Core Apprenticeship Library
Apprenticeship Sector: *Coding & Gaming*
Unit Guide: *Pencil Code*
Lesson #2: *Loops & Text Code*

Essential Question

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

Lesson Overview

Students will learn how to create loops in today's lesson! Additionally they will learn how to switch back and forth between block code and text code and learn the benefits of each. In block code, the spelling, spacing and commas are all "blocked" together so that students can drag and drop code into place. Students have a palette of code to choose from. In text code, however, there is more that students can control, but also syntax and details must be mastered. Students will have the chance to work with five new cards and begin to formally consider the process of innovation. They will add a project to their WOW! Portfolios and continue building skills that they will use to complete their final projects.

Lesson Objectives

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.

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

- SWBAT to use loops and indented code to create drawings that use repetition.
- SWBAT use and switch between text code and block code language.

Lesson Agenda

5 Minutes	Hook: Feature Project
10 Minutes	Introduction to New Material: Loops & Text Code
30 Minutes	Activity 1: Cards
15 Minutes	Activity 2: Game OR Share Projects
25 Minutes	Activity 3: Innovations
5 Minutes	Assessment: Work Products

Lesson Preparation

- Preparation:
  - Watch Tutorial 2 videos (1 and 2). Decide if you want to show these.
  - Practice the Pencil Code Cards for this week. 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 taught in this apprenticeship (week#/cardname). Innovation ideas are included for some cards.
  - If you haven't already, create a master site that links to each student's directory. A sample is provided on the Citizen Schools Pencil Code site. Having a master or a teacher view will allow you to easily pull up any student's work to evaluate what they are doing, review assessments, provide feedback, or share their work with the class in the Hook or Activity 2.
  - Prepare words to add to the Word Wall.
  - Prepare copies of handouts.
  - Review student assessment work from the previous week. Write a note to the student appreciating some aspect of their work, making a recommendation for further innovation, and recognizing their work and learning.
  - Pick a project from last week to feature for the Hook. Ideally you feature a different student's work each week. Select a student whose work is exemplary and might push other students' innovations, thinking, and creativity further.
  - Watch the video "Mark Zuckerberg explains Loops" by code.org. Cue it up ahead of class to play during Intro to New Material.
  - Look ahead to Lesson 5's guest speaker panel. Begin planning this week if you haven't already coordinated speakers. Decide who will take the lead on communication and coordination.

- Co-teaching plan:
  - You are likely to have new students joining the class this week, so consider having a co-teacher pull them aside and facilitate a modified lesson to the small group combining a review of expectations from Week 1 (and having the students sign the poster) along with an introduction to Pencil Code and the cards from Week 1 and Week 2.
  - This apprenticeship is structured as an ongoing workshop, but some students might have a hard time working so independently immediately. If you have a co-teacher, consider having this teacher sit down with a small group of students who need more guidance and teach Activities 1 and 3 in a more guided and structured format.

Materials

Every Class:

- Computer with internet access (one per student)
- Computer with projector (one per class)
- Student folders (one per student)
  - Student Guide*
  - Pencil Code Cards completed by each student (from previous weeks)
- Previous week's cards organized in a binder or file box
- Roadmap to WOW!
- Expectations poster
- Word Wall
- Stickers (to indicate earning badges)

This Class:

- Week 2 Pencil Code Cards (one per student per card)*
  - (Traffic Light, Bracelet, Star, Rainbow, Stop Sign)
- Brainstorm of Snowman Innovations (from Week 1)
- Word Wall words and definitions (innovate, innovation)
- Handout - Innovation - Lesson #2

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

Hook: Feature Project (5 Minutes)

Teacher's Note: *This Hook repeats each week and is an opportunity for you to highlight each student's work. Ideally you feature a different student's work each week. Select a student whose work from the previous week is exemplary and might push other students' innovations, thinking, and creativity further.*

- **Say:** Welcome back to Pencil Code! Let's begin this week by taking a look at some of the awesome work you completed last week!
  - **Play** the program from the featured student of the week.
- **Point out** why you selected this project to share. Highlight an innovation, a unique piece of code, or way the student was thinking creatively.
- **Ask** the student who is featured if he or she wants to share anything about the project or ask for any feedback.
  - **For example,** the student could share a future innovation that he or she thought would be cool, but isn't sure how to do or the student could ask for other students' ideas of what to try next.
- **Transition:** Today we're going to learn how to include loops and repetition in our code. Let's get started!

Introduction to New Material: Loops & Text Code (10 Minutes)

Teacher's Note: *Decide ahead of time if you'd like to show the tutorial video or if you want to demonstrate how to use text code and create loops.*

- Briefly **review** commands covered in Week 1 (dot, draw, fd, bk, rt, lt, etc)
  - **Say:** We talked about this briefly last week, but the computer language that we're using in Pencil Code is called Coffeescript.
    - **Word Wall Definition: Coffeescript** = a programming language based off of the popular Javascript language. It is generally considered to be a more concise and readable version of Javascript.
  - **Note:** If you have new students joining the class this week, consider having a co-teacher pull them aside and facilitate a modified lesson to the small group combining a review of expectations from Week 1 (and having the students sign the poster) along with an introduction to Pencil Code and the cards from Week 1 and Week 2.
- **Show** the video "Mark Zuckerberg explains Loops" by code.org. Cue it up ahead of class to play during Intro to New Material.

- **Say:** Before we dive into Pencil Code today, we need to learn a few new concepts.
    - **Show** Tutorial 2-1 and/ or **highlight** key messages on how to use the command "For" and switch between text code and block code. Outline the benefits of each.
    - Key Messages:
        - **Show** how switching from block code to text code allows you to easily cut and paste code to repeat.
        - **Show** how block code provides a palette of commands to choose from and makes it easy to explore whereas text code requires more detail but allows a broader range of commands.
        - **Demonstrate** using the "For" command to tell the computer to repeat a set of code a certain number of times.
        - **Word Wall Definition: Parameter =** Limits or boundaries that tell a computer how to run a function.
        - **Explain** that the parameters of the "for" function in this example is the number of times you tell it to repeat.
    - **Show** Tutorial 2-2 and/ or **highlight** key messages on how to use the command "For" with a variable.
    - Key Messages:
        - **Demonstrate** using the "For" command with a variable to tell the computer to repeat a certain set of code with a variation each time.
        - **Word Wall Definition: Variable =** A piece of text, such as "a" or "t" that stands for a specific piece of information.
        - **Explain** that the variable in this example is "x" and is used to change the degree of the turn slightly or the color of the dot in these two examples (in the tutorial video).
- **Review** the basic agenda for the day. Note that this is generally the schedule that we will use in this class each week. The class is very much run workshop style, and students will guide their own work.
    - **Note:** Remember to review, reteach, and practice all procedures you taught in Week 1. 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.
    - Agenda:
        - Work time
        - Pause for sharing projects
        - Think about innovations
        - More work time
        - Save an assessment for the week
- **Transition:** Cue the procedure to get folders, turn on computers, and get to work!

Activity One:
Cards
(30 Minutes)

Teacher's Note: *Some students might have a hard time working so independently immediately. If you have a co-teacher, consider having this teacher sit down with a small group of students who need more guidance and teach this activity in a more guided and structured format.*

Teacher's Note: *If work time is off-task for a majority of the group, stop the class and start over. Reset with clearer directions. Emphasize accountability and clear incentives for at least two cards completed. If absolutely necessary, consider providing an "off computer" coding activity for students who continue to not be able to stay on task. Sample computer programming "unplugged" activities can be found here or here.*

- **Handout** Bracelet Card for all students to start with today. This will connect their actual coding to the tutorial videos.
    - **Give** students 25 minutes of work time.
    - Once students complete the Bracelet Card, they should place it in their "Completed Cards" section in their folder and follow the appropriate procedure to select another card.
    - **Remind** students to NAME and SAVE their projects regularly. Remember, all projects saved today should start with "week2/" to create a "week2" file folder or directory.
    - During this work time, CTs should be rotating around the room, checking in with students, assisting as needed, and encouraging students to try a new card or a new step in their programming.
        - Encourage students to switch back and forth between text code and block code as shown in the Tutorial video. There are benefits to each.
        - **Reminder:** Hand out stickers to students as they complete challenges for the badges. In this activity, most of the students will complete the "for" challenge in Badge #2.
- **Bring** the group back together
- **Say:** We are going to have more work time in a bit, but first I want to pause and highlight some great work that's being accomplished and think more about the innovations we can make in our projects.

Activity Two:
Game or Share Projects
(15 Minutes)

Teacher's Note: *Use this time to either share projects that students are working on or share a project that pushes their skills further.*

- **Say:** let's take a look at xx student's project this afternoon…
    - **Share** two-four different students projects. Highlight how they used loops to create repetitions. Additionally, highlight ways that it might have been easier to use text code vs. block code while writing the code.
    - **Share** feedback and ideas for further work.

Alternatively, if students are not ready to publicly share work, consider playing a quick "Guess the Code" game.

- **Say:** I'm going to show you two sets of written code and two programs being run. The game is to guess which code goes with which program.
- **Use** the following links to two rounds of "Guess the Code"
    - Day 2, Game 1
    - Day 2, Game 2
- See all games available for this section at http://inventgame.pencilcode.net
- **Connect** to concepts from today and recognize the innovative ways these programs use loops.

Activity Three:
Innovations
(25 Minutes)

- **Say:** As you know, we are also focusing on innovations in this apprenticeship. We've already started using the word and talking about innovations we've made, but we haven't really defined what innovation really means so let's stop and do that now.
    - **Display** the word and definition for Innovate and Innovation.
    - **Word Wall Definition: Innovate** = To make changes to something to make it better or different in some way.
    - **Word Wall Definition: Innovation =** Changes made to make something better or different.
    - **Explain:** To innovate is to create something new or to create a new way of doing something. Contrary to popular belief, innovation does not require raw creativity, rather it requires the willingness to learn and explore from what exists, ultimately to imagine and create what does not.
    - Add both of these words to the Word Wall.
- **Say:** There are three stages of innovation and today we are going to focus on the first, but you'll likely do all three. The three stages are:
    - 1. Generating Ideas--Brainstorm ideas of what you can do and pick the best idea
    - 2. Design Process--Create a plan for how you can create it
    - 3. Realizing the Idea--Do it!
- **Say:** Last week we did stage 1 together when we talked about what innovations we could possibly make to the snowman to make it different and/ or better.
    - **Show** the brainstormed list from the previous week, if possible. The list likely included ideas to change the color or size of the snowman or buttons, add a fourth circle to the snowman, add ground, add a carrot nose, etc.
    - **Say:** This is the first step of innovation: Generating Ideas.
- **Say:** Let's do this together. Let's go back to the bracelet and brainstorm various innovations we could make to the bracelet.
    - **Solicit** student answers. **Write** them down on a piece of chart paper or a whiteboard. Encourage students to give any and all answers.
    - **Sample responses:** Lengthen the bracelet, change the color of the bracelet, make some beads large and some beads small, make the bracelet spin, make the dots look like real pearls, put the bracelet on a hand, etc.
    - Ensure you have a **wide range of responses--**some that are feasible and some that, for the students' current coding abilities, are not.
- **Say:** Now that we have a lot of great ideas of how we could innovate on the bracelet, let's take the next step. We need to look at our list again in terms of what is feasible: What can we do? What are the advantages or limitations of the idea? Additionally, consider what interests you about the idea: Is it an idea you're excited about?
    - **Model** how you would go back through the list and consider what ideas you can currently accomplish.
        - Lengthen the bracelet, change the color of the bracelet--one advantage is that these ideas are fairly basic. I could easily make these innovations. They are definitely feasible, but one disadvantage is that they aren't be too interesting to me.
        - Make some beads large and some beads small--this is interesting! I can change the size of a dot, so I bet I can figure out how to program this idea. By pursuing this idea, I'd learn something new. That is one advantage.
        - Make the bracelet spin, make the dots look like real pearls, put the bracelet on a hand, etc.--these ideas all seem good, but a little complicated for what I know now. This is not feasible and that alone is a disadvantage of these ideas.
- **Hand out:** Handout **-** Innovation
- **Explain** the expectation that students complete this handout for one card, one they have already been working on and want to innovate on (from Week 1 or 2) or for a new card (from Week 1 or 2).
- **Give** students 10-15 minutes of work time.
- **Remind** students to NAME and SAVE their projects regularly.
    - **Note:** Consider setting a "Time to Save" timer and when it goes off, everyone saves their work. It will take some routine for students to get used to pausing to save so plan ahead to help them build in this routine.

- During this work time, CTs should be rotating around the room, checking in with students, assisting as needed, and encouraging students to try a new card or a new step in their programming.
  - Check-in on the Innovation handout also. Ensure that students have generated at least two ideas for innovation on one card for each category.
  - CTs should also hand out stickers to award badges for the day. Most students should complete the "for" challenge in Badge #2. Some may also complete the "moveto jumpto" challenge. If students complete the entire Badge #2, place an additional sticker on Badge #2 in the table after Badge #6 to indicate that the student has completed the whole badge.
- **Bring** the group back together and **Transition** to the Assessment.

Assessment
Work Product
(5 Minutes)

- **Say:** Today I will collect the Innovation handouts, along with your submitting a project for your assessment. Be sure to pick the project that you are most proud of from today. Pick that project now and save it as "week2/assessment." In your student folder, fill out the Week 2 Assessment section.
  - **Walk students through** how to fill out the Week 2 Assessment written portion, if necessary. (See Student Guide).
    - I'm choosing this program to submit because ______________________
    - *Make sure your program uses the command: **For**!*
  - **Say:** Before next week I will check out your work, and next week I'll give you some feedback! Also, next week we will get a chance to share some of our work with each other. We will learn new programming techniques and have new cards to create.
  - **Collect** student folders and the Innovation Week 2 handouts.
  - **Note:** If there is time available, share one or two student projects now.
- **Connect to WOW!** This project will also be a part of your WOW! Portfolio. Earlier 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 student's work--one project per week--and a final project that will feature all their skills and creativity.
- **Connect to Week 3**: Next week we will learn how to add buttons, keydowns, and clicks to our programs in Pencil Code!
  - If possible, demonstrate by showing a program in Pencil Code.
- **Ask** students to follow the procedure for shutting down the computers and cleaning up for the day.
- **Ensure** all students have received stickers for the badges they have earned!

Word Wall - Lesson #2

Coffeescript

A programming language based off of the popular Javascript language. It is generally considered to be a more concise and readable version of Javascript.

Innovate

To make changes to something to make it better or different in some way.

Innovation

Changes made to make something better or different.

Variable

A piece of text, such as “x” or “t” that stands for a specific piece of information.

Parameter

Limits or boundaries that tell a
computer how to run a function.

Handout - Innovation - Lesson #2

Name:____
_

Card to Innovate: ____

Generating Ideas: In the box below, **write down as many ideas as you can think of** that reflect ways you could innovate on the card you have chosen.

Sort Ideas: Consider if the idea is feasible given the programming skills you have now. Would it be easy, medium, or difficult to accomplish? What are the advantages or limitations to each idea? **Write at least 2 ideas in each category.**

Easy	Medium	Difficult

Consider Interest: Consider how interested you are in trying each idea. Do you want to do try to create it? **Circle at least 3 of the above ideas** that you are genuinely interested in trying to do. **Star one and try to create it!**