completed cards
- Previous weeks' cards
- Roadmap to WOW!
- Expectations poster
- Word Wall
- Stickers (to indicate earning badges)
 - **Note:** It is recommended that you use a simple sticker such as a metallic star or even a colored dot to award to students for completing each challenge to earn each badge.

This Class:

- Index cards
- Ball
- Word Wall words and definitions (computer programming, program, running a program, programmer or coder)
- Blank poster with "Expectations" written at the top
- Human Coding Cards (one per student group in Activity 2) - four samples provided*
- Measuring tapes (one per student group in Activity 2)
- "Treasures" (one per student group in Activity 2) (optional - see Lesson Preparation)
- Week 1 Pencil Code Cards (one per student per card)*
 - (Snowman, House, Monkey, and Flowers)
- Chart paper

*These materials are located in the lesson folder, not in the Lesson Resources at the end of this lesson.

Hook: Welcome to Pencil Code! (10 Minutes)

Teacher's Note: This hook is the first introduction of the teachers to the students in this apprenticeship. While the activity cues the introduction of the CT, ensure that every teacher has equal voice and introduction time to message to the students that this is truly a shared partnership. While each teacher brings unique skills to the class, each teacher will carry equal authority and deserves equal respect when facilitating.

Teacher's Note: If you only have two-three teachers, consider having each teacher share something that they love about Pencil Code and play a previously prepared demonstration of the program feature. For example, one teacher might say that he or she really loves how you can program Pencil Code to make loops and could play a five-second program showing a neat pattern of loops. The teacher could also then share what week the skill will be mastered.

- **Reflect:** We use computers and computer programs everyday for many different purposes! Think about how you use computers (phones and tablets can count!). Write down one or two ways that computers have made your life easier.
 - **Write** this list on the board or display on a slide. **Hand out** index cards to each student.



Core Apprenticeship Library

Apprenticeship Sector: Coding & Gaming

Unit Guide: Pencil Code

Lesson #1: Welcome to Pencil Code!



- Computers can help you...
 - Find information
 - Write
 - Take pictures
 - Stay in touch with friends and family
 - Share highlights from your day
 - Buy items you need/ want
 - Work
 - **Note:** This is an excellent opportunity for CTs to share their own favorite examples of how computers have made their lives easier and why they love computers.
 - Give students one-three minutes to think and write down their answer on the index card.
- **Introduce yourself:** Citizen Teacher(s) (CT) introduces self by explaining connection to coding and as a teacher in this apprenticeship over the next ten weeks. Share one reason why you chose to volunteer as a CT and what you are excited to teach students. The TL should also introduce him/ herself as a co-teacher of the apprenticeship.
- **Facilitate shareback:** Sit or stand in a circle. Tell students that for this activity, you are going to roll or toss the ball to one of them, then they will introduce themselves and share one of their responses to the Hook. After they have shared, have them roll or toss the ball back to you. Then roll the ball to another student.
 - Note: Use this as an opportunity to memorize your students names. When the ball gets tossed back to you, say, "Thank you ___" and pass to the next student. Go back and see if you can remember everyone's name at the end and remind students of your name as well. Make it a fun challenge. Model that you make mistakes and it is ok. Laugh at yourself, but demonstrate diligence to accomplish the task--memorizing every student's name.
- **Transition:** All the examples you shared of how you like to use computers were so great! In this apprenticeship, we are going to learn the beginnings of how to create our own computer programs to do things we dream up. It will be a very creative process and you will learn from the perspective of an artist, a designer, and an engineer. Let's take a closer look at what that means.

Introduction to New Material: What is Computer Programming? (5 Minutes)

Teacher's Note: If you are not a computer programmer by trade or training, practice explaining what computer programming is as part of your preparation for this class. While simple, it can feel complex and is an explanation that needs to be practiced.

- **Say:** In this apprenticeship, you will use a programming language, Coffeescript, to create programs to complete simple tasks. These programs might draw a picture or they might create a game that you can play. We will, as computer programmers, write lots of different types of programs. This is how you will learn and build these skills. In the real world, people have careers in computer programming--whole careers dedicated to creating computer code!
 - Some people are artists: they focus on the visual aspect of the experience.
 - Some people are engineers: they make the building blocks and fit all those different pieces together to deliver a unified application.
 - Some people are designers: they create the content and the guiding premise for the code and focus on making the experience fun and making it fit together so it makes sense to people.
 - In this apprenticeship you will have the opportunity to learn skills that each of these types of computer programming careers use on a daily basis.
- **Say:** As a computer programmer, you might be all three of these--artist, engineer, or designer--or you might specialize in one. To better understand the nuanced differences between them, let's consider an example.



Core Apprenticeship Library

Apprenticeship Sector: Coding & Gaming

Unit Guide: Pencil Code

Lesson #1: Welcome to Pencil Code!



I'm sure you all know that there are shortcuts to emojis in texting or messaging. For example, an "<" and a "3" together will turn into a heart image. A ":" and a ")" together will make a smiley face. Here is each of these roles in making that happen:

- Artist: designs the heart or smiley face
- Engineer: writes the code to turn those keys into the image when used together
- Designer: knows from reviewing data and running tests that people will like these shortcuts!
- **Say:** When we're talking about designing video games, these roles play a big role because video games (like Angry Birds or Call of Duty) take tons and tons of work to create.
 - Artist: creates all the images and pictures you see
 - Engineer: creates all the code to make the game function
 - Designer: comes up with the concept for the game, plays the games or has others play the games to collect data to make the game as user friendly as possible AND as fun as possible
- **Say:** But what is computer programming? It is basically a lot like writing down steps it takes to do something. You, the computer programmer, write down all the steps it takes to do something in a specific language that the computer can understand and then the computer does each of the steps. We will talk more about this throughout this apprenticeship!
 - **Introduce** the Word Wall. Add "computer programming" to the word wall. **Read** the definition and **note** that we will add many words over the course of this apprenticeship.
 - **Word Wall Definition: Computer Programming** = The process of writing various sets of instructions to enable a computer to do certain tasks.
 - **Word Wall Definition: Programmer** = A person who writes the code that tells the computer to do certain tasks.
- **Say:** You will also master two standards that you will use again and again throughout the rest of your education in middle school, high school, and college--and on into your careers.
- **Review Standards: Say:** To that point, as in all apprenticeships, you will master one content area standard and one 21st century skill. Let's look at what they are for this apprenticeship.
 - **Reveal the standards:**
 - Innovation (21st cent)
 - Standard #1: Citizen Schools students will generate an original idea or product that suits a practical or artistic purpose
 - Standard #2: Citizen Schools students will use a design process to create ideas or products
 - Standard #3: Citizen Schools students will realize a product or idea that suits a given purpose
 - Computer Science
 - Standard #4: Implement problem solutions using a programming language, including: looping behavior, conditional statements, logic, expressions, variables, and functions.
- **Say:** As you can see from these standards, you're going to have a lot of opportunity to be creative while learning how to create computer code.
- **Say:** By the end of this apprenticeship, you will have created numerous computer programs! In fact, for your WOW! you will create a portfolio of your work by picking one program from each week to feature what you created and how you innovated upon the program. You will also create a final project to feature all your skills and creativity as the showcase project of your WOW! Portfolio.
 - **Note:** If relevant, share highlights from a past apprenticeship's WOW! Portfolio. Show how students will be able to create websites and more by the end of the apprenticeship.
- **Transition:** Before we dive into programming, though, we need to set some expectations for how we will work together and for how we will use the tools of this apprenticeship--our computers!

Activity One: Expectations (20 Minutes)



Core Apprenticeship Library

Apprenticeship Sector: Coding & Gaming

Unit Guide: Pencil Code

Lesson #1: Welcome to Pencil Code!



Teacher's Note: Be sure to include both apprenticeship-specific expectations and general campus expectations in this activity. When possible, the team leader can help make connections to campus' vision for collaboration or teamwork.

- **Say:** We are going to have a lot of fun together and I am looking forward to seeing your creative computer programs! To ensure we make the most of our time together, let's establish expectations for how we will interact with one another during apprenticeships.
- **Facilitate:** Let's think about this in terms of what will make a great environment for this apprenticeship. We will work hard and have a lot of fun; what type of learning environment will help us learn the most and have the most fun? Let's ask ourselves: how do we want to be treated and how do we NOT want to be treated? What do we want to see or hear and what don't we want to see or hear?
 - **Reveal (or turn around)** the Expectations chart paper. Currently this will be a blank piece of chart paper with "Expectations" written at the top and a big circle drawn on it.
 - **List:** Let's start with ideas about what we *want* to see, hear, and how we want to be treated.
 - **Label** the area inside the circle "A good programmer DOES:"
 - **Solicit responses** from students and make sure your top expectations make it on to the list as well.
 - **Write** these on the chart paper **inside** the circle.
 - Now, let's think about what we *don't want* to see, hear and/ or how we don't want to be treated.
 - **Label** the area outside the circle "A good programmer does NOT:"
 - **Solicit responses** from students and make sure your top expectations make it on to the list as well.
 - **Write** these on the chart paper **outside** of the circle.
 - **List:** Include any existing campus expectations that also apply during this apprenticeship. If applicable, write them in the circle as well. If the apprenticeship is taking place off-campus, add expectations that are applicable to that space. For example, if the apprenticeship takes place in an office, there may be specific expectations for noise level.
 - **Note:** If the apprenticeship is off-campus, you may also have specific procedures to review as well. For example, bathroom, materials, and arriving/ leaving the facility may all be procedures that are different from on campus. Although not written into this lesson, you will need to plan to explicitly teach and review these procedures.
- **Say:** Let's all agree to upholding these expectations by signing the inside of the circle.
 - **Facilitate:** Have each student come up and sign their name to the visual. Post this visual every class and refer back to it when necessary.

Activity Two: Human Coding (20 Minutes)

Teacher's Note: This is one of the few activities that takes place outside the classroom. Plan ahead for the location and secure extra permissions as necessary to take the students outside. Additionally, if you have co-teachers, consider using Parallel Teaching to facilitate this activity in smaller groups.

- **Say:** So coding or programming is essentially breaking a task down into the most specific steps possible to complete the task. This is harder than it seems! Let's try it.
 - **Transition** outside to where the Human Coding challenge will take place.
- **Say:** Think about what it takes to go from lying down on the floor to standing up. Let's stand in a circle around me. I am going to lie down on and I want you to take turns giving me instructions to get me to a standing position. We'll work our way around the circle until I'm fully standing!
 - **Assign** a student to begin and **facilitate** the activity.



Core Apprenticeship Library

Apprenticeship Sector: Coding & Gaming

Unit Guide: Pencil Code

Lesson #1: Welcome to Pencil Code!



- Encourage laughter as some of the instructions will certainly not be clear or specific enough for you to get up quickly.
- **Debrief:** What is difficult about giving instructions? How did you feel when you gave an instruction and what I did wasn't what you expected?
 - **Note:** Make sure students are taking away that breaking a simple task down into a set of instructions takes some thought--and sometimes it means trying a set of instructions and then trying again as needed.
- **Say:** In this apprenticeship, you will be given or will choose Pencil Code Cards. Each card challenges you to create a piece of code--a set of instructions--to make the computer do something. We'll get into the computer code later, but right now, we're going to try some "Human Coding," to see if you can physically follow a set of instructions to find the treasure.
 - **Show** a sample Human Code Card and demonstrate how to follow the instructions using the measuring tape and hints on the card.
- **Facilitate:**
 - **Put** students in groups of two-three.
 - **Hand out** cards. There are four unique cards that will send students to a different spot within a 20 ft. by 20 ft. outdoor area. However, each card will likely need to be tailored to your location.
 - **Establish** starting points and **give** student groups five-seven minutes to follow the instructions given on the cards.
 - **Note:** If you have more than four student groups, you might need to create more cards or simply plan ahead for different starting points to create more unique routes. As noted above in the Teacher's Note, you can also use Parallel Teaching to facilitate this activity with a smaller group. You will, however, need multiple locations if you go this route.
 - **Debrief:** What was difficult about this activity?
 - Draw out that it was tough to follow the instructions literally, and once they got the hang of literally just doing exactly what the card said, it was much easier. Highlight that this is similar to coding. The computer will only do literally what it is told, so we have to learn to think in this way to become great computer programmers.
- **Introduce** badge stickers.
 - **Say:** In this apprenticeship, as you master skills, you can earn badges. Open your student folder to the correct page. This is your badge page. To earn a badge, you must complete three challenges. Each of you has just completed your first challenge for Badge #1: Human Coding!
 - **Place** a sticker in each student's folder on the Human Coding challenge in Badge #1.
 - **Note:** If this is too time intensive, you can also hand out the stickers in Activity Three as you work your way around the room while students work on their cards.
 - **Highlight** badges that students will have the opportunity to earn today.
 - **Say:** Depending on which cards you chose to work on and complete, you might be able to complete the other two challenges for Badge #1 today. You might even be able to complete one challenge in Badge #2!
 - **Teacher's Note:** Make a big deal about the challenges and badges. This is a great opportunity to invest students in completing--and learning--each of these programming skills. Additionally, consider providing a bigger reward for completing all six badges. For example, all students who complete all six badges by the end of Week 8 will be able to choose their favorite swag from my company: a mini football, a magnetic picture frame, or a pair of sunglasses. Prep co-teachers before class to know which challenges will likely be completed today, and have all co-teachers award stickers for challenge completion as they circulate the room.
- **Transition:** Head back to the classroom and get ready to get on the computers! Let's meet Pencil Code!

Activity Three: Drawing & Cards



Core Apprenticeship Library

Apprenticeship Sector: Coding & Gaming

Unit Guide: Pencil Code

Lesson #1: Welcome to Pencil Code!



(30 Minutes)

Teacher's Note: This activity is where you will also teach the majority of your procedures for creating file names, saving versions of programs, getting and using cards, student folder use, and getting attention from the class during work time. Plan ahead not only for what you will expect students to do for each of these procedures, but also for how you will teach and reinforce each of these procedures.

- **Review** expectations for computer use, as discussed in Activity 1.
- **Teach** any procedures for getting on the computers. Plan to implement the following procedures:
 - Assign seats. You can rearrange as needed since the software is cloud-based instead of downloaded on the computer, but having an assigned workstation is recommended.
 - Getting attention. You will need a way to indicate that it is “teacher talk” time vs “individual work” time. For example, you could say “1,2,3 all eyes on me!” and students will be expected to lower their screens or turn off their monitors and look at teacher. Then when you are finished talking, you could say “1,2 back to you,” at which point students could raise their screens or turn their monitors back on and get back to work.
 - Getting student folders. Students could pick up their folders as they come in the room or they could already be at their workstation.
 - **Note:** See the Teacher Handout below for a complete list of procedures and planning materials.
 - **Practice** each procedure you teach at least twice, and expect it to be followed each time it is used.
 - It is recommended that you take five-seven minutes here to teach and practice both the getting set up and the wrapping-up/ shutting-down procedures. By teaching and practicing both, you will be able to reinforce both. For example, you might cue students to:
 - Turn on computers
 - Get folders
 - Open Pencil Code
 - Wrap up work (fake this one since they are not in the middle of work)
 - Shutdown computers, put away folders
 - Then, repeat the initial procedures: Turn on computers, get folders, and open Pencil Code before continuing.
- **Note:** At this point students should all be seated at their computers with their student folders.
- **Introduce** students to Pencil Code. **Lead students** through creating usernames and passwords.
 - Find site (pencilcode.net).
 - Create a login (give parameters for students to create usernames and passwords).
 - Write the username and password in their student folders.
 - **Note:** It is recommended that all students have the same password. Keep a teacher copy of all usernames.
- **Show** the [Tutorial 1 video](#). (5 mins)
 - (If needed or if the video isn't an option) **Demonstrate** how to drag and drop blocks of code, how to modify the parameters, and try them out by pressing the play button.
 - **Word Wall Definition: Program** = A group of code that performs a specific task OR many tasks.
 - Give the example of a program they might create in Pencil Code and Microsoft Word as another program. The only difference is scale.
 - **Word Wall Definition: Running a Program** = When a program runs, this means that the computer is reading the code and executing the tasks that are written in the code.
- **Handout** the first card, “Do you want to build a snowman?”
 - **Give** students five-ten minutes work time. Encourage them to use trial and error as well as the tips on the card to successfully create the snowman.
- **Pause:** Teach file saving procedures. Reference the correct page of the Student Guide.
 - Save your file before you create something new.
 - Save files with the name beginning with the week#. This week, all files should start with “week1....”
 - After the week #, type a backslash. This backslash will create a file folder - also known as a directory - and it will put all the programs you create today in that directory.



Core Apprenticeship Library

Apprenticeship Sector: Coding & Gaming

Unit Guide: Pencil Code

Lesson #1: Welcome to Pencil Code!



- After the backslash, the file name should connect to what you're making. For example, you might start with "week1/snowman." When you make an innovation on the snowman, you might call the new program "week1/snowman1."
- Note: Each file will be saved as a unique file in your file folder. Show students how to access their files.
- **Say:** At this point, if you've already completed your snowman, you have a choice! You can either try a new card or you can create some innovations to your snowman.
 - **Brainstorm** some ideas for innovations to the snowman. Take answers from students popcorn-style. Sample answers might include: change the color or size of the snowman or buttons, add a fourth circle to the snowman, add ground, add a carrot nose, etc.
 - **Write** this brainstorm on a piece of chart paper if possible, and save it for use in Week 2.
 - **Teach** procedure for getting a new card.
 - **Plan** ahead for what you want this procedure to be. As noted in the Unit Guide, one of two options is recommended.
 - Option 1: Display the cards for the day on a table that the students can see as they enter the class. Use a file box or binder with sheet protectors to store past weeks' cards in an organized and easy to access manner. To obtain a new card, students learn to go to the table to pick up a new card for the day or find a past week's card in the file box or binder.
 - Option 2: Put all the new cards for the day in each student's folder before class begins. Students can access and review the cards as they look at their folders. Only cards used remain in folders and all past weeks' cards are stored in a file box or binder with sheet protectors.
 - **Note:** Today all students start with the snowman, but in future lessons, students will select which card to do first and/or the teacher will recommend a card to start with based on the student's interests and abilities.
 - **Say:** Today, I'd like to know which you plan to do. Please hold up one finger if you plan to innovate on your snowman and two fingers if you'd like to get a new card.
 - **Note:** If you're using a table to display and house your cards weekly and 90% of the students want to get a new card, assign three-five students to go up to get a new card at a time.
- **Give** students another ten minutes work time, or more if available.
 - **Award badges** for completing the challenges in Badge 1 and perhaps the pen challenge in Badge #2. After students have completed all three challenges in Badge #1, put a sticker on Badge #1 in the table after Badge #6 as well to mark that the student has completed the entire Badge.
- **Bring** the class back together. Reinforce your "Getting Attention" procedure.
- **Teach** wrapping up procedure. Give students a designated amount of time (e.g. 45 seconds) to finish their thought and save their work.
- **Transition:** Each week, we will have cards that teach us programming techniques and skills. Our class will feel very much like a workshop, lots of individual work time and lots of your choice in terms of what you create. This is the way that many career computer programmers work--independently or in collaboration with small groups on projects--so in this apprenticeship, this is how we will work too.

Assessment Work Product (5 Minutes)

- **Say:** Sometimes we will have a handout to turn in, but in addition, each week you will pick one of your projects from the day to save as your assessment. This should be whatever you are most proud of from today. Pick that project now and save it as "week1/assessment." In your student folder, fill out the Week 1 Assessment in the Work Product Assessment section.
 - **Walk students through** how to fill out the Week 1 Assessment written portion. (See Student Guide.)



Core Apprenticeship Library
Apprenticeship Sector: Coding & Gaming
Unit Guide: Pencil Code
Lesson #1: Welcome to Pencil Code!



- I'm proud of this program because _____
- Check off at least 5 commands this program uses: _____
- What is our WOW!? _____

Say: Before next week I will check out your work and next week I'll give you some feedback! Also, next week we'll get a chance to share some of our work with each other. We will learn new programming techniques and have new cards to create.

- **Note:** If there is time available, share one or two student projects from today or foreshadow skills to be learned next week.
- **Connect to WOW!** This project will also be a part of your WOW! Portfolio. Earlier in this apprenticeship we talked about your WOW! Portfolio. Does anyone remember what it is?
 - Take several student answers and fill in the gaps to reiterate that the WOW! Portfolio is the collection of each students' work--one project per week--and a final project that will feature all their skills and creativity.
- **Connect to Week 2:** Next week we will learn loops and more ways to control our turtles in Pencil Code! Loops mean that the turtle draws something in a repeated fashion to create patterns or multicolored objects.
 - If possible, demonstrate using Pencil Code.
- **Review and follow** the procedure for shutting down the computers and cleaning up for the day.

Word Wall - Lesson #1

Computer Programming



or Computer Coding

The process of writing various sets of instructions to enable a computer to do a certain task.

Programmer or Coder

A person who writes the code that tells the computer to do certain tasks.

Program

A group of code that performs a specific task
OR many useful tasks.

Run a Program

When a program runs, this means that the computer is reading the code and executing the tasks that



Core Apprenticeship Library
Apprenticeship Sector: Coding & Gaming
Unit Guide: *Pencil Code*
Lesson #1: *Welcome to Pencil Code!*



are written in the code.



Core Apprenticeship Library
Apprenticeship Sector: Coding & Gaming
Unit Guide: Pencil Code
Lesson #1: Welcome to Pencil Code!



Teacher Preparation - Planning for Procedures - Lesson #1

Use the following table to plan for the procedures you will use in this class. Teach them in Lesson 1 and 2, and reinforce/ reteach them as needed throughout the apprenticeship.

Procedure	Comments & Recommended Options	Apprenticeship Procedure: Note what option you're going to use, plus any modifications and notes
Assign Seats	Assign seats for students to sit in each week.	
Get Attention	You will need a way to indicate that it is "teacher talk" time vs "individual work" time. For example, you could say "1,2,3 all eyes on me!" and students will be expected to lower their screens or turn off their monitors and look at teacher. Then when you are finished talking, you could say "1,2 back to you," at which point students could raise their screens or turn their monitors back on and get back to work.	
Student Folders	Students could pick up their folders as they come in the room or they could already be at their workstation.	
File Naming and Saving	Save files with the name beginning with the week #. This week, all files should start with "week1...". After the week #, type a backslash. This backslash will create a file folder - also known as a directory - and it will put all the programs you create today in that directory. After the backslash, the file name should connect to what you're making. For example, you might start with "week1/snowman." When you make an innovation on the snowman, you might call the new program "week1/snowman1."	
Getting a New Card	Option 1: Display the cards for the day on a table that the students can see as they enter the class. Use a file box or binder with sheet protectors to store past week's cards in an organized and easy to access manner. To obtain a new card, students learn to go to the table to pick up a new card for the day or find a past week's card in the file box or binder. Option 2: Put all the new cards for the day in each student's folder before class begins. Students can access and review the cards as they look at their folders. Only cards used remain in folders and all past weeks' cards	



Core Apprenticeship Library
Apprenticeship Sector: Coding & Gaming
Unit Guide: Pencil Code
Lesson #1: Welcome to Pencil Code!



	are stored in a file box or binder with sheet protectors.	
Wrapping Up	Option 1: Play a song. When the song concludes, all work must be saved and completed for the day. Option 2: Announce that it is time to wrap up. Set a timer for two minutes. When the timer goes off, all work must be saved and completed for the day.	
Assessment File	Save your best work for the day as "week1assessment"	
Shutting Down	Consider how and when to have students shut down their computers and clean up for the day.	



Labels for Student Folders



PENCIL
code

