Computer Science Fundamentals

The Kano Curriculum
Contents

Intro to Kano

Why Kano?

Who Uses Kano?

Why Is Kano Valuable for Students?
  STEM, STEAM, STEAMED...
  21st Century Skills
  Differentiated Learning

Lesson Plan Pedagogy

How to Use Worksheets with Kano

Learning Objectives

What Is in This Lesson Plan Packet

Lesson 1: Build a Computer!

Lesson 2: What Is Code?

Lesson 3: What Is Code Pt. 2?

Lesson 4: Explore Story Mode

Lesson 5: Intro to Make Art

Lesson 6: Intro to Hack Minecraft

Lesson 7: Into to Make Snake + Make Pong

Lesson 8: Algorithms

Lesson 9: Abstraction

Lesson 10: Pattern Recognition

Lesson 11: Decomposition
Lesson 12.1: Variables
  12.2: Data Types - Numbers
  12.3: Data Types - Strings

Lesson 13: Loops

Lesson 14.1: Conditionals
  Lesson 14.2: Conditionals - Booleans

Lesson 15: Debug It!

Lesson 16: Impact of Technology

Design Thinking

Building a Better Community

Celebrate Creation!

Student Engineering Log

Surveys + Kano

KANO Student Survey
Introduction to the Kano Curriculum

Every day, Kano Educators all over the globe empower young people to build technology and create with code. In the 3 years we’ve been supporting our educator community, we’ve learned so much about the inspiring ways Kano is used in all different types of learning spaces. Kano is being used to teach computer science and coding in classrooms, in makerspaces to develop a DIY mindset, in libraries to enhance digital literacy, and more! In those learning spaces, educators and facilitators are helping young people demystify the hardware and software that makes computers work and we’ve seen some amazing creations.

With help from our champion educator community, we’ve put together this 18-week Kano Curriculum to help you integrate computing and coding in your learning space. The lesson plans in this packet are a mixture of teacher-led structure and student-led learning, designed to enable your students to progress in their understanding of technology, while inspiring them to create with it. You don’t have to be a computer science teacher, or have any background in computer science, to teach with Kano – these lessons are designed to help you get started with creative computing, no matter your level of experience.

Each lesson comes with an accompanying worksheet to help check what students understand and to celebrate their successes with them, their peers, and their family.

Why Kano?

Kano is focused on bringing a creative approach to computer and coding education. We are creating a world where anyone, anywhere, can learn make and play with technology, not just consume it. From students in Sierra Leone who have never had a computer to young makers in the UK and the US – all are part of the new creative generation.

Who Uses Kano?

Over 1,000 classrooms, clubs, camps, and community programs are using Kano to engage students in understanding and participating in the digital world. With Kano, you’re truly part of a global community; 150,000 Kano Kits have been shipped to young people in over 86 countries. Our aim with Kano for classrooms is to make computing education accessible not only to students but also to teachers.
Why Is Kano Valuable for Students?

Kano is designed to build a generation of creators, not just consumers. Our education programs are part of this movement, to empower the innovators and designers that will address some of the grand challenges we face as a global society. Kano is the best tool for programs at the intersection of STEM education, 21st century skill development, and differentiated learning.

STEM, STEAM, STEAMED...! Education

In recent years there has been a push to promote STEM (Science, Technology, Engineering, and Mathematics) Education. You may have also seen the acronym with the “A” included for “Arts,” and even with an “ED” at the end to include Entrepreneurship and Design! The objective in any case is clear: we know that developing learners able to solve problems with technology is critical for the jobs of the future. Kano believes that developing STEM skills is not limited to these subjects, but as a cross-curricular movement.

Kano’s projects and challenges pushes learners to develop STEM skills in a fun and engaging environment. Kano develops challenges that are available online for free so learners can select projects they find fun and relevant.

21st Century Skills

How we teach must reflect how our students learn, it must also reflect the world they will emerge into and shape – a world that is rapidly changing and is increasingly connected. Our style and approach to teaching must emphasize acquiring basic digital literacy necessary for succeeding in the 21st century, with both teachers and students at the core. Kano’s hardware and software weave creativity, technology, and coding across subject areas, to promote deeper learning in an engaging and hands-on way.

Kano’s technology, along with our guided lesson plans, ensure that we weave the 4Cs into teaching and learning:

Creativity and Innovation
Critical Thinking and Problem-Solving
Communication
Collaboration
Our lesson plans and student-led approach help to build life and career skills, including:

- Flexibility and Adaptability
- Initiative and Self-Direction
- Social and Cross-Cultural Skills
- Productivity and Accountability
- Leadership and Responsibility

**Differentiated Learning**

Kano offers immediate differentiated learning in its operating system. When you log into Kano, you can access all of the apps and go through walkthrough tutorials to learn about different coding languages. All of the apps have challenges that range from basic, to medium, to advance. Each app also includes “playground mode,” where anyone can explore coding and make creations all their own! Some of our most advanced coders like to explore what others have created and remix the code to make their own piece of art, game, or song; the possibilities are endless!
Lesson Plan Pedagogy

All lesson plans in this packet are designed with structure in mind. Below is a guide to help you understand the structure and adapt it to your learning environment.

We use scaffolding to help learners acquire new knowledge. As with other subjects, with computer science, the scaffold provides a temporary framework that is put up for support and access to meaning and is then taken away as needed, when the learner secures control of success with a task.

In this framework, learners are explicitly taught and instructed in the following way:

1. Linking to prior knowledge
2. Teacher Connection (“I Do”): Modeling a new concept or topic either through lecture or examples on the board. This is the explain time where teachers talk 80% and students talk 20% (more listening).
3. Engage (“We Do”): This is the point where the educator works with the students to learn a new concept. The teacher talks 40% of time and students talk 60%.
4. Practice/Exploration time (“You Do”): The students run free. They are actively exploring a new concept or building things on their own. The majority of class time should be spent with the students actively engaged in the material. 90% of talking is from students, 10% from teachers.
5. Evaluate (“What can you do?”): How can we check for understanding? Is this in the form of remodeling what was explored by rethinking in new ways? Is it with some type of formal evaluation?
6. Closing: Wrap up the day in some type of closure. Is it giving a homework assignment? Is it reflecting in their engineering logs?

These 6 steps can help organize lesson plans and projects for Kano.

The key areas are:

1. Linking
2. Engage
3. Exploration: Challenges
4. Sharing
5. Evaluation
6. Closing
Within each section there are guiding questions that can be incorporated:

1. **Linking**
   a. What is the objective of your lesson, and how can you get students interested in it? Activate their prior knowledge!
   b. Get students involved and thinking by presenting a problem, brainstorming, playing a game. Energize the students.

2. **Engage**
   a. What is the specific knowledge you are trying to get across? Is it the goal for the day – for example Exploring Make Art? Is it designing a logo?
      i. Name the logo, explain what it is and how it is used.
   b. Get feedback from students and play off of each other’s ideas

3. **Exploration: Challenges**
   a. This is the time where students explore, and it should be the bulk of activity time. When you are making lessons make sure that the activity in some way:
      i. Links to your greater objective
      ii. Helps students focus and work toward the objective
      iii. Helps students think of new and creative ways to explore and create
   b. There are multiple challenges in each lesson. If a learner finishes one challenge, allow them to move on to the next challenge in the lesson.

4. **Sharing**
   a. Always give your students the opportunity to share their thoughts with the community, be it with their partners or the larger Kano community on Kano World.

5. **Evaluation**
   a. What questions, processes, or steps can you use to help students evaluate themselves?
   b. What can you do to push students to rethink what they created?

6. **Closing**
   a. This is the last remarks you make for the day. What is a closing thought you want to get across?
   b. This section will also connect to homework examples.
How to Use Worksheets with Kano

All lesson plans in this packet will have a worksheet that can be used during or after the Kano session. Each lesson is divided into three parts: Explorer, Programmer, and Creator.

Explorer:
This section can be done during the Kano session or after. It is typically an activity where learners have to find or remember key elements of the lesson. For example, the “Explorer” section in “Intro to Kano Code” would be to identify what blocks live under which category in Kano Code.

Programmer:
This section should be done after the Kano session and be used as a formative assessment on the learner’s use of the Kano for the day. Typically the Programmer Section is 1 - 2 exercises that ask probing questions on what the learners learned. For example, in the “Intro to Scratch” lesson, the learner may be given a code block sequence, and they have to explain what the code would do.

Creator:
This section can be after the Kano session. In the “Creator” section learners are asked to be creative and think of new ideas for a design or code. For example, learners may be asked from the “Make Art Pattern Perfection” lesson to design a new pattern and explain how they think they could code this new pattern in Make Art!

Impact:
This section can be after the Kano session. The “Impact” section asks more probing questions on how the concept we learned is connected with a wider global impact of technology. Learners will take time to reflect on how technology impacts their day-to-day lives. Toward the end of the curriculum packet, learners will finish the “Impact of Technology” lesson and use the “Impact” section as their guide to answer questions on the positive and negative consequences of technology.

Worksheets can be found at the end of each lesson with an answer key for the lesson in the appendix.
Learning Objectives

While your students use Kano, they will be having fun exploring and learning at the same time! Although each lesson has its own set of learning objectives that connects cross-curricularly, the packet has core concepts and objectives embedded throughout. These concepts include:

- Hardware
- Software
- Program Development
- Control
- Algorithms
- Variables

The curriculum objectives include:

- Learners will develop skills to engage in creative coding.
- Learners will be able to articulate the difference between hardware and software.
- Learners will be able to identify programming languages and how they are used on Kano.
- Learners will be able to navigate the Kano computer and understand the components of Kano Apps.
- Learners will be able to work collaboratively on projects.
- Learners will be able to communicate ideas articulately to their peers.
- Learners will be able to identify community needs and think of innovative solutions to address them.
- Learners will be able to use Kano to create innovative projects.
What Is in This Lesson Plan Packet

These lessons are designed to help you and your learners understand computers and coding! Learners will first build their Kanos to gain an appreciation for physical computing and get hyped to explore how computers work! Use these lessons to develop computational thinking methods and learn more about the commonalities amongst programming languages!

You can go in your own order for the lesson plans, but a recommended order is suggested below. We recommend you complete phases 1 and 2 before changing the order.

1. **PHASE 1: INTRO TO CODE**
   - Lesson 1: Physical Computing (Build a Computer Lesson)
   - Lesson 2: What is Code Pt. 1
   - Lesson 3: What is Code Pt. 2
   - Lesson 4: Explore StoryMode

2. **PHASE 2: INTRO TO KANO APPS**
   - Lesson 5: Make Art
   - Lesson 6: Hack Minecraft
   - Lesson 7: Make Snake + Make Pong

3. **PHASE 3: CODING PLUS**
   - Lesson 8: Algorithms
   - Lesson 9: Abstraction
   - Lesson 10: Pattern Recognition
   - Lesson 11: Decomposition
   - Lesson 12.1: Variables
   - Lesson 12.2 Data Types - Numbers
   - Lesson 12.3 Data Types - Strings
   - Lesson 13: Loops
   - Lesson 14.1: Conditionals
   - Lesson 14.2: Conditionals - Booleans
   - Lesson 15: Debugging
   - Lesson 16: Impacts of Technology

4. **PROJECT PACK**
   - Design Thinking
   - Building a Better Community
   - Celebrate Creation!
LESSON 1

Build a Computer!

Beginner 1 hr 10 min

Learners will discuss ahead of time what the components of a computer are and then use that knowledge to BUILD one.

OBJECTIVES

Lightning bolt Learners can describe the components of a computer system, including hardware, operating systems, and applications.

Lightning bolt Learners can give examples of how computers are used in the real world.

Lightning bolt Learners build computers and write lines of code.

STANDARDS

Lightning bolt K12 CS: Computing Systems, Devices, Computing Systems, Hardware and Software

Lightning bolt CSTA: K - 2: 1A-CS-01, 1A-CS-02

Lightning bolt 3 - 5: 1B-CS-01, 1B-CS-02

Lightning bolt Common Core: CCSS.MATH.PRACTICE.MP1

MATERIALS NEEDED

Print out a diagram of the Raspberry Pi and talk through each component.
Introduction

Linking

Today, learners will build a computer! Before they do, challenge them. Ask them to write down as many parts of a computer (mouse, speaker, etc.) as they can in 2 minutes. After the time is up, have them compare with their partner. As a group, share out your answers and the leader should write them down so the class can see the parts. Discuss the difference between hardware and software.

Engage

We were able to name a lot of parts for a computer, and today we will focus on these hardware pieces. Some we always knew, some we just learned. Today, we are going to see how all these parts interact and make a computer. We will have our own components, and by the end of the day we will build, and have a computer that works!

Exploration Activities

Challenge 1: What Can Computers Do?

Ask learners what they think computers can do. As they discuss, write their responses down.

- What are they used for?
- What is code?
- What is a Raspberry Pi?

Challenge 2: Build a Computer!

Pass out the Kano kits. Ask learners whether any of them have built their own computers before to get them excited.

Build! Learners build their Kanos. Clap and cheer as learners get powered up and type “cd rabbithole.”
Challenge 3: Explore Kano

If you have a little time, let the learners play around and move into different apps on Kano. Direct them to Make Art so they can start coding art! If you have a lot of time left to explore then move on to the Explore Kano lesson.

Closing

Evaluation

Once the computers are picked up, ask students to fill out their engineering log. Also, in the in-class reflection section, answer these questions:

- What was your favorite part about today's session?
- What was your least favorite part about today's session?
- What are three things you learned today?
- Why is it important to create, not just consume, technology?

Call on four people to answer one of the four questions above.

Extension

Before everyone leaves, ask the learners to go home tonight and draw or write what it was like to build a Kano Computer. Did they enjoy it? Were they excited? What was the computer like?

Also, pass out the “Build Worksheet” to be completed at home. You can find the worksheet in the appendix section of this packet.

Kano Clean up

As you start using Kano in class, how you clean up will become important. Take a substantial amount of time (10 - 15 minutes) to practice your clean up routine with your students.
Explorer Questions

Directions: Under each image, write the name and what each part is used for on the Kano!

[Image of Kano card and USB cable]

[Image of Kano computer and keyboard]

[Image of Kano microphone and(?)]

[Image of Kano microphone and(?)]
Programmer Questions

In the sections below answer the questions to the best of your ability. If needed, draw examples to support your statements!

1. Today you built and coded your Kano Computer! Can you name the components that were **hardware** of the Kano? Can you name the components that were **software** on the Kano?

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Creator Question

You just built your own computer! Now how about you design your own computer? Below draw the perfect computer! What components would it need to turn on? What would make it unique? Be creative!

Impact Question

How has technology changed in your life? What was it like when you were 5? What was it like 3 months ago? What new technologies have you heard about or seen?

2. While you were using the Kano there were components you used that were inputs (you putting information into the Kano) and there were things that were outputs (the Kano showed you something). What do you think of the Kano hardware was an input device? What do you think was an output device?
LESSON 2

What is Code?

Beginner 1 hr 5 minutes

Students will develop an understanding for what code and programming languages by participating in a hands-on learning experience.

OBJECTIVES

- Learners will be able to explain how computers follow instructions.
- Learners will be able to explain in their own words what code/programming languages are.
- Learners will gain an understanding of the different uses of code.
- Learners will be able to use pseudocode to make basic programs.

STANDARDS

- K12 CS: Computing Systems Devices, Computing Systems Hardware and Software
- CSTA: K-2: 1A-AP-08, 1A-AP-10, 1A-AP-11, 1A-AP-12, 1A-AP-14, 3-5: 1B-AP-11, 1B-AP-15
- Common Core: CCSS.MATH.PRACTICE.MP1
Introduction

Linking
What do you think is the difference between hardware and software? Turn and talk to your neighbor for three minutes.

Remind learners that a way to help you remember the difference is to think of hardware as a physical device. Something you can touch and put in your hands. With software you can’t do the same thing. It lives inside your computer or in another device.

Engage

What do you think programming is? (Below is a possible answer)
When you write instructions or steps for a computer to follow. For example, a program for a wedding or a formal event. It is the outline of the structure of the event or day.

What is a programming language? (Below is a possible answer)
Programming language is the language of computers. It is where we write in an almost human language steps for a computer to follow to do something very specific.

What are some programming languages we already know about? (Below are possible answers)
Scratch  CoffeeScript  Python  Kano Blocks
HTML  JavaScript  CSS

Today, we're going to talk more about what code and programming languages are to get to demystify our computers more!

Collect Kanos

Retrieve Computers, Turn On, Log In
Exploration Activities

Challenge 1: Marching Orders

So we talked about programming and programming languages or code earlier, but what do we know about computers? (3 minutes)
Students may give things that use computers, give programming languages, etc.

Tell students to finish your sentence, “computers are super _____”
Here out answers. When done say “stupid!”
Explain to students that computers need very specific code with details to be able to complete a task. If a door is shut and you tell a human to walkthrough it, they will stop and open the door. A computer or robot wouldn’t do that. If you said go through the door, it would run into it!
So computers need very specific instructions!

Let’s draw a drawing activity together. Think of yourselves as the computer. I will give you very detailed instructions and you will draw whatever I describe! Make sure you listen very carefully!
• Draw a dot in the center of your page.
• Starting at the top left-hand corner of the page rule a straight line through the dot finishing at the bottom right hand corner.
• Starting at the bottom left-hand corner of the page rule a line through the dot, finishing at the top right hand corner.
• Write your name in the triangle in the centre of the left-hand side of the page.

Your image should look like this:

Now take another volunteer from the group. This time another student will give instructions and we will all have to draw the image. The volunteer will only say the instruction two times though, so make sure you are paying attention!
Challenge 2: Program a Human Robot!

Now again, computers follow very detailed instructions and computers are what (stupid) so we need to give a lot of details and instructions. For example, if I told you, a human, to walkthrough a door and you notice it is closed, you will stop to open the door. A robot will full force try to keep going, not trying to open a door because you didn't tell it to!

Now let's see just HOW specific instructions need to be. Split the class into small groups of three students each. Place an obstacle course in front of you (it can be a box in front of you or multiple boxes) and tell the learners we have a challenge today. For this challenge one person in our group will be a “robot.” Our mission is to program one of our teammates to walk around an obstacle course in a very specific way and BLINDFOLDED. To make it harder we have to avoid crashing as much as possible!

Demonstrate the course the robot must walk to complete the exercise. Repeat it a few times so the students understand the start/end points and the direction they need to walk.

All groups will now have 15 minutes to write down the instructions and steps they will give their “robot.” Afterward, there will be trial runs to complete the course. Everyone will get two chances to complete the course. Also, if during a trial the robot “crashes” two times they will have to start over and figure out what went wrong with their code!

NOTE!
- For this activity do not assign the robot. When the learners are ready to try the obstacle course select a random learner. The code should work for all robots!
- Students may struggle. Mention specific details like 90 degrees, small, or large steps, which are helpful to the blindfolded person. The more details the better!
- When students are done, explain computers need very specific details. We may say turn left, but a computer doesn't know what left means. We have to be specific
- There is a cheat in this exercise. We could have said “input intelligence, take off blindfold, complete the course with intelligence!”
Closing

Evaluation

We talked a lot about code and programming languages today. In your engineering logs, can you write down a summary of what we did today and a definition of what coding and programming languages are?

Kano Clean up

Power down and put away Kanos
Name: ____________________________________________ Date: _____________________________

Explorer Questions

Directions: Answer the questions below to the best of your ability. Remember to reference what you spoke about during the session today!

What are some examples of coding languages?

When you did the drawing activity what happened?

When you did the robot activity what happened?

Programmer Questions

Why is it important to be detailed when you give instructions?

If you wanted to draw the image below, what instructions would you give someone? Let's try it in class tomorrow!
Creator Question

Today you had to program one of your team mates like a robot around a simple obstacle course. Why don't you design your own? In the space below draw an obstacle course you would have a robot move around.

Impact Question

Code and technology have changed the way we interact with the world. From the way we communicate with friends and family to the way we travel. Think about your own life. If you could create a new technology to help you in life, what would it be? Why would it be helpful for you? Do you think others would want it and why?
What is Code? Pt. 2

**OBJECTIVES**

- Learners will be able to identify examples of code when they see it in a program
- Learners be able to explain in their own words what code/programming languages are
- Learners will be able to identify components that make up the hardware and software of a computer

**STANDARDS**

- **CSTA:** K-2: 1A-CS-01
- **Common Core:** CCSS.MATH.PRACTICE.MP1
Introduction  

Linking  

We said computers follow very detailed steps. In your notebook can you write down the instructions you would need to give a robot to open a book to a specific page (for example page 10)? Allow learners 5 minutes to write the instructions down. Then let two students share their instructions!

Engage  

Yesterday, we did a lot of talking and a lot of movement to explain code and programming languages. Today, we will be on the Kanos and we want to see when we use code, specific details, and steps to create!

Collect Kanos  

Retrieve Computers, Turn On, Log In

Exploration Activities  

Challenge 1: Kano You Find the Code?  

Before giving out instructions pass out the scavenger hunt worksheet (attached at the end of this document) and ask learners to write their names on it.

Today, we will get to explore the Kano and play around with as many games as we want! However, there is an additional challenge today. We want to be looking for examples of code in the apps. Any time we have to follow instructions or specific steps, we want to make a note of it in our handout.

Let's look at the example. If I was in Hack Minecraft I may have noticed I used a code block to create a loop or you may have noticed if you followed detailed instructions to create that loop. We will be using the rest of this time period to work on the worksheet! It is okay if you don't get to every app, we will be able to talk as a class on where we saw code.
Closing

Evaluation
We did a lot of exploring, so let’s go through each app and share where we saw examples of code!
- Make Light
- Make Snake
- Make Pong
- Hack Minecraft
- Terminal Quest
- Make Art
- Sonic Pi
- Scratch

Now that we have gone over the different types of examples where we saw of code, write down in your logs how each app had similar programming languages. How were they different? Was one way of coding easier than others? Why do you think so?

Kano Clean up
Power down and put away Kanos
Explorer Questions

Directions: For each Kano App identify what type of coding is used. CIRCLE if text code and put in a SQUARE if block code.

Make Light  Make Snake  Make Pong

Hack Minecraft  Terminal Quest

Make Art  Sonic Pi  Scratch

Programmer Question

What do you think is the difference between text based and block based code?
Creator Question

If you had to design a new app for Kano what would it be? What would the app do? How would you code in it?

In the space below:

• Draw the logo for your app
• Explain what you would create in the app
• Tell us what you would use to code in it (text, block)

Impact Question

You are going to go home tonight to create a Kano World account with your parents or guardians. Once you are there you will be able to share your work with other Kano users around the world. How do you think you should interact with them when you receive a comment? Why is it important to be respectful and not to be rude online?
LESSON 4

Explore Story Mode

Beginner 1 hr 5 minutes

Learners will explore Story Mode to learn more about the hardware and software of Kano!

OBJECTIVES

- Learners will be able to give examples of the hardware and software of a computer.
- Learners will be able to explain in their own words how the hardware and software of a computer interact together.
- Learners will be able to identify the different coding environments on the Kano.

STANDARDS

- **CSTA**: K-2: 1A-CS-01, 1A-CS-02, 3-5: 1B-CS-01, 1B-CS-02
- **Common Core**: CCSS.MATH.PRACTICE.MP1
Introduction

Linking

Have you ever been on a quest or adventure before? What was it like? Turn and talk to a partner about an adventure you have been on, read about, or watched before. What did you like about the adventure? What made it fun?

Engage

Like a modern day adventure or quest, we will be playing a game that will let us be explorers today! Working with the Kanos we have really dived into playing with hardware and software. Now we want to further our knowledge. It is one thing to know that the Raspberry Pi is the brain of our computer, it is another thing to understand the specific parts that make up each!

For the rest of this session we will be exploring in an app called Story Mode on the Kano. Has anyone played with Story Mode before and want to tell us what it is? For those who don’t know, Story Mode is a game where you are your profile character and you are walking through the Kano Computer. Each portion of the world has a name that relates to the hardware and/or software of the Kano. We want to explore and learn as much as possible today!

Collect Kanos

Retrieve Computers, Turn On, Log In

Exploration Activities

Challenge 1: Story Mode Scavenger Hunt

Pass out the Story Mode Scavenger Hunt worksheet before giving instructions so the learners have a reference point.

As already said, we will be exploring in Story Mode today! We have worksheet that will help us record our notes and thoughts! One side has a map of the world and the other has the CodeX badges we earn as we explore!
While you explore, make sure you write down what you learn. For example, we start on SD Beach. This is where the operating system is stored, so on my worksheet I would draw an arrow to SD Beach and then write “the operating system lives here!” on the white space around the map.

You have the whole session to explore! It is fine to play but by the end of the session you should have 100% of the world discovered and at least three CodeX discovered!

**Closing**

**Evaluation**

Find another person who you did not work with in Story Mode and see how much you discovered. Did you notice the same things? What did they find that you did not? Try to fill out as much of the world and share your information together.

**Closing**

Tonight try to finish up your worksheet if you did not. Also, in your engineering logs write down what was your favorite part of Story Mode and why!

**Kano Clean up**

Power down and put away Kanos
Explorer Questions

Directions: Explore as much of the world in Story Mode as you can. Your mission is to find all the known areas of Story Mode and briefly explain what type of hardware and/or software lives in the explored area! It is okay if you don't finish today! We will have more days to explore.

In the map, can you name each area? Draw an arrow to a section on the map and in the blank space around the map write the name of that area and a sentence explaining what type of hardware or software lives there!
Identify the CodeX

In each numbered area provide the name of the CodeX and a brief description of what it is!

1. 
2. 
3. 
4. 
5. 
6. 
7. 
8. 
9. 
10. 
11. 
12. 
13. 
14. 
15. 
16. 
17. 
18. 
19. 
20. 
21.
**Programmer Question**

In Story Mode you were able to explore software, hardware, and how they interact with each other. If you had to explain in your own words how hardware and software interact with each other how would you?

---

**Creator Question**

Parts of Story Mode are still unfinished! Look at the map and decide what should go in the unknown area. Below, draw what would be put in the map and label: what you would do in that portion of the world, what codex you could earn, and what Kano apps you would play!

---

**Impact Question**

You should have your Kano World account created. To make one you needed to create a username and password. Why do you think it is important that we do not share our username and password? How can we make sure that our information is secure and we aren’t sharing it with the wrong people?
LESSON 5

Intro to Make Art

Beginner 2 hr 20 mins

Have your creators get used to Make Art and begin creating unique creations to share!

OBJECTIVES

- Learners will understand how to access Make Art
- Learners will be able to identify the features of Make Art and use them to create their own pieces of art

STANDARDS

K12 CS:
- Computing Systems.Troubleshooting;
- Algorithms and Programming.Algorithms
- Algorithms and Programming.Control;
- Algorithms and Programming.Program Development

CSTA:
- K-2: 1A-AP-10, 1A-AP-11, 1A-AP-12, 1A-AP-15
- 3-5: 1B-AP-10, 1B-AP-11, 1B-AP-12, 1B-AP-16, 1B-AP-17
- 6-8: 2-AP-12, 2-AP-13, 2-AP-16
- 9-10: 3A-AP-18

MATERIALS NEEDED

Visit http://art.kano.me/challenges to find all our Make Art challenges
Introduction

Linking
Ask your students to draw anything, using a pen and paper. It can be as simple or as complex as they want but they only have 5 minutes. Put a countdown up to get them energized to go!

Engage
Tell your students to turn and talk to each other for 1 minute each. What is your favorite program you have used to edit images (Paint, Photoshop, Code)? Do they think they can transfer their drawings to the computer with one of those softwares? If you can how? After they are finished with the 2 minute turn and talk discuss how computers are becoming used more often in artwork, from Disney films, to 3D printed jewellery, to laser cut wood prints, to art made from code! Art is changing and using technology to create this change. Today they will learn one platform to use code to create art and share it with the world!

Collect Kanos
Retrieve Computers, Turn On, Log In

Exploration Activities

Challenge 0: Get to know Make Art!
Briefly go over the features of Make Art and how you access the app.

Click this icon to launch Make Art!

Once the app is open they will have the option to work on “challenges” or hang out in the “playground.” Challenges are exactly that, challenges, that the Kano team creates to help creators get used to the environment and start learning code. They are walkthroughs that provide specific instructions to move forward and you can customize your creations Afterward!

The playground is a little different. There are no walkthroughs, it is a playground for your imagination! Take the skills you learned in the walkthroughs or tinker around to see what you can create!
When you open up Make Art in “challenges” or “playground” you will see a screen like the one below. The left side is where you will code, and the right side is where you will see your creations start to grow and be built!

Try hitting enter a few times in the dark left side. You will notice numbers appear. This is a number system to help you create individual lines of code.

Now on the right side move your cursor around. You should notice that in the bottom left corner of the white screen you will see the “x” and the “y” move positions. In Make Art you are creating masterpieces on a grid that corresponds to a coordinate plane. You can use your cursor to help you identify where on the grid you need to draw!

On the left side you will notice seven gray boxes. These are buttons to different windows that give you information on Make Art!

**Code:** This will be where you will usually have your left screen working on. Here you type in the code that will help build your creations on the right side.

**Shapes:** Do you want to create objects? You can create shapes: circle, ellipse, square, rectangle, and polygon in this section with the example code shown.
**Lines:** With this button you can draw lines! These include drawing a line of a certain size, and drawing a line to a certain point.

**Position:** We want to create in different spaces on our digital canvas. To do this we need to learn how to move in Make Art. You can see example to move the cursor a certain distance away and move the cursor to a particular position.

**Text:** Maybe you want to say something in words. Use the text box to learn examples to: write a message, set size and/or font, sets bold text on (true) or off (false), sets italic text on (true) or off (false).

**General:** This button explains a few things that don’t fit in the other categories. Such as: repeat code or get a random number in a range.

**Colors:** What is a piece of art without color? In the color button you will see examples to: set the background color, change the color in use, change the width and color of the stroke (border, set a color’s brightness, set a color’s saturation, rotate a color’s hue angle by a given amount, set how see through a color is.

**Challenge 1: Explore Make Art!**

Have your students click on Make Art and explore the beginner challenges. Don’t forget to tell them to share their art pieces on Kano World!

**Challenge 2: Are your students ready for a challenge?**

Are your students ready for a challenge? Have them open up the playground and attempt the following scenarios:

**Scenario 1:** You coded a Swiss flag, but can you code another flag? See if your students can create a flag themselves! Suggested flags: Japan, Bangladesh, Nigeria, and France.
**Scenario 2:** In the code “your face” we created a smiling face! Can you figure out how to make the face frown?

**Challenge 3: Challenge a classmate!**

Challenge a classmate! Pair up your students. Have each student write a challenge on a piece of paper. Tell them to make it reasonable and something that can be created in Make Art only. Make sure their partner doesn’t see! Once they have their challenge thought out they will exchange with their partner. Can each person finish the challenge before they run out of time?

**Closing**

**Sharing and Evaluation**

Your students should have finished Challenge 3. Ask the challenger to critique the other student’s artwork. Did it make the grade? Ask the challenger to grade them! If they did, give them a 3, if it was okay a 2, if not that much a 1. Ask the challenger to give two lines of feedback, one helpful comment to make it better and one positive comment!

Educator Note: It is recommended you model this to the students to make sure they fully understand the task.

Tell the students you hope they enjoyed this introduction lesson and that they will be doing more exciting work with Make Art. While students leave, tell them to turn in their post it notes with their grade for their challenge partner’s work. Also, have students complete their daily engineering log!

**Kano Clean up**

Power down and put away Kanos
Explorer Questions
Directions: Under each image, write a brief description about each Make Art feature. Think about what you can code with each feature.
Programmer Questions
What does Syntax error mean? How do you fix it?

In the challenge “Sunny Day” you have the completed code below. In the below code, explain which words can be changed to modify the outcome and which things have to stay the same.

Creator Question
You created  on Make Art today, but challenge yourself now! Draw a simple drawing only using 3 shapes (rectangles, circles, and/or triangles). Try to code this drawing tonight or in class later!
LESSON 6

Intro to Hack Minecraft

Beginner  1 hr 10 min

Get introduced to the controls, tools, and workspace to create virtual worlds!

OBJECTIVES

Learners will be able to identify the controls in Minecraft and create simple creations

STANDARDS

K12 CS:
Algorithms and Programming.Control; Algorithms and Programming.Program Development

CSTA:
K-2: 1A-AP-10, 1A-AP-11, 1A-AP-12, 1A-AP-15
3-5: 1B-AP-10, 1B-AP-11, 1B-AP-12, 1B-AP-16, 1B-AP-17
6-8: 2-AP-12, 2-AP-13, 2-AP-16
9-10: 3A-AP-18
Introduction

Warm up
Split students up into pairs and ask them, “If you could live anywhere in the world where would it be? What would it be like? What would the weather be like?” Have students turn and talk to each other to discuss. After the time is up have students share out where they would want to live and what it is like.

Today we are going to learn how to use Minecraft. Minecraft is like a giant sandbox. Like in a sandbox, in Minecraft you can create anything from castles to mermaids. Minecraft is a virtual world where you can use different materials to create landscapes, buildings, and communities.

Your imagination is limitless!

Collect Kanos
Retrieve Computers, Turn On, Log In

Exploration Activities

Challenge 1: Keyboard Shortcuts
Before we start using Minecraft we need to learn about keyboard controls and navigating in Minecraft. As each key is explained, try it out on your Kano!

- w = forward
- s = backward
- a = left
- d = right
- e = inventory. Materials to build with
- numbers 1 - 0 are for choosing different materials
- Move mouse = turn around
- Left click mouse = build (or destroy)
- Space bar = jump (repeated tapping will raise you into the air. Quick double tap while take you back down)
Let's walkthrough accessing a virtual world in Minecraft. First, access Minecraft through the dashboard. Then:

- Click on “Start”
- Click on “Your Origin Story” (it will bring you to a page to select your own challenges)
- The first challenge is called “T is for Tower.” Click on the challenge

Take a bit of time to explore the environment around you, trying out the new controls you learned!

Challenge 2:

The workspace, tools, making floors.

Minecraft is pretty cool, but how do we get to the fun part of making things? In order to create things, we use the “tab” button. Click the “tab” button, and we should see the workspace and tools on the right hand side:
Craft
When we click here we can see blocks that allow us to create. We can create individual blocks, groups, floors, pyramids, even doors!

Text
Maybe you want to add some text to your screen. A set of instructions or action words!

Blocks
Maybe you want to be able to select a block or have a random block appear.

Position
You can use these blocks to designate the position (x, y, and z) of your blocks.

Player
You can also change the position of your player!

Control
Maybe you want to repeat the same code block sequence. Use control blocks to create loops.

Logic
Logic statements are “if this happens, do this” or maybe you are comparing something to make a new creation.

Math
Sometimes we need to use math operations in our code!

Variables
Variables are things we create that will hold specific information we want. It can be an item or it can a list of items. It can even be a sentence or word we type out!

Variables are things we create that will hold specific information we want. It can be an item or it can a list of items. It can even be a sentence or word we type out!
In the “T is for Tower” challenge it asks you to change the glass to TNT. Can you go back into the code space and make 5 TNT towers appear at once? What about 10? 20? 100!? 

**Challenge 3:**

Before we move on to the challenges in Hack Minecraft let’s try creating things JUST with code. To do this we will stay in the T is for Tower challenge and modify the code blocks.

- Press “tab” to enter back into the workspace if not already there.
- Click on “Change Key” and pick a key to start the code.
- Click on “Craft” and drag the “Set Floor” block.
  - Note: This will create a floor and will go one block under the player.
- Click on the single number block and enter “3”.
  - This means you will create a 3 x 3 square for a floor.
- Click on “Type” and select “flowers”.

Your block should look like this:

![Code Block](image)

Click tab to go back into the Minecraft world. Once there, click the letter key you selected to see the floor of flowers appear!

Can you go back into the code space and change the floor size? What about the type of block the floor is made from?
• Have students click the letter key assigned to creating the flower bed.
• Ask students: What do you currently see?
• Now walk for a little and then to press the same letter key again. Ask students again: What do you see?
• Look for the first flower bed. Ask students: Is it still there? What does this mean?
  ◊ There is a correlation between the position of the user to the things you make and that whatever is on the workspace will be made if you click on the letter key.
• If time ask students to find some more space and to change the “type” of block for water, grass, or ice. Make some more floors.

Congratulate your students! You just wrote code to hack Minecraft!!

Challenge 4:
Using blocks and positions

• Now let’s learn some new buttons.
• The small gear on the top right corner can clear the workspace of code blocks which will ensure previous code is not replicated. Press “rese”.
• Now, let’s place a few blocks of grass. We can go into “Craft” and select the “Set Blocks” block that has position and size.
• For this we will want to change the numbers for position and size. Change the numbers to be something between 0 and 5
  ◊ If you leave the position at 0,0,0: when you create your blocks, you will be inside them and it will be dark!
  ◊ If you leave the size at 0,0,0: it will only be one block! We want to create larger structures.
• Change the “Type” block to “Grass”
Your block should look something like this:

Challenge 5:

Try out some of the challenges in "Your Origin Story." Remember to remix the code afterward to make the projects your own!

Closing

Sharing

In their engineering logs have students write down what they have learned. What were some of the key phrases or words we learned (‘blocks, numbers, setters, getters, type, length, width, orientation, etc.’)? What did you like? What did you not like? What was hard? What was easy?

Call students up to the board and have them draw something they want to create next time!

Extension

Have students fill out the “Intro to Hack Minecraft” worksheet that can be found in the appendix!

Kano Clean-up

Power down and put away Kanos
Explorer Questions
Directions: Match the Hack Minecraft block with the category it would live under in Hack Minecraft! Some categories may be used more than once or not at all!

Craft

Blocks

Text

Position

Player

Control

Logic

Math

Variables
Directions: Below are some examples of Hack Minecraft blocks. In your own words, describe what is happening in each image with regards to Hack Minecraft.

1.

2.

3.
Programmer Question

Directions: There are a lot of cool things you can create in Minecraft! Below draw what you would like to build next time on Hack Minecraft and label the drawing explaining what code blocks you would use!
LESSON 7

Intro to Make Snake + Make Pong

Beginner 1 hr

Continuing the process of exploring the Kano, learners will play in Make Snake and Make Pong to get an understanding of what game development is and how they can modify code to make unique games.

OBJECTIVES

- Learners will be able to demonstrate examples of game development in coding
- Learners will gain an understanding of the different uses of code
- Learners will be able to use code to make basic creations

STANDARDS

K12 CS:
- Computing Systems.Troubleshooting
- Algorithms and Programming.Algorithms
- Algorithms and Programming.Control
- Algorithms and Programming.Program Development

CSTA:
- K-2: 1A-AP-10, 1A-AP-11, 1A-AP-12, 1A-AP-15
- 3-5: 1B-AP-10, 1B-AP-11, 1B-AP-12, 1B-AP-16, 1B-AP-17
- 6-8: 2-AP-12, 2-AP-13, 2-AP-16
- 9-10: 3A-AP-18
Introduction

Linking

What is the oldest computer or video game that you can think of? What is the name and when was it created or when was it popular? Think about it for a few seconds and then talk to a partner about it. Did you like this game? What made it fun?

Engage

Today we are going to play with some older video games. The two games we will play with are Pong and Snake. Both of these games were some of the first arcade video games out there! We will play with the original versions and then add on to it to make it more modern, and more importantly, MORE FUN!

Collect Kanos

Retrieve Computers, Turn On, Log In

Exploration Activities

Challenge 1: Our Mission Today

Before we even start the games and challenges we want to have a mission, a goal in mind while we play the games. Both Snake and Pong are older games, but besides being old what are the similarities between them? Why do you think they were popular and why do you think these were some of the first games ever coded? Think of these questions while you play on the apps and record your thoughts in your journal!

Challenge 2: Explore Make Snake

There are a lot of different apps on the Kano Computer. But can you hack a game? Yes! Now, let's go to Make Snake. If your group is young, tell them that snake was a wildly popular game on phones where you were a snake moving around the board eating apples or little dots. It was very basic, but now they get to hack it and make it challenging with code! Don't forget to share your game with Kano World!
Challenge 3: Explore Make Pong

How about one more game? This time let’s hack Pong! Again, this was a very popular game (the first video game) and we can hack it to be AWESOME now! This platform is a little different than Snake. Instead of just using lines of code we can use block code similar to Scratch to customize the game! Don't forget to share your game with Kano World!

Closing

Evaluation

Tell the learners to get into pairs and write in their logs about which game they liked the best. What about our mission for today? Why do you think these were some of the first coded games? What was similar about them?

Kano Clean up

Power down and put away Kanos
Explorer Questions

Directions: Today, we explored Make Pong and used coding blocks to create cool games! Below are some of the blocks found in Make Pong. Can you match them to the correct category they live under in Make Pong? The first one is done as an example.

Start Ups

Colour

Logic

Events

Actions

Numbers

Getters

Setters
Snake + Pong

Explorer Question
Directions: While playing Make Snake, you used terminal commands to play and modify the game. Can you match each command with the explanation of it? The first one is done as an example.

```
--lives enter editor mode
-s (s, m, f) argument to change the number of lives
f shows the help message
-t argument to change the board size
-e argument to change the speed
--help when used with the argument -s will make the speed fast
-b (s, m, l) argument to change the game theme
```

Programmer Question
Below is a sample of code from Make Pong. Can you give a brief explanation on what is happening for each block code?

1. if do
   1.1 Ball size
   1.2 Ball size +
   1.3 Ball size -
   1.4 Ball size 30

2. if do
   2.1 Goal (Either ▼)
   2.2 Ball size 15
Creator Question

Today, you modified and designed games, but what if you had to make your own? Can you make a game that combines the rules for both Make Snake and Make Pong? Draw the game below and write out the rules for the game!

Impact Question

Snake and Pong were some of the first games that came out and it has completely changed our world. Instead of going outside people play video games inside, sometimes for hours! These new technologies, like video games, have an impact on how we socially interact with each other. Thinking of all the technology (social media, video games, TV) that you use in your life. How does that affect the way you communicate with people? How does it affect the way you manage your friendships?
LESSON 8

Algorithms

~ Beginner  

Learners will discuss what algorithms are in computer science and use examples from art to games to demonstrate their understanding.

OBJECTIVES

⚡ Learners will be able to explain in their own words what an algorithm is in computer science

⚡ Learners will be able to remix a program that has an algorithm in it

STANDARDS

⚡ K12 CS:
  Algorithms and Programming.Control; Algorithms and Programming.Program Development

⚡ CSTA:
  K-2: 1A-AP-08, 1A-AP-10, 1A-AP-11, 1A-AP-12, 1A-AP-14
  3-5: 1B-AP-10, 1B-AP-11, 1B-AP-15
  6-8: 2-AP-13
Introduction

Linking
If I asked you how to teach me to write your first name what instructions would you give me? Turn and talk to a neighbor and think about what things you would need to tell me. What materials would I need? What order would I need to write the letters? Where would I write them?

Engage
Like writing your name, computers need really basic instructions to complete things. We as the programmers and coders provide step-by-step instructions for our programs so computers can perform a specific task. An algorithm is a sequence of instructions or a set of rules to get something done.

You probably know the fastest route from school to home, for example, turn left, drive for 5 miles, turn right. You can think of this as an “algorithm” – as a sequence of instructions to get you to your chosen destination. There are plenty of algorithms (i.e. routes) that will accomplish the same goal; in this case, there are even algorithms for working out the shortest or fastest route.

Collect Kanos
Retrieve Computers, Turn On, Log In

Exploration Activities

Challenge 0 : What Is an Algorithm?

Before we look at algorithms in code we need to make sure we understand algorithm as a concept. The definition of an algorithm is:

“An algorithm is a sequence of instructions or a set of rules that are followed to complete a task. This task can be anything, so long as you can give clear instructions for it.” - Code.org

Let’s think about the Marching Order activity we did when we programmed another person.

- Did you follow specific steps to move around the obstacle course (yes)
- What would have happened if you went out of order for those instructions? (we wouldn't finish the course, our robot would have crashed!)
Challenge 1: Why Order Matters

Before we start our challenges today let's look at a challenge in Make Art. We can see why order matters immediately with the first challenge in Basic called “Sunny Day.”

When you type out the code it looks like the following:

Background blue
Color yellow
Circle 150

That is pretty simple code, but what happens when you change the code to the following:
Background blue
Circle 150
Color yellow

What happens? (Get responses from the learners)

You may have noticed that the circle is no longer yellow! Why do you think this happens? (Get responses)

The reason this happens is because of the algorithm. The default color in Make Art is black. We draw the circle before the color is changed to yellow so the circle is drawn as black!

Fix the code so it looks like the original again. Now what happens when you move the first line “background blue?” What happens when you move it? Why do you think this happens? Turn and talk to a neighbor about it!

When programming the algorithm, order, matters!

Challenge 2: Where Does the Order Matter?

What about the rest of Make Art and the code? Let's go into Make Art Pixel Hack challenges and try changing a few things up. In each challenge after you finish it move the code around.

In your engineering logs record:
- How the code changes. Does it change at all?
- Why do you think the code changes or does not change?
Challenge 3 : Create!
Can you code a traffic light like the image below?

Try it out in Make Art. When coding this what kind of algorithm did you create? Did order matter? Did it not matter? After you are done coding the light record your thoughts in your engineering log!

```
moveTo 200,66
rectangle 99,270

moveTo 250,100
color yellow
circle 30

moveTo 250,200
color green
circle 30

moveTo 250,300
color red
circle 30
```

Closing

Evaluation
Turn and talk to your partner and explain in your own words what an algorithm is. Why do you think algorithms are important?

Closing
Tonight in your engineering logs write a definition for algorithm in your own words and give an example in your life where you would use an algorithm.

Kano Clean up
Power down and put away Kanos
The first one has been done as an example.

<table>
<thead>
<tr>
<th>Challenge Name</th>
<th>What Did You Change?</th>
<th>What Did the Change Do and Why?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loot Chest</td>
<td>I moved the location of color gold from line 4 to line 19</td>
<td>This made the bottom shapes black because I didn’t tell the program to change colors before drawing the shape.</td>
</tr>
<tr>
<td>Tetris</td>
<td></td>
<td></td>
</tr>
<tr>
<td>For Loop</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diamond Sword</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Color Frenzy</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Programmer Question
Below is a sample set of code. Can you draw the shape that would be there based on the code from Make Art? Your image does not need to be exact, but as close to the correct dimensions as possible!

```
background blue
color orange
circle 200
color yellow
circle 150
color green
circle 100
color red
circle 50
```

Creator Question
Make Art uses simple shapes and lines to create magnificent art work with code! Why not try to create an art piece yourself? Use simple shapes to create a unique art piece. Afterward, write out the algorithm you would need to create that art piece!
Impact Question

Algorithms are used everywhere. From sports to websites. Some algorithms have even changed the world and the way we interact with it! For example, Google search engines use a specific algorithm to filter websites. Google has changed the way business works! What other technologies that are out there that have changed business? Brainstorm some ideas!
LESSON 9

Abstraction

Beginner 1 hr 30 mins

Learners will discuss what abstraction is in computer science and use examples from the real world to demonstrate their understanding.

OBJECTIVES

- Learners will be able to explain in their own words what abstraction is in computer science.
- Learners will be able to relate abstraction to a real world application of maps to help their learning.

STANDARDS

K12 CS:
- Computing Systems:Troubleshooting
- Algorithms and Programming:Program Development
- Algorithms and Programming:Control

CSTA:
- K-2: 1A-AP-08, 1A-AP-10, 1A-AP-11, 1A-AP-12, 1A-AP-14
- 3-5: 1B-AP-10, 1B-AP-11, 1B-AP-15
- 6-8: 2-AP-13

MATERIALS NEEDED

Introduction

Linking

I have a subway, or underground, map of London for you! Let's go on a bit of a scavenger hunt to see if you can identify a few things! Once you have your map can you:

• Identify how many different colored lines (routes/subway lines) are on the map?
• Can you find Whitechapel?
• Can you find Stepney Green?
• Can you find Angel?
• Can you find Elephant and Castle?
• What lines (routes/subway lines) meet at Bank? (the colors)

Let learners have 5 - 10 minutes to go through the map and work with a partner to find everything. Afterward, collect the answers from the learners.

Engage

Maps are very cool. They help us find things with ease rather than have to search on foot for something! But what is super cool about a map is that you can find the things you want immediately and ignore all the other details. The computer science topic of abstraction is the same way! Abstraction allows us to focus on the big picture, ignoring the things we don't need in order to focus on the things we do!

Collect Kanos

Retrieve Computers, Turn On, Log In

Exploration Activities

Challenge 0 : What Is Abstraction?

The BBC defines abstraction as “Abstraction is the gathering of the general characteristics we need and the filtering out of the details and characteristics that we do not need.”

As said when we looked at the subway map, we see a simpler version of the city that is just how each subway line interacts with each other. Now, let's look at what the London Underground looks like on a real map. Check out this website for a simulation of movement or just this map.

Does the map look the same as the subway map? (Answer NO!)

This happens because when we design maps like ones for public transportation we don't care about all the details! We don't need to know where all the buildings are, where a mountain/hill is, where the line curves a little bit. We just want to know kind of where the lines are (north, south, east, west) and how they connect with each other.
Abstraction is the same way. For example if I was asking you to draw series of dogs. I wouldn’t want all the details about the dog (fur length, length of tail, color of eyes, height). I would just want to know the basic characteristics of a dog (number of legs, eyes, mouth, ears, tail, has fur). With the very basic characteristics, the similarities, I can create a model of a dog!

**Challenge 1: Abstraction of Maps**

Let’s play with the maps a little more. I am going to give you a route (the start and end point) and I need you to figure out the algorithm to get to that point (or the route). If you have a smartphone you can download Citymapper London. If not, work with a partner to see if you can figure out the route I would need to take!

Routes to figure out:
Whitechapel to Angel
Shadwell to Kensington
Notting Hill to Shoreditch High Street

Record your algorithm in your engineering logs!

**Challenge 2: Create Your Own Worlds!**

What if we had to create our own maps? Say for a video game? With a partner design an overworld map for a video game. Remember, this needs to be a general map. I do not need to see all the details, rather than main/general features!

Before the learners start on their own show them a few examples of overworld maps. It is important for them to see that in each map you do not see the specific details of the world (the people living there, the bad guys, or what it looks like) but rather a general route they follow to complete the game!

Yoshi Overworld, Mario Overworld, Zelda Overworld
Closing

Evaluation

If I asked you to create a model of a cat what would it look like? Remember to use abstraction to pick out the general characteristics of a cat and not the specific details! Write down your response and when you are done share your answer with a partner.

Closing

Abstraction allows us to pick out the general characteristics and not get bogged down by all the details! This way of computational thinking allows us to create more abstractly!

Tonight think of ways that you use abstraction at home. Record your responses in your engineering log!

Kano Clean up

Power down and put away Kanos
Explorer Questions

Today, we created an overworld map for a game we would design ourselves. What were some of the things you included in the map? What were some of the things you did not include in the map?

Programmer Question

Abstraction helps us focus on the big picture and not get too caught up in the details. For example, when we created a model of a dog, we focused on the general characteristics of a dog. If you had to create a model of a tiger using abstraction what would it be?
Abstraction

Creator Question
You just created a basic model of a tiger. Can you now draw different variations of that tiger using the model you created? Draw them below!

Impact Question
Abstraction allows us to focus on the big picture rather than the details. Do you think this way of thinking is good or bad? Why?
LESSON 10

Pattern Recognition

Beginner  1 hr 10 minutes

Learners will discuss what patterns are and develop skills to help identify patterns when they see them appear.

OBJECTIVES

- Learners will be able to explain in their own words what patterns are in computer science.
- Learners will be able to understand that identifying patterns allows a programmer to make more efficient code.
- Learners will be able to demonstrate their understanding of pattern recognition by identifying patterns in a program and then brainstorming ideas to make their code more efficient.

STANDARDS

- CSTA:
  - K-2: 1A-AP-10, 1A-AP-11, 1A-AP-12, 1A-AP-15, 1A-DA-07
  - 3-5: 1B-AP-10, 1B-AP-11, 1B-AP-12, 1B-AP-16, 1B-AP-17
  - 6-8: 2-AP-12, 2-AP-13, 2-AP-16
  - 9-10: 3A-AP-18
Introduction

Linking

Before the students come in draw a shape pattern on the board. Have the students guess what the next three images would be for that pattern. For example, you may draw a circle, square, triangle, circle. The next three shapes would be square, triangle and circle following the pattern!

If you have an advanced class you could also use numbers to show a pattern and ask students to figure out what the formula would be to find the next number!

Engage

Have you seen a pattern before? What do you think is a pattern? Allow students to have some time to think about what the definition of a pattern is.

Patterns are everywhere! A pattern is essentially a sequence that is repeated more than once. It could have been something on a shirt, something in math, or maybe you have a routine for getting ready for school.

Many times in life we do the same thing in the same order. Knowing when there is a pattern we can identify it when it occurs and do something to make our lives more efficient. For example, maybe you always brush your teeth in the morning. If that is the case, maybe it would be less time consuming if you placed your toothbrush and toothpaste near the sink rather than moving it somewhere far away?

Or maybe a family memeber needs to pay bills on the 28th of every month. Instead of writing a check each time, maybe they will set up an automatic payment to pay the bills each time. Identifying patterns can help us be more efficient with our time!

Collect Kanos

Retrieve Computers, Turn On, Log In

Exploration Activities

Challenge 1 : Pattern Investigator

Today, we will be looking at some challenges on the Kano, we want to identify when we see a pattern in different code environments.
You will see a few different coding environments (maybe blocks, maybe text based) that will show you code; however, maybe it isn’t the most efficient use of code. Your job is to find the possibilities of when a pattern may be involved. Make sure you record the example of a pattern in your Engineering Logs!

Afterward, you and your partner want to think of how you could make the code more efficient. You may already know the answer, but brainstorm and think of ways you could do less work in your coding space! It is important to record your ideas because at the next session we will see if our ideas would work!

**Challenge 2 : Find the Pattern**

For the rest of the time you will be opening up a few challenges. Go through each one and see if you can find a pattern and think of ways to make it more efficient!

- Make Light: Emoji Challenge
- Make Art: All the Medium Challenges or Pixel Hack

**Challenge 3 : Brainstorm!**

You may have come across it, but brainstorm with your partner on ways to make coding easier when you find a pattern. What kind of code words or blocks were used to make code more efficient on the Kano Apps?

**Closing**

**Evaluation**

Turn and talk to another group and see what you brainstormed. What ideas did they think of to make the code more efficient once a pattern was identified? Was it the same as your idea? Was it something different? Which way seems better?

**Closing**

Tonight try to think of three times you or a family member may use a routine or a pattern. How could you make your life more efficient after identifying the pattern?

**Kano Clean up**

Power down and put away Kanos
Explorer Questions

Directions: Patterns are created with a specific repetition. In one of the Make Art challenges you had the code below:

```
background tan
stroke 1
moveTo 30, 250
color olivedrab
circle 30
for [i..7]
    move 50, 0
color olivedrab
circle 30
square 30
color darkgreen
circle 20
square 20
moveTo 70, 240
color black
ellipse 10, 5
color salmon
move -15, 20
rectangle 20, 4
```

In your own words, can you explain each line of code and why it created the image you see on the right? What if you wanted to change the loop? How could you change it?
**Programmer Question**

In Make Art I was trying to create a grid, so I created individual squares multiple times. Is there an easier way to do this? Can you find the pattern and write the more efficient code that uses less lines of code? Below is my code to make one row of squares.

<table>
<thead>
<tr>
<th>Code to Create One Row</th>
<th>Better Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>stroke black</td>
<td></td>
</tr>
<tr>
<td>color white</td>
<td></td>
</tr>
<tr>
<td>#to create one row of squares</td>
<td></td>
</tr>
<tr>
<td>moveTo 0,1</td>
<td></td>
</tr>
<tr>
<td>square 50</td>
<td></td>
</tr>
<tr>
<td>moveTo 50,1</td>
<td></td>
</tr>
<tr>
<td>square 50</td>
<td></td>
</tr>
<tr>
<td>moveTo 100,1</td>
<td></td>
</tr>
<tr>
<td>square 50</td>
<td></td>
</tr>
<tr>
<td>moveTo 150,1</td>
<td></td>
</tr>
<tr>
<td>square 50</td>
<td></td>
</tr>
<tr>
<td>moveTo 200,1</td>
<td></td>
</tr>
<tr>
<td>square 50</td>
<td></td>
</tr>
<tr>
<td>moveTo 250,1</td>
<td></td>
</tr>
<tr>
<td>square 50</td>
<td></td>
</tr>
<tr>
<td>moveTo 300,1</td>
<td></td>
</tr>
<tr>
<td>square 50</td>
<td></td>
</tr>
<tr>
<td>moveTo 350,1</td>
<td></td>
</tr>
<tr>
<td>square 50</td>
<td></td>
</tr>
<tr>
<td>moveTo 400,1</td>
<td></td>
</tr>
<tr>
<td>square 50</td>
<td></td>
</tr>
<tr>
<td>moveTo 450,1</td>
<td></td>
</tr>
<tr>
<td>square 50</td>
<td></td>
</tr>
</tbody>
</table>
Creator Question
If you had to make a 10 x 10 grid like the image below how do you think you would code it? Think about what you did in the Explorer and Programmer sections.
Impact Question
Pattern recognition allows us to look at data, code, basically anything and find a pattern and then create something to address that pattern. For example, weather technology can collect tons of data, detect a pattern, and then potentially predict when a storm may come into an area. In this case, we are relying on a machine to make predictions for humans based on a pattern observed. Do you think this is a good thing, allowing computers to make big predictions for us? Why or why not?
LESSON 11

Decomposition

Beginner 1 hr 10 mins

Learners will discuss what decomposition is in computer science and breakdown shapes in Make Art to understand how larger parts make up smaller parts.

OBJECTIVES

Learners will be able to explain in their own words what decomposition is in computer science

Learners will be able to show an example of decomposition in Make Art and explain why it can be helpful to use when coding

STANDARDS

K12 CS:
Algorithms and Programming.Control; Algorithms and Programming.Program Development

CSTA:
K-2: 1A-AP-08, 1A-AP-10, 1A-AP-11, 1A-AP-12, 1A-AP-14
3-5: 1B-AP-10, 1B-AP-11, 1B-AP-15
6-8: 2-AP-13
Introduction

Linking
I love pie. Pie is delicious. However, I have no idea what is inside the delicious things. Can you write down all the things that you think are inside a pie?

Engage
Like pies, life is made up of bigger picture things. I may think about a computer, but there are small parts inside that computer that make up a computer. What this is known as in computational thinking is decomposition. Decomposition is the process of breaking a complex problem or idea into smaller parts that are easier to understand. Usually these smaller pieces are easier to solve and understand, so we can work on them individually to create something larger!

For the rest of this session we will be looking at decomposition and then using this method of thinking to help us solve a problem!

Collect Kanos
Retrieve Computers, Turn On, Log In

Exploration Activities

Challenge 1: Create Then Break It Down!
Let’s start at a really basic level. We will create an art piece in Make Art and then identify what those shapes are that make up the art piece. Take 7 minutes to create an original art piece in Make Art! After 7 minutes, switch spots with another group and see if you can identify all the shapes used to make the art piece. They will do the same for you!

For this activity we created a larger art piece. However, we did not just create a picture of a dog with just one movement. We had to use a variety of shapes to create the dog! Even more, we most likely broke down each part of the dog. We had to think individually about what shapes to use to make the dog’s: eyes, mouth, head, ears, mouth, etc.
Challenge 2: Create a Game!

We played with Make Art to show the breakdown of a larger piece into a smaller one. Now let's think about games! Open up Make Pong and go into playground mode. For this part of the session I am going to give you a very general task. Your job is to work with your partner and create the game!

Your mission, if you choose to accept it is this: In Make Pong create a 2 player game that lets you win the game after scoring 10 points. Also, add two unique extra features to the game. I will not tell you what these two unique features are, you decide!

For this challenge you are given the bigger picture of creating a game, but try to use decomposition to help you build it! Think about all the small pieces of the game you need to create to make THE GAME.

Closing

Evaluation

Go to someone else's Kano and play their game. Does the game meet the requirements set? How do you think they used computational thinking to create this game? Be prepared to discuss your thoughts with the group!

Kano Clean up

Power down and put away Kanos
Explorer Questions

Directions: Today we created a game in Make Pong. Our task was to make a game that:

- Was a 2-player game
- Lets you win the game after scoring 10 points
- Had two unique extra features to the game

How did you decompose this game into smaller tasks? Use the space below to explain how you went about creating the code!

Programmer Question

Decomposition helps us in a variety of tasks and ways! Now what about in solving a crime? How would you decompose the task of solving a crime?
Decomposition

Creator Question
Decomposition helps us in a variety of tasks and ways! Now what about in solving a crime? Can you create a picture and decompose it to its basic shapes? Draw the picture below and then below it draw all the individual shapes that make the picture!

Impact Question
Decomposition allows us to break down a complex task into simpler steps. This helps us not get confused and focus on the immediate needs. Do you think this way of thinking is good or bad? Why?
Lesson 12.1

Variables

Beginner 1 hr

Learners will discuss what variables are in computer science and use examples from art to games to demonstrate their understanding.

Objectives

- Learners will be able to explain in their own words what a variable is in computer science.
- Learners will be able to create a program that uses at least one variable.

Standards

K12 CS:
- Algorithms and Programming.Algorithms
- Algorithms and Programming.Control
- Algorithms and Programming.Program Development

CSTA:
- K-2: 1A-AP-10, 1A-AP-11, 1A-AP-12, 1A-AP-15
- 3-5: 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-12, 1B-AP-16, 1B-AP-17
- 6-8: 2-AP-12, 2-AP-13, 2-AP-16
- 9-10: 3A-AP-18
**Introduction**

**Linking**

Have you ever heard of a variable before? If so in what context? If you haven't heard of the word before what do you think it would be?

**Engage**

Life is full of unknowns. We don't always know the answer to everything (it is impossible!) so our brains do some cool things to help us. For example, I can look at this room, but I may not know on the top of my head what the height of the room is. In my mind I just think of it as “height of the room.” For now I can just make a guess and say the room is 3 meters high. Later I can get a ruler and measure the room to discover the real height of the room. In computer science when you create something to hold the place of something you may not already know, you are creating a variable. So in this scenario “height of the room” or simply “height” is my variable. Today we will play with variables and see how they are used in different environments on the Kano!

**Collect Kanos**

Retrieve Computers, Turn On, Log In

**Exploration Activities**

**Challenge 0: What Are Variables??**

Let's make sure we really understand variables before we start to code with them. Again, we can create variables to hold information. It can hold numbers, words, or even lists! Probably the most common place you will find variables, outside of coding, is in math. In math, variables are used to hold the place for an unknown number. For example, I can say $2 + x = 5$. In this case $x$ is a variable holding the place for an unknown number. Now, what __ plus 2 will equal 5 (answer: 3)? Now we have solved for the variable, so I would say $x = 3$! Computers can do the same thing!

Another way you can use variables is in games. Say we are all playing a game right now. We want to keep score. At the start of the game your score is zero because you have not earned any points. As you earn points the score changes. You can think of score as a variable. It is holding the value of how many points you earn and changes!
Challenge 1: Define Yourself with Variables!

Again, variables hold the place for data we can get later. So maybe let’s try it out! I am a person and there are a lot of things that make me who I am! I know what these things are, but maybe you don’t. Let’s do a few together:

What is my name? Students will call out your name
Let’s write it on the board like this name = "[insert name]"

Now what about a few other things that describe me? What is my eye color? What is my age? What is my height? Am I a student? Am I a male? Let students call out the answers. Record their responses as follows:

eyecolor = "[insert eye color]"
age =
height =
amstudent = true/false
male = true/false

For some of the responses, the answer may be unknown so leave it blank! As a note on syntax, anything that is letters like color, name - place the word in “ ” and if it is a number just leave it as a number.

What are some other characteristics about myself that we could try to define? Let students call out maybe five other descriptions and write them on the board and have them guess.

All these different descriptors can be viewed as a variable. We defined a variable (name, eyecolor, amstudent) as something we didn’t know and then were later able to give those variables information.

Are all the answers for the variables the same? Were the answers all letters or numbers? No, you had true/false statements, numbers, and words

When we code we can have variables be defined in different ways. We can have variables that hold number values, letter values (known as strings), and true/false statements (known as Booleans). Variables do this so code can perform complex tasks like video games!

Video games are fun because we can earn things and progress through them after completing challenges. Well, how does the game know you complete a challenge? Through variables!
Take this scenario for example. I collect coins in a game. Maybe when I collect so many coins I get a key to a door. Say this number is 20 coins. I could have a variable called totalcoins. When totalcoins = 20 I get a key! Pretty cool right?

We will talk more about the specific types of variables we briefly mentioned (strings, numbers, Booleans) in the following sessions, but let’s explore variables as we understand them now on the Kano!

**Challenge 1: Variables + Art**

Open up Make Art and go to Pixel Hack. Open up the challenge called “variables.” It should have an image of a red ghost. If you are unable to click on the challenge it means you will need to code the challenges before it (Pong, Asteroid Ship, Tetris, and Loot Chest). In this challenge you will see an example of a variable being used. When you complete the challenge write down when you used a variable. What was the variable and how did you use it?

**Challenge 2: Variables + Games**

We saw a use of a variable in Make Art, but what about another app? Let’s open up Make Pong this time! We specifically want to look at Challenge 5: Crazy Ball. If you are unable to click on Challenge 5, this means you will need to complete challenges 1 - 4 before. Like the Make Art Challenge, complete the challenge and in your journal explain how a variable was used. What was the variable and how did you use it?

---

**Closing**

**Evaluation**

Turn and talk to your partner and explain how you used variables today. Did you create a game where you needed to use a variable? Did using a variable make your game or program more fun? Why or why not?

**Closing**

Tonight in your engineering logs write down in your own words what a variable is and give an example of when you would use a loop in real life. Be prepared to share with us tomorrow!

**Kano Clean up**

Power down and put away Kanos
Explorer Questions

**Directions:** You just completed a series of challenges where you created and modified variables. In the sections below you have the code from each of those challenges. Can you circle the variables in the code?

### Make Art Challenge: Variables

```plaintext
background navy
stroke 0
color red
ghostsize = 100
move 0, -50
circle ghostsize
move -ghostsize, 0
square ghostsize * 2
move ghostsize / 2
color white
circle 20
color blue
circle 10
move ghostsize, 0
color white
circle 20
color blue
circle 10
```

### Make Pong Challenge: Crazy Ball

- **Board**
- **Lines Colour**
- **Scene**
- **Forest**
- **Ball**
  - **Size**
  - **Speed**
  - **Style**
  - **Colour**

- **Ball size**
- **Ball position**
In Make Art and Make Pong you used variables but for different reasons. For each code set, can you explain what the variable was used for?

<table>
<thead>
<tr>
<th>Code Section</th>
<th>What was the variable used for?</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>background navy</code></td>
<td></td>
</tr>
<tr>
<td><code>stroke 0</code></td>
<td></td>
</tr>
<tr>
<td><code>color red</code></td>
<td></td>
</tr>
<tr>
<td><code>ghostsize = 100</code></td>
<td></td>
</tr>
<tr>
<td><code>move 0, -50</code></td>
<td></td>
</tr>
<tr>
<td><code>circle ghostsize</code></td>
<td></td>
</tr>
<tr>
<td><code>move -ghostsize, 0</code></td>
<td></td>
</tr>
<tr>
<td><code>square ghostsize * 2</code></td>
<td></td>
</tr>
<tr>
<td><code>move ghostsize / 2</code></td>
<td></td>
</tr>
<tr>
<td>Ball size</td>
<td></td>
</tr>
<tr>
<td>Ball position</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
</tr>
<tr>
<td>▼</td>
<td></td>
</tr>
</tbody>
</table>


**Creator Question**
The last challenge for the variables lesson was to create a game in Scratch that kept score. Can you draw your game below, identify how you would code each part of the game, and identify the variables you used for the game?

**Impact Question**
Variables are super cool and allow your programs to hold all kinds of information. It isn't just information that can be held from your own language, it can be any language or information from anyone, anywhere! If you built a website that anyone around the world could use what kind of website would it be? What languages would it be in? Would people from all over the world be able to communicate on it?
**LESSON 12.2**

**Data Types: Numbers**

*Beginner*  
*1 hr*

Learners will discuss number data, identify when it is used in code, and create their own coded pieces using numbers.

**OBJECTIVES**

- Learners will be able to explain in their own words what a variable is in computer science.
- Learners will be able to understand that variables can hold data, specifically number values.
- Learners will be able to create a program that uses at least one variable.

**STANDARDS**

**K12 CS:**
- Computing Systems.Troubleshooting;
- Algorithms and Programming.Algorithms
- Algorithms and Programming.Control;
- Algorithms and Programming.Program Development

**CSTA:**
- K-2: 1A-AP-10, 1A-AP-11, 1A-AP-12, 1A-AP-15
- 3-5: 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-12, 1B-AP-16, 1B-AP-17
- 6-8: 2-AP-12, 2-AP-13, 2-AP-16
- 9-10: 3A-AP-18
**Introduction**

**Linking**

Turn and talk to your partner and go over what a variable is again. What is a definition we could create for a variable? What kind of things can a variable hold?

**Engage**

Yesterday we did an activity to understand what a variable is, but now we want to focus on the different data a variable can represent. Today we will talk specifically about number data types. We use number a lot, and you will see them in a lot of different coding environments. Today I think of yourself as an investigator looking for numbers and variables in as many places as you can!

**Collect Kanos**

Retrieve Computers, Turn On, Log In

**Exploration Activities**

**Challenge 0: Variable Investigator**

For the next three sessions we will be playing a few activities on the Kano in relation to a specific computational thinking category we talked about. This means that while we are completing the challenges you will be actively looking for examples of something. When you find an example, you will record it in your engineering log and explain when it was used/what it did.

For example, right now we are looking for examples of a variable that uses numbers. If I find one in Make Art, I would write in my log the name of the variable then how it was used. In this example it was a variable that was "height = 200" so I would write it was a variable that was the height of the shape being made.

You want to look very carefully because it may be hidden! It is up to you and your partner to find as many examples as you can!
Challenge 1: Number Variables on the Kano

On the board, I will list a Kano App and the challenge name. Your job is to go into these challenges and see if you can find an example of a number variable. There may be more than one example! It may be hard to find! You need to think critically with your partner to find it! Also, remember to record your findings in your engineering logs!

Let's look at the first one together. Can you open up Hack Minecraft and complete the “Treemendous” challenge. Once you finish turn and talk to a partner. Can you see where the variable is? Get student responses.

If I look at the code blocks we can see a block that says set item. This means we are creating a variable called item and we are assigning a number value of 3!

Now that I found my number variable I would write in my engineering log:
“In Hack Minecraft- Treemendous challenge the set item is a variable block that can be used to hold the values of numbers. I can even draw a picture of the block if I want!

Now why don't you try the last ones out yourself?

- Make Pong: Challenge 6, 11
- Make Art: Summer Camp Challenge - Table Tennis, Tree

Closing

Evaluation

Turn and talk to another group and compare the variable examples that you found. Did you find the same amounts? Was it more or less?

Closing

Tonight try to think of five times you may use a variable to store a number in your life!

Kano Clean up

Power down and put away Kanos
Variable Questions

Directions:

You just completed a series of challenges where you created and modified variables. In the sections below you have the code from each of those challenges. Can you circle the variables in the code?

Hack Minecraft Challenge: Treemendous

Make Pong Challenge: Make It Noisy
Programmer Question

In Make Art and Make Pong you used variables but for different reasons. For each code set, can you explain what the variable was used for?

<table>
<thead>
<tr>
<th>Code Section</th>
<th>What was the variable used for?</th>
</tr>
</thead>
</table>
| x = random 40, 460  
y = random 150, 250  
moveto x, y | |
| Ball  
Size 90  
Speed 90  
Style Colour ▼ | |
Creator Question
If you had to create a website that would tell you about the weather what kind of variables do you think you would need? For example, you may need one variable that stores the temperature (number). Write below all the variables you would need for a website that would tell someone what the weather was and what kind of clothes they should wear for that weather!

Impact Question
You learned that variables can hold numbers and data for later use. You may have a complex topic be stored as a number. For example, maybe you have a variable that says number of females in technology is equal to 10. Although the number represents a specific thing (the number of women in a tech company) it can sometimes simplify a complex issue into just a statistic. What can we do to make sure that we are showing more than just numbers when we look at data?
LESSON 12.3

Data Types : Strings

Beginner 1 hr

Learners will discuss what a string data type is, identify it in code, and create their own programs using strings.

OBJECTIVES

- Learners will be able to explain in their own words what a string is in computer science.
- Learners will be able to understand that string variables can hold letters, words, and phrases.
- Learners will be able to demonstrate their understanding of strings by identifying string data types in an example set of code.

STANDARDS

K12 CS:
- Computing Systems.Troubleshooting;
- Algorithms and Programming.Algorithms;
- Algorithms and Programming.Control;
- Algorithms and Programming.Program Development;

CSTA:
- K-2: 1A-AP-10, 1A-AP-11, 1A-AP-12, 1A-AP-15
- 3-5: 1B-AP-09, 1B-AP-10, 1B-AP-11, 1B-AP-12, 1B-AP-15, 1B-AP-16, 1B-AP-17
- 6-8: 2-AP-12, 2-AP-13, 2-AP-16
- 9-10: 3A-AP-18
Introduction

Linking

We have been talking a lot about variables and what they can do and what they represent. Now let me ask a question, if I said create a variable to hold your name, what types of data do you think your variable would hold? Talk to your partner about this?

Students may get confused. If they do, mention should your name be represented in numbers? Maybe it should be represented in binary? How can a variable hold letters? Technically, we have only talked about numbers!

Engage

In the example of a variable storing your name we can not limit ourselves to just numbers. Technology can do more than that! Variables can actually hold data that is represented in the form of letters, words, and phrases. When variables do this, we call it a “String.” A string is essentially anything that is represented by letters.

For example: “A”, “My name is Taylor.” “FAQ”

All the combinations above use letters of some sort to represent them! Today we will be exploring ways the Kano uses strings!

Collect Kanos

Retrieve Computers, Turn On, Log In

Exploration Activities

Challenge 0: Variable Investigator

Like the last session, we want to put our investigation hat on! While you are completing the challenges, you will actively look for examples of variables that use strings. When you find an example, you will record it in your engineering log and explain when it was used/what it did.

For example, right now we are looking for examples of a variable that uses strings. If I find one in Make Art, I would write in my journal the name of the variable then how it was used. In this example it was a variable that was “name = ‘Ms. Taylor’” so I would write it was a variable that was holding the value of a name to be used later for a text display.
You want to look very carefully because it maybe be hidden! It is up to you and your partner to find as many examples as you can!

**Challenge 1: Find the String Challenge**

On the board I will write a Kano App and the challenge name. Your job is to go into these challenges and see if you can find an example of a string. There may be more than one example! It may be hard to find! You need to think critically with your partner to find it! Don’t forget to record your findings in your engineering logs!

Let’s look at the first one together. Can you open up Hack Minecraft and complete the “Namify” challenge. Once you finish turn and talk to a partner. Can you see where the variable is? Get student responses.

If I look at the code blocks we can see a block that has " " in it. This means that the block is holding the place for a string! What did you put in it?

Now that I’ve found my string variable I would write in my engineering log:

“In Hack Minecraft- Namify challenge the block “ “ holds the value for strings (words or sentences)”

I can even draw a picture of the block if I want!

Now why don’t you try the last ones out yourself?

- Make Pong: Challenge 6
- Make Light: Challenge 9 (Scrolling Light)
- Make Art: Summer Camp Challenge - Hiking

**Challenge 2: Create Your Own Strings!**

You came and you conquered! Now why not create? You saw how different applications on the Kano used strings to show different outputs, now why don’t do something yourself? Pick an application from the ones you played on and create something that uses two examples of a string. It can be anything! Once you are done, record in your engineering logs what you created! If you finish early, show a neighboring group what you created and see how you can modify it!
Evaluation
We talked a lot about variables, how they can hold numbers and strings, but what is the point? Turn and talk to your partner and explain why it would be helpful to have a variable store multiple types of data!

Closing
Tonight try to think of five times you may use a variable to store a string in your life!

Kano Clean up
Power down and put away Kanos
Name: ____________________________ Date: ____________________________

Explorer Questions

Directions: You just completed a series of challenges where you created and modified variables. In the sections below, you have the code from each of those challenges. Can you circle the variables in the code?

Make Art Challenge: Hiking

```javascript
stroke 0
color (opacity white, 1)
for j in [0 ... 250]
x = random 0, 500
y = random 300, 500
moveTo x, y
circle 100
moveTo 250, 350
color red
bold true
italic true
font 'Bariol', 40
text 'Hike the Blue Mountains of'
font 90
move 0, 90
text 'Camp Kano!'"
Programmer Question

In Make Art and Make Light, you used variables but for different reasons. For each code set, can you explain what the variable was used for?

<table>
<thead>
<tr>
<th>Code Section</th>
<th>What was the variable used for?</th>
</tr>
</thead>
<tbody>
<tr>
<td>color red</td>
<td></td>
</tr>
<tr>
<td>bold true</td>
<td></td>
</tr>
<tr>
<td>italic true</td>
<td></td>
</tr>
<tr>
<td>font 'Bariol', 40</td>
<td></td>
</tr>
<tr>
<td>text 'Hike the Blue Mountains of'</td>
<td></td>
</tr>
<tr>
<td>font 90</td>
<td></td>
</tr>
<tr>
<td>move 0, 90</td>
<td></td>
</tr>
<tr>
<td>text 'Camp Kano!'</td>
<td></td>
</tr>
<tr>
<td>mytime = timenow()</td>
<td></td>
</tr>
<tr>
<td>message = 'Time : ' +mytime</td>
<td></td>
</tr>
<tr>
<td>light.scroll(message)</td>
<td></td>
</tr>
</tbody>
</table>
Variables: Strings

Creator Question
The last challenge for the variables lesson was to modify all the challenges with different strings. What did you change these strings to? Below, draw an example of the code you changed and how you did it. If you could hack something else with the same code what would you add?

Impact Question
We can store letters, words, and even sentences in string variables. Even in different languages! This allows programs and websites to more accessible to a larger audience. In your opinion, what are other ways we can make technology more accessible to everyone? More accessible to people in other countries? More accessible to people with physical disabilities?
**Loops**

Learners will discuss what loops are in computer science and use examples from art to games to demonstrate their understanding.

**OBJECTIVES**

- Learners will be able to explain in their own words what a loop is in computer science.
- Learners will be able to create a program that uses at least one loop.

**STANDARDS**

- **K12 CS:**
  - Computing Systems.Troubleshooting;
  - Algorithms and Programming.Algorithms
  - Algorithms and Programming.Control;
  - Algorithms and Programming.Program Development

- **CSTA:**
  - K-2: 1A-AP-10, 1A-AP-11, 1A-AP-12, 1A-AP-15
  - 3-5: 1B-AP-10, 1B-AP-11, 1B-AP-12, 1B-AP-12, 1B-AP-16, 1B-AP-17
  - 6-8: 2-AP-12, 2-AP-13, 2-AP-16
  - 9-10: 3A-AP-18
**Introduction**

**Linking**

I am writing the number 1 on the board. I want to add 1 to it. What is my new number? (Answer 2). What if I add 1 to that, what is my new number? (Answer 3) What if I add 1 to that, what is my new number? (Answer 4) What if I add 1 to that, what is my new number? (Answer 5)

**Engage**

I could keep doing this for hours, even days, but it would waste a lot of my time and energy for other things I want to do (like play soccer). This is where computers come in! Computers and programming have evolved from simple math machines to complex devices that can do tasks that would take humans days to do! Instead of me writing the answer to the earlier activity over and over again, I could create what is known as a loop to add 1 to the original equation for a few more times or even FOREVER. Loops are a powerful part of coding and we will explore it more today!

**Collect Kanos**

Retrieve Computers, Turn On, Log In

**Exploration Activities**

**Challenge 0: What Are Loops?**

Before starting, what is a loop? “The action of doing something over and over again”

We can also think of a loop repeating an action or set of instructions until a condition is met.

Loops in code are pretty cool because we may want something to repeat until a condition is met (say play a song on repeat until the user says stop) or maybe the loop will not start until a specific condition is met (read the data, then do a loop analyzing it). Or maybe we want a FOREVER loop, something that will repeat forever unless told to stop.

Loops allow us to change the game. We see loops most often in applications, games, and computers because humans just simply can’t repeat the same steps over and over infinitely without having to rest or eat!
We can play a game of “Simon Says” to illustrate this! When I say “Simon Says” you must do that action. You have to continue that action until I say stop for that action. For example, if I say “Simon Says jump up and down,” you will need to jump up and down until I say again “Simon Say stop jumping.”

Let’s try it out,
“Simon says jump” (Students should be jumping)
“Simon says clap your hands” (Students should be jumping AND clapping their hands)
“Simon says stop!”

How did you do with the last instruction? Sometimes we are given a loop and then ANOTHER one. It happens in life and in programming!

At this point, play the game for another 5 minutes. If you would like to have another student be the leader as well try it out!

For the rest of the day we will be playing in a few different Kano applications to see how we use loops. While we go through each app we should make a note in our engineering logs where we used a loop and what it did in the application!

**Challenge 1 : Loops in Minecraft**

For our first challenge we want to open up Hack Minecraft. We want to look at the challenge called “Whammy Walk” specifically. Complete the challenge and then change a few things with it. Again, we want to be writing in our engineering logs how we used a loop and what it is for!

After you finish the challenge, try to create a loop of your own for a new power in Minecraft!

**Challenge 2 : Loops in Make Art**

We saw how we can use loops in an interactive game like Minecraft, but what about in art? Open up Make Art and specifically look at the Medium Challenges in Make Art. Complete as many as you can and follow the same process you went through for Hack Minecraft. In a challenge, remix it and change it, but also think about and recording in your logs how you used loops to create cool new art pieces!
Closing

Evaluation

Turn and talk to your partner and share how you used a loop. Did you use it in Minecraft? Did you use it in Make Art or another app? Why do you think loops are used in computer science and games?

Closing

Tonight, in your engineering logs write a definition for a loop in your own words and give an examples in your life where you have seen a loop being used.

Kano Clean up

Power down and put away Kanos
Explorer Questions

Directions: You just completed a series of challenges where you created and modified loops. In the sections below, you have the code from each of those challenges. Can you circle the loops in the code?

Hack Minecraft Challenge: Whammy Walk

![Code for Hack Minecraft Challenge]

Make Art Challenge: Rainbow Gradient

```
for x in [0 .. 25]
    for y in [0 .. 25]
        color rotate red, 10 * x + 10 * y
        moveTo 20 * x, 20 * y
        square 20
```
In Make Art, you saw a lot of examples of loops. For each code set, can you explain what the loop was used for?

<table>
<thead>
<tr>
<th>Code Section</th>
<th>What was the loop used for?</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>stroke gray, 3</code></td>
<td></td>
</tr>
<tr>
<td><code>color null</code></td>
<td></td>
</tr>
<tr>
<td><code>for i in [0..32]</code></td>
<td></td>
</tr>
<tr>
<td><code>circle 10 * i</code></td>
<td></td>
</tr>
<tr>
<td><code>for x in [0..25]</code></td>
<td></td>
</tr>
<tr>
<td><code>for y in [0..25]</code></td>
<td></td>
</tr>
<tr>
<td><code>color rotate red, 10 * x + 10 * y</code></td>
<td></td>
</tr>
<tr>
<td><code>moveTo 20 * x, 20 * y</code></td>
<td></td>
</tr>
<tr>
<td><code>square 20</code></td>
<td></td>
</tr>
</tbody>
</table>
Creator Question
The last challenge for the variables lesson was to modify all the challenges with different strings. Instead of having to type something over and over again, a loop can help us write less lines of code! If you had to create a new piece of art on Make Art using at least one loop what would it be? Draw an example of the artwork you would create below and label what parts would need a loop.

Impact Question
Technology can do things over and over again without stopping! That is a cool thing that computers can just do! If there was a task in your life that you do every day that you could create a program/robot/or other tech to do every day for you, what would it be? Is there something your school does every day that you could create a program for?
LESSON 14.1

Conditionals

Beginner 1 hr

Learners will discuss what conditionals are in computer science and use examples from art to games to demonstrate their understanding.

OBJECTIVES

🔥 Learners will be able to explain in their own words what a conditional is in computer science

🔥 Learners will be able to create a program that uses at least one conditional

STANDARDS

🔥 K12 CS:
  Algorithms and Programming.Control; Algorithms and Programming.Program Development

🔥 CSTA:
  K-2: 1A-AP-10, 1A-AP-11, 1A-AP-12, 1A-AP-15
  3-5: 1B-AP-10, 1B-AP-11, 1B-AP-12, 1B-AP-12, 1B-AP-16, 1B-AP-17
  6-8: 2-AP-12, 2-AP-13, 2-AP-16
  9-10: 3A-AP-18
Introduction

Linking

If I told you to walkthrough a door and it was closed, what would you do? Hear responses from the students. Maybe some of them say they will keep walking, some may say they will open the door, some may even say they won’t do the activity at all. Have students turn and talk to a partner and explain why they did or didn’t continue walking through the door when they observed it was closed.

Engage

Just like in the activity Marching Orders, if we had our robot hit a wall or a barrier we had to think of the next step or a condition to move the “robot” or reset the robot so it could continue the course. We do this all the time in our brains. We want to do something, like walkthrough a door, so we try the activity out. However, some conditions apply. If the door is open we walkthrough it, else we stop. This process we do in our brains every day is known as a conditional! Today, we are going to talk more about what a conditional statement is in computer science and play a few games on the Kano to see it illustrated in real life!

Collect Kanos
Retrieval Computers, Turn On, Log In

Exploration Activities

Challenge 0: What Are Conditionals?
What is a conditional statement? Well, the easiest way to think of it is:

“Statements that only run under certain conditions”
What this means is when we give a task or a job to a program it will only complete that task under a certain condition being true. If it is false, then it will not complete it.

So...

If this happens, do this --> else, if this happens do this

if (a condition evaluates to True):
then do these things only for “True”
else:
otherwise do these things only for “False.”
Let’s look at an example together in a card game. In this game if you draw a red card your team receive a point. If you do not draw a red card then the other team receives a point. We can think of it like this:

If (card == RED)
Teampoint = teampoint + 1
Else
   Otherteampoint = otherteampoint + 1

In this case we used == to equal true! Meaning, if this statement is true follow the below instructions. Else, go on to the next line of instructions.

This is just one example of a condition. We could add a lot more to this game to make it more complex and interesting, so why don’t we! Sit with your partner and design two more rules for the card game. Write your ideas down in your engineering log. Use the deck of cards to practice the game using the 2 new conditions you have created for the game!

Give students about 10 - 15 minutes to develop rules and play them. If they create rules quickly, then they can combine their rules with another pair to make a more complex game!

Now that we have taken some time to really play with conditionals, let’s see a few more examples of Conditional Statements on the Kano!

**Challenge 1 : Play with Conditionals!**

We will be doing a lot of game playing on the Kano today so we want to open up Make Pong. In Make Pong starting at challenge 7 you will see yourself using the conditional block a lot. While you go through each challenge write in your engineering logs how you used the conditional block. What would happen if you didn’t have the conditional block?

Levels to focus on:
Pong: Level 7 (instant rebound), Level 8 (instant win), Level 9 (Disco Pong), Level 10 (10 points to win), Level 11 (make it noisy), Level 12, Level 13 (Paddle Shrink), Level 14 (Turbo Mode), Level 15 (Monster Ball), Level 16 (add a second player)
**Challenge 2: Create Your Own!**

At this point you have gone through all the challenges on Make Pong. How about making your own game? Create your own version of Pong and do (at the minimum) the following:

- Change your background
- Give your game a title
- Create at least five conditions in your game!

If we have extra time, you can switch spots with a neighbor and play their version of Pong!

Remember to save your game because we will still use it for the next session!

---

**Closing**

**Evaluation**

Turn and talk to your partner and share how you used a conditional. Did you use it in Scratch? Did you use it in Make Pong? Why do you think conditional statements are used so much in games?

**Closing**

Tonight in your engineering logs write a definition for a conditional statement in your own words and give an examples in your life where you would use an algorithm.

---

**Kano Clean up**

Power down and put away Kanos
Explorer Questions

Directions: During the card game you had to create conditions or rules to win the game/get points. What were some of the rules you came up with?

While you were exploring in Make Pong you ended up seeing a lot of different conditional statements. Below, write what conditional statements you created in each challenge? The first one has been done for you.

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Conditional Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 7</td>
<td>If the W key is pressed the ball is instantly rebounded</td>
</tr>
<tr>
<td>Level 8</td>
<td></td>
</tr>
<tr>
<td>Level 9</td>
<td></td>
</tr>
<tr>
<td>Level 10</td>
<td></td>
</tr>
<tr>
<td>Level 11</td>
<td></td>
</tr>
<tr>
<td>Level 12</td>
<td></td>
</tr>
<tr>
<td>Level 13</td>
<td></td>
</tr>
<tr>
<td>Level 14</td>
<td></td>
</tr>
<tr>
<td>Level 15</td>
<td></td>
</tr>
<tr>
<td>Level 16</td>
<td></td>
</tr>
</tbody>
</table>
Conditionals

Programmer Question

1. Can you fill in the blanks with a condition if you draw a red card?

If (card == RED)
#Write below what will happen if the card is red!

Else
#Write below what will happen if the card is black!

2. Can you create your own conditions for a card game? Use the space below to add your own conditions!

If ( )
#Write below what will happen if the card IS met!

Else
#Write below what will happen if the above statement is NOT met!

If ( )
#Write below what will happen if the card IS met!

Else
#Write below what will happen if the above statement is NOT met!
Conditionals

Creator Question
The last challenge for the variables lesson was to modify all the challenges with different strings. Instead of having to type something over and over again, a loop can help us write less lines of code! If you had to create a new piece of art on Make Art using at least one loop, what would it be? Draw an example of the artwork you would create below and label what parts would need a loop.

Condition 1

Condition 2

Condition 3

New Condition

Impact Question
The use of conditional statements allows us to create smarter programs. Artificial intelligence is an example of a smarter programming. Artificial intelligence is being used on websites, robots, and so much more! What do you know about artificial intelligence? What do you think the next 5 years will look like in regards to the development of artificial intelligence?
LESSON 14.2

Conditionals : Booleans

俵 Beginner 1 hr

Learners will discuss what are Booleans are in computer science.

OBJECTIVES

醒目 Learners will be able to explain in their own words what a Booleans is in computer science
醒目 Learners will be able to understand that Booleans can hold values that will equal True or False
醒目 Learners will be able to identify how Booleans operators can help in coding environments

STANDARDS

⌘ K12 CS:
Algorithms and Programming.Control; Algorithms and Programming.Program Development

⌘ CSTA:
K-2: 1A-AP-10, 1A-AP-11, 1A-AP-12, 1A-AP-15
3-5: 1B-AP-10, 1B-AP-11, 1B-AP-12, 1B-AP-16, 1B-AP-17
6-8: 2-AP-12, 2-AP-13, 2-AP-16
9-10: 3A-AP-18
Introduction

Linking
On the board, write a series of statements that you can answer with “true” or “false.” When the students arrive, have them decide if those statements are true or false! Statements could be:

- There are no windows in the room
- There are five doors in the room
- The classroom has a desk

Let students decide and then go over each type together.

Engage
Yesterday we talked about conditionals, but now we want to go another step forward. When you use a conditional you are comparing something typically. Usually it is to see if something is true/false or if the value is within a certain range.

Take the door for example:
IF door is open, walkthrough the door.
ELSE open the door.

In this scenario, we are trying to see if the statement “door is open” is true or false. IF it is true, we walkthrough the door. ELSE if it is false we open the door.

Booleans are pretty simple, but powerful. Basically a Boolean data type has only two values “true” or “false.” This means Booleans only return a value of true or false after it evaluates a condition. For example, you all looked at the statements I gave and determined if they were true or false, meaning you returned to me a Boolean value.

Let’s deep dive into Booleans a little more!

Collect Kanos
Retrieve Computers, Turn On, Log In
**Exploration Activities**

**Challenge 1: What ARE Booleans?**

As already stated, Booleans are a very simple concept, but they are very powerful. If a program can evaluate a line of code and determine if it is true or false it can be used in different environments or in a new scenario. Booleans allow code to adapt and become intelligent!

But how does it do that? Well, let's do a very basic demonstration. Everyone sit in a circle on the floor. I will say a series of statements. Stand up if the statement is true for you. Sit down if the statement is false for you. We will all start in the “false” position.

- Are you wearing jeans?
- Are you wearing jeans and a T-shirt?
- Are you wearing jeans that are not blue jeans?
- Are you wearing sneakers?
- Are you wearing sneakers that are not mostly white or sandals?
- Are you wearing earrings?
- Are you wearing earrings or hair ornaments?

Okay, pretty simple but now I have a question for you. WHY did you stand up or sit down? What were the keywords in the statements I gave you that allowed you to know if you should continue standing?

Let students turn and talk to each other at this point. You may need to say the phrases one more time or write them on the board so they can look at it.

The keywords are AND, NOT, OR. These words helped us to decide if the statement would be true or false. These keywords are known as Boolean operators! We can use these to help us evaluate situations and determine if they are true or false.

**Challenge 2: Practice with Booleans**

Now let's try another activity together. You and your partner will take turns to play a game of 20 questions. I have a pre-determined list of 30 items I will write on the board. One partner will select in their head what item they want to be. It is up to their partner to determine what the item is.
For example, I picked a slice of bread as my item. My partner may ask the following questions and I would respond this way:

- Are you alive? False
- Are you very big? False
- Are you used for fun OR learning? False
- Are you eaten? True
- Are you eaten in solid AND liquid form? False
- Are you eating in solid form? True

My partner would continue until they get to the item!

**Challenge 3: Play with Booleans on the Kano!**

In the conditional session we created our own games in Make Pong. If you have finished your game switch spots and sit with another person's Kano. Play the game and then look at their code. Can you determine what their conditions mean? What was being evaluated as true or false? Record your answers in your engineering log!

---

**Closing**

**Evaluation**

Turn and talk to your partner and think of WHY code would need to use Booleans? If it helps, think back to our robot activity? Would it have been helpful to be able to use true and false statements?

**Closing**

Tonight try looking up more information on Booleans! See if you can figure out who discovered Booleans and if you can, see if you can find out something about logic gates!

---

**Kano Clean up**

Power down and put away Kanos
In Make Pong, you saw a lot of Booleans being used. Below, can you determine what the Booleans were evaluating as true or false? The first one has been done as an example.

**Code Blocks**

**What Does the Boolean Evaluate**

- **If hit = = true; if the ball is hit**
  - if hit = = true; if the ball is hit
- **Score Player 1 wins**
  - if Score = = Player 1 wins
- **Key pressed W Player 1 wins**
  - if Key pressed W Player 1 wins
- **Ball position < Middle point > Rebound**
  - if Ball position < Middle point > Rebound
Conditionals: Booleans

Creator Question
Booleans help us create fun and complex games. Now design your own! Use the space to add more advanced modifications to your Pong game. Also, label how you would code the new parts of your game with a conditional statement and Boolean!

Impact Question
You gave your thoughts on the development of artificial intelligence but what about the positive and negative consequences of having intelligent machines? What are some of the good things about it? What are some of the bad?
LESSON 15

Debug It!

Beginner 1 hr

Learners will use Scratch to understand debugging!
(It is recommended for learners who know PEMDAS or BODMAS)

OBJECTIVES

- Learners to investigate the problem and find a solution to a debugging challenges
- Learners to explore a range of concepts (including sequence) through the practices of testing and debugging
- Learners to develop a list of strategies for debugging projects

STANDARDS

K12 CS:
- Computing Systems.Troubleshooting;
- Algorithms and Programming.Algorithms
- Algorithms and Programming.Control;
- Algorithms and Programming.Program Development

CSTA:
- K-2: 1A-AP-10, 1A-AP-11, 1A-AP-12, 1A-AP-14, 1A-AP-15
- 3-5: 1B-AP-10, 1B-AP-11, 1B-AP-12, 1B-AP-16, 1B-AP-17
- 9-10: 3A-AP-18
Introduction

Linking
Write on the board two math problems and the given solution:
5 - 1 x 3 + 1 = 13
2 x (6 + 1) - 1 = 12

Are these solutions correct? Why or why not? When you have solved the problems turn and talk to a partner and see what they think.

Engage
In math, you follow very specific steps to get an answer. If we did something wrong in the process, then the answer will not be what we wanted. Code is very similar. You may want the program to do a very specific function; however, if you did not code the sequence correctly the program will not run the way you want to. When you are looking back at your code to figure out what went wrong, programmers call this debugging.

Today, we will put ourselves in the shoes of a programmer. We will look at code for programs and try to figure out what went wrong and how we can fix it.

Before we start though, do you know why it is called debugging? (Give the students a few minutes to chat). Not too long ago computers were huge! They took up entire rooms. Sometimes bugs would get stuck in them. The story goes that in the 1940s, Grace Hopper, the mother of computer programming who was one of the first and developed the first compiler, noticed a moth stuck in the computer which was making it not work. She said they were “debugging” the computer.

Collect Kanos
Retrieve Computers, Turn On, Log In

Exploration Activities

Challenge 0: Review Scratch
Before we start debugging challenges in Scratch, we need to make sure that everyone understands the basic blocks in Scratch. Start with a simple activity of “When green flag is clicked move 10 steps.” You can vary this with the movements, sound, or the signal for starting the command.
Now that we have a refresher on Scratch, let’s code something as a class! I want to code my sprite to turn in a circle when the space bar is clicked. The image below is what your code should look like for the learners:

The blocks should be the following:
- When “space” key pressed
- Turn 90 degrees
- Turn 90 degrees
- Turn 90 degrees
- Turn 90 degrees

Now try clicking the space bar. What happens? (nothing)

Ask the learners why they think this happens. (Four 90 degrees equals 360 degrees so it will not appear like it is turning)

What can we do to make the code work now? (Have less than four 90 degree move blocks)

Let’s do one more together. When we click the green flag we want the sprite to have a conversation bubble that comes up and says “meow, meow, meow” and then makes the same sound three times.

The blocks should be the following:
- When green flag clicked
- Say “Meow, meow, meow! For 2 secs
- Repeat 3
- Play sound meow

Now try clicking the green flag. What happens? (The conversation bubble comes up but the sprite says meow once)

Ask the learners why they think this happens. (The blocks are not in the correct order)

What can we do to make the code work now? (Move the blocks so you have the sound and text in the loop, and the sound block should be first)
Challenge 2: Create a Flawed Program!

If you have extra time or your programmers are quickly figuring out the problems for flawed code move them to Challenge 2!

Now that the students understand what it means to debug have each pair create a program. It can be as simple or as complex as they want, but they have to have a clear idea of what the code should do. Give each pair 15 minutes to code their program.

After the 15 minutes, the pairs will switch with another group. Their mission is to figure out what is wrong with their code and fix it!

Closing

Sharing

Have students share the flawed program that they were at. What was the program supposed to do? What was wrong with it? Were you able to fix it?

Closing

Tonight, in your logs reflect and answer the following questions:

- What was one thing you learned when you debugged the Guess A Number Game in Scratch?
- What was your favorite part about debugging?
- What are other ways you find debugging in day-to-day life? (sports example: - didn't score a goal in soccer, what went wrong?)

Kano Clean up

Power down and put away Kanos
**Explorer Question**

While you were in Scratch, you saw a few examples of code that was supposed to do something, but it didn’t work. In the images below, can you identify why the code does not work?

<table>
<thead>
<tr>
<th>Code Blocks</th>
<th>What the code is supposed to do</th>
<th>Why it doesn’t work</th>
</tr>
</thead>
</table>
| ![Code Blocks](image1.png) | The blocks should be the following:  
- When “space” key pressed  
- Turn 90 degrees  
- Turn 90 degrees  
- Turn 90 degrees  
- Turn 90 degrees | |
| ![Code Blocks](image2.png) | The blocks should be the following:  
- When green flag clicked  
- Say “Meow, meow, meow! For 2 secs  
- Repeat 3  
- Play sound meow | |
| ![Code Blocks](image3.png) | The blocks should be the following:  
- When the sprite is clicked  
- Move across the screen  
- If it hits the wall it will reverse directions and repeat | |
Programmer Question
In your own words, what does it mean to “debug” something? Can you provide an example of something you may need to “debug”?

Creator Question
Can you design a simple program in Scratch? For this project, program a character to dance around the screen in any way you want! Below, write out the code blocks you would need to make the character dance! You can draw out your idea as well!

Impact Question
In life we encounter problems and we have to think of ways to fix the problem. Debugging programs is the same process. Sometimes your program doesn’t work the way you want and you have to figure out what went wrong! Think about things you have had to “debug” in your life. What was supposed to happen but didn’t? How did you go about fixing it?
LESSON 16

Impact of Technology

Beginner 1 hour 30 mins

Students will reflect on the impact questions they answered throughout the course and discuss the implications of a more technologically driven world.

OBJECTIVES

- Learners will be able to discuss some of the major trends in technology
- Learners will be able to articulate ways to protect their information in an increasingly digital world

STANDARDS

K12 CS:
- Impacts of Computing.Culture
- Impacts of Computing.Social Interactions

CSTA:
- K-2: 1A-IC-16, 1A-IC-17, 1A-IC-18
- 3-5: 1B-IC-18, 1B-IC-19, 1B-NI-05
- 6-8: 2-IC-20, 2-IC-23, 2-NI-05
- 9-10: 3A-IC-24
Introduction

Linking
The time is 2040 and the future is now. You are the scientist designing all the new inventions and developments of the world! What are your top three new inventions you have created for this new future? Draw and/or write out the descriptions of each invention. If you are done share your inventions with a friend!

For about 2 minutes, have learners share some of the things they “invented.”

Engage
The future seems super fantastic! We all thought of cool ideas that COULD become reality with enough brains and work going into it.

Almost all technology we interact with now wasn’t here 20 years ago, and even wasn’t here just 1 year ago! Technology is constantly updating and improving. With these updates our lives change, sometimes dramatically for better or worse.

Today, we are going to look a little more critically at what these changes mean for not only ourselves but the world!

Collect Kanos
Retrieve Computers, Turn On, Log In

Exploration Activities

Challenge 1: Where Does Technology Have an Impact?
While we have been learning more about coding we have also been answering a few discussion questions each session called the “Impact Question.” Can you take out your worksheets now and read over the questions. While you read over the questions, think about what the questions were asking you. Give learners 5 - 6 minutes to read over them.

Now that we’ve had a bit of a refresher, what were some of these questions asking us? (Potential answers: how do we interact with technology, how do we form relationships, how has technology changed us?)
If we had to separate these questions into categories how would we separate them? (Possible themes: relationships, culture, society/government, innovation, privacy/security)

We can see that the questions look into a lot of different ways that technology has an impact on our lives. What I will do now is separate you into groups so that you are inside one of the categories. Separate the learners into the different categories.

Now that we are in our category groups, I want you to discuss together and answer the following questions:

- How does technology affect my category in a good way?
- How does technology affect my category in a bad way?
- How does technology affect my category in how I deal with my friends and family?

For example. If your category is security/privacy some answers may be:

- A positive thing: Technology has made it that I can have more secure accounts that have passwords.
- A negative thing: People are now trying to hack and take our personal information from online.
- Relationships: With regards to privacy, I may have an online account I do not share with my family so I can be honest online or vent to an online community. This can be good and bad.

You are required to:

- Answer each question in at least three sentences
- Draw a picture for each answer

Challenge 2: How Can We Grow with Technology?

Stay in your current groups and now answer the following questions.
We talked about these different categories, but how can we grow with technology in these areas? How do you think technology will change our category in the future? In what ways?

Example (security/privacy)
We can grow with technology in regards to security by making sure that as technology develops we are aware of how to code and how it affects our lives. Such as making sure we are active participants in understanding how companies or websites collect our information and use it. As technology changes we will want to have more secure networks to make sure our information isn't abused.

Again, answer each question in at least three sentences and draw a picture to represent your answer.
Challenge 3: Give Advice for the Future!

You and your group are the experts of your category now! Can you share that information with the rest of the group? In the next 20 minutes, create a poster that presents the following information:

- What is your category?
- How has your category been affected by technology in a positive way?
- How has your category been affected by technology in a negative way?
- What are ways we can make sure there are not as many negative impacts for our category?
- What advice would you give the group about how to handle technology in your category?

When you present, make sure everyone takes a turn sharing information. You can present your information on a poster or, if you have access to a computer, in a PowerPoint.

Closing

Sharing

Let each group have about 5 minutes to share their category. Let learners give advice to each other and have time to ask questions about each category.

Closing

Tonight, talk to an adult in your life about the category you researched and see if they would give the same advice you would or if it would be different. Record their answers in your engineering log and be prepared to share it with the group!

Kano Clean up

Power down and put away Kanos
Design Thinking

Intermediate 2hrs (Can split into multiple sessions)

In this lesson you will use the popular Design Thinking framework to design your own product and give important feedback to Kano!

OBJECTIVES

- Learners can identify the steps in design thinking.
- Learners understand what it means to give constructive feedback and how that applies to design thinking.

STANDARDS

- NGSS:
  - 3-5-ETS1-1 D  3-5-ETS1-2  MS-ETS1-1 D  MS-ETS1-2
- CSTA:
  - K-2: 1A-AP-15
  - 3-5: 1B-AP-13, 1B-IC-20
  - 6-8: 2-AP-16, 2-AP-17
  - 9-12: 3A-AP-13
Introduction

What is your favorite toy or game? Gather a list from the students and write it on a board. Everything you named was thought out before it was made. Like when we decide which way to walk to a store, or when we think about writing an essay we think about it before making a decision.

Today we are going to talk about Design Thinking. Walkthrough each of the steps and ask students for examples of things they’ve tested or been part of testing.

Empathize

Work to fully understand the experience of the user for whom you are designing. Do this through observation, interaction and immersing yourself in their experiences.

Define

Process and synthesise the findings from your empathy work in order to form a user point of view that you will address with your design.

Ideate

Explore a wide variety of possible solutions through generating a large quantity of diverse possible solutions, allowing you to step beyond the obvious and explore a range of ideas.

Prototype

Transform your ideas into a physical form so that you can experience and interact with them and, in the process, learn and develop more empathy.

Test

Try out high-resolution products and use observations and feedback to refine prototypes, learn more about the user and refine your original point of view.

Explain what Kickstarter is and why it’s a good way to validate an idea - the public gives you a response on whether they are interested in your idea and puts down money to support. It also helps early companies to get a product out and test with people instead of creating in a room and launching without user feedback. You can even show some Kickstarter projects on the projector.

Explain that Kano was a Kickstarter that raised over 1.5m USD on the platform. Kano continues
to grow with feedback from students and teachers around the world. Today, you will give Kano feedback. You can use this presentation!

**Collect Kanos**
Retrieve Computers, Turn On, Log In

**Other resources:**
Paper, pencil, crayons (optional)

**Exploration Activities**

**Challenge 1:**
Break into groups of four and each group chooses one Kano app:
- Make Art
- Make Minecraft
- Terminal Quest
- Make Snake

Answer the questions (in writing to share with Kano!):
- What is your favorite part of the app?
- What is challenging about the app?
- What would you improve about the app?

Class to discuss some of the feedback.

**Challenge 2:**
You have just used Design Thinking to give Kano feedback on one of the apps already on it. Now what if you can give a recommendation for a new app? Taking the feedback you heard from the class, use Design Thinking to create your own app.
• What problem are you looking to solve with the app?
• What is the app and its purpose? Who would want to use it? In your groups use pen and paper to write out your thoughts.

**Challenge 3:**

Break students into groups and have them think of a few brands and their logos. What do you like about those brands? What about their logos? What don’t you like about the brands or logos? What does a logo express about a brand?

Discuss as a class.

Then, open up Kano kits and launch Make Art.
Logo frenzy!

Choose two Logos from world.kano.me and recreate them in Make Art - add your own creative flair to these logos.

**Challenge 4:**

Create your logo for the app you designed in Challenge 2!
First, sketch out the logo on paper and pencil (if available).

Then Code in Make Art!
Save your work and share to Kano World!

**Closing**

**Sharing**

Groups share their apps and logos. The class can vote on their favorite!

Remind students to fill out their daily logs in their engineering logs!

**Kano Clean up**

Power down and put away Kanos
Building a Better World

Beginner 1 hr 30 mins (Recommended over 2 days)

Learners critically evaluate their world and surroundings and use Make Art and Make Minecraft to picture a more ideal world.

OBJECTIVES

- Learners to get introduced to product development and a popular framework, design thinking.
- Learners to get exposure to Kickstarter and the idea of raising money to start a business.
- Learners to understand the importance of getting and giving feedback when designing a product.

STANDARDS

NGSS:
- 3-5-ETS1-1 D
- 3-5-ETS1-2
- MS-ETS1-1 D
- MS-ETS1-2

CSTA:
- K-2: 1A-AP-15
- 3-5: 1B-AP-13, 1B-IC-20
- 6-8: 2-AP-16, 2-AP-17, 2-AP-17
- 9-12: 3A-AP-13

MATERIALS NEEDED
Design Thinking Lesson..
Introduction

Linking
If you could change one thing about your community, what would it be? Do you want access to more parks and bike lanes? Do you want a heavy metal concert to happen every Friday? Take a few minutes to write or draw your thoughts out.

Once students are done, have a few people share out their thoughts on what they would like to add to their community space.

Engage
For the next few days, we want to think of ourselves as engineers, policy makers, and activists. There are a lot of problems in our community, but that means that there are a lot of creative solutions to these problems!

Collect Kanos
Retrieve Computers, Turn On, Log In

Exploration Activities

Challenge 1: What is Design Thinking?
If your students have forgotten, review what Design Thinking is. Make a note that we will be using this thought process to think of ways we can improve our communities.

Challenge 2: Community Problems + Solutions
So what does Design Thinking have to do with anything? Well, today we are all going to receive a challenge. Before we do this, break into groups of four. You are a team tasked with identifying a problem in your community and to propose an innovative solution to solve that problem.
In groups of four, you will need to:

- Identify the problem in your community
- Use Design Thinking to define your problem
- Use Kano to prototype and test a solution
- Pitch your proposal to the class

Be creative! You can choose any app (Minecraft, Make Art, Make Music) to express your ideas.

Students work on defining their problem and coming up with ideas for their test and prototype. They come up with a plan to use Kano to express their ideas.

Save your work and share to Kano World!

---

**Closing**

20 mins

**Closing**

**Sharing**

Students will share their ideas (as they are now) with the class. Ask students to state what the problem identified was, why it was important to them, and then their creative idea.

**Evaluation**

Before students leave have them write in their engineering logs what was difficult about this challenge. Also, have them answer how they overcame this difficulty.

As a homework extension, have students finish their proposed solutions to the challenge. They can also spend the next day in class finalizing their work.

---

**Kano Clean up**

5 mins

**Kano Clean up**

Power down and put away Kanos
Explorer Questions
Directions: Fill out each section by reflecting on your Design Challenge Project!

**Empathize**
Watch and Listen: What population will you work with?

**Define**
What is your idea/vision based upon the population and the problem you will work with?

**Ideate**
What ideas did you brainstorm?

**Prototype**
Provide a brief explanation of your prototype.

**Test**
How will you test your prototype?
Programmer Question
Directions: When you created your prototype, you had to code a portion of the project on Kano. What did you code? Why did you use that coding environment? What were the limitations of that coding environment for your project?

Creator Question
Directions: Part of coding is thinking about ways to improve your projects. Below, draw a design on how you could improve your Design Thinking Project for next time! Label the drawing explaining how you would improve it.
Celebrate Creation!

気軽に 1 hr 30 min

Learners will show off the work they have completed and flex their new coding skills!

OBJECTIVES

⚡️ Learners will articulate their projects and what they did to complete them

⚡️ Learners will critique their classmates and give constructive feedback
Introduction

Linking
We have done a lot with our Kanos! Learned to build, create, and make new things! Can you draw a picture representing your time with Kano?

Engage
Today we get to celebrate and show off all the super cool things we have created! Today is about giving constructive feedback to our friends and thinking of more ways we can be innovative!

Exploration Activities

Challenge 1: How to Critique?
Today, we want to share with everyone the work we did over the past few weeks and also give a critique on our work so we can improve for next time. A critique is when we give an analysis or feedback on something.

It is great to have an opinion on something but we want to make sure we can articulate something. We are never allowed to say we do or do not like something “just because.”

If I am giving feedback to another person, I want to be able to explain my thoughts, feelings, and reasoning.

For example, if I am asked if I like a book and I say “yes” I may explain myself by saying, “The book was very interesting. It had characters that I could identify with. Like the main character in the story, I am the younger sister and know what it is like to have an older sibling.”

In this example, I explain why I liked a book. I could identify with the main character and it helped me appreciate the story more.
We all have feelings and opinions with reasons behind them. We want to make sure we are expressing this!

Also, for this exercise we want to frame our critiques in a positive framework. If we see something we don’t like we should not say “that is stupid” or “that is dumb or ugly.” Instead we want to frame everything in a way that the person can grow. If you are wondering why there is a rock on top of the tree ask the question, “Hey, maybe next time we should have rocks on the ground.”

Let’s practice this! Everybody draw a quick picture in 5 minutes. I'll count down and yell “Go” loudly and you can start. After the 5 minutes, you will stop and turn and talk to a partner. Critique their work!

**Challenge 2: Kano World Gallery Walk**

Now that we have learned to critique, we will go around in groups to share our work! Each group will spend 10 minutes at the Kanos to explain their work and then we will switch!

**Challenge 3: Group Celebration**

Hopefully we got to see everyone’s work! Now turn and talk to your neighbor and share what was your favorite Kano creation! What idea was super cool.

---

**Closing**

15 mins

**Sharing**

This is our last day with the Kanos, so let’s give some feedback on what our experience was. In your engineering logs answer the following questions:

- What did you like about the Kanos?
- What was your favorite class?
- What class do you wish was different? Why?

On a post it note before you leave today, rate your time with Kano. If you loved it give it a 5! If you hated it give it a 1. Post it on the door as you walk out!

**Kano Clean up**

5 mins

Power down and put away Kanos
Appendix
Kano Implementation Tips

Bringing Kano to your learning environment is a new and exciting step! Below are a few tips to get you ready for your first day!

1. **Label Your Kanos:** Number your Kano Computer and Screen boxes for easy assignment
2. **Set Up a Routine:** Before the first day using the Kanos decide where you want the Kanos to live and how learners will take them out and put them away each session
3. **Assign Partners or Groups:** Decide who you would like to work together.
4. **Know How to Put Kanos Away:** Did you know that all your Kano pieces can fit inside the yellow box? Once learners have all built their Kano, show them how the keyboard and all the cables will fit easily in the back of the screen and yellow box. This will help cables stay in good condition!
5. **Set Up a Kano World Account:** If you would like to create a class account where you can have every learner share their work online, you can create a class profile at world.kano.me
6. **Creating Kano Roles:** Creating jobs and responsibilities for the Kanos gives ownership to the kits.

Sample Roles Include:

- Kano Keeper: Checks to make sure Kanos are put away correctly
- Time Keeper: Manages time for each activity and tells the group when to put Kanos away
## Fundamentals of Computer Science

### Curriculum Standard Alignment

#### Computing Systems
- Devices

#### Algorithms + Programming
- Algorithms
- Control
- Program Dev
- Variables

#### Impacts of Computing
- Culture
- Social Interactions
- Cybersecurity

#### Network
- Cybersecurity

<table>
<thead>
<tr>
<th>US K12CS</th>
<th>3-5</th>
<th>K-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-2: 1A-AP-08</td>
<td>Impact of Computing</td>
<td>Impact of Computing</td>
</tr>
<tr>
<td>K-2: 1A-AP-10</td>
<td>Algorithms + Programs</td>
<td>Impact of Computing</td>
</tr>
<tr>
<td>K-2: 1A-AP-11</td>
<td>Algorithms + Programs</td>
<td>Impact of Computing</td>
</tr>
<tr>
<td>K-2: 1A-AP-12</td>
<td>Algorithms + Programs</td>
<td>Impact of Computing</td>
</tr>
<tr>
<td>K-2: 1A-AP-14</td>
<td>Algorithms + Programs</td>
<td>Impact of Computing</td>
</tr>
<tr>
<td>K-2: 1A-AP-15</td>
<td>Algorithms + Programs</td>
<td>Impact of Computing</td>
</tr>
<tr>
<td>K-2: 1A-CS-01</td>
<td>Algorithms + Programs</td>
<td>Impact of Computing</td>
</tr>
<tr>
<td>K-2: 1A-CS-02</td>
<td>Computing Systems</td>
<td>Impact of Computing</td>
</tr>
<tr>
<td>K-2: 1A-DA-07</td>
<td>Impact of Computing</td>
<td>Impact of Computing</td>
</tr>
<tr>
<td>K-2: 1A-IC-16</td>
<td>Impact of Computing</td>
<td>Impact of Computing</td>
</tr>
<tr>
<td>K-2: 1A-IC-17</td>
<td>Impact of Computing</td>
<td>Impact of Computing</td>
</tr>
<tr>
<td>K-2: 1A-IC-18</td>
<td>Impact of Computing</td>
<td>Impact of Computing</td>
</tr>
<tr>
<td>3-5: 1B-AP-10</td>
<td>Impact of Computing</td>
<td>Impact of Computing</td>
</tr>
<tr>
<td>3-5: 1B-AP-11</td>
<td>Impact of Computing</td>
<td>Impact of Computing</td>
</tr>
<tr>
<td>3-5: 1B-AP-12</td>
<td>Algorithms + Programs</td>
<td>Impact of Computing</td>
</tr>
<tr>
<td>3-5: 1B-AP-13</td>
<td>Algorithms + Programs</td>
<td>Impact of Computing</td>
</tr>
<tr>
<td>3-5: 1B-AP-14</td>
<td>Algorithms + Programs</td>
<td>Impact of Computing</td>
</tr>
<tr>
<td>3-5: 1B-AP-15</td>
<td>Algorithms + Programs</td>
<td>Impact of Computing</td>
</tr>
<tr>
<td>3-5: 1B-AP-16</td>
<td>Algorithms + Programs</td>
<td>Impact of Computing</td>
</tr>
<tr>
<td>3-5: 1B-AP-17</td>
<td>Algorithms + Programs</td>
<td>Impact of Computing</td>
</tr>
<tr>
<td>3-5: 1B-CS-01</td>
<td>Algorithms + Programs</td>
<td>Impact of Computing</td>
</tr>
<tr>
<td>3-5: 1B-CS-02</td>
<td>Algorithms + Programs</td>
<td>Impact of Computing</td>
</tr>
<tr>
<td>3-5: 1B-IC-18</td>
<td>Impact of Computing</td>
<td>Impact of Computing</td>
</tr>
<tr>
<td>3-5: 1B-IC-19</td>
<td>Impact of Computing</td>
<td>Impact of Computing</td>
</tr>
<tr>
<td>3-5: 1B-IC-20</td>
<td>Computing Systems</td>
<td>Impact of Computing</td>
</tr>
<tr>
<td>3-5: 1B-NI-05</td>
<td>Impact of Computing</td>
<td>Impact of Computing</td>
</tr>
</tbody>
</table>
## Fundamentals of Computer Science Curriculum Standard Alignment

<table>
<thead>
<tr>
<th>CSTA</th>
<th>6-8</th>
<th>9-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>6-8: 2-AP-12</td>
<td>Computing Systems</td>
<td></td>
</tr>
<tr>
<td>6-8: 2-AP-13</td>
<td>Impact of Computing</td>
<td></td>
</tr>
<tr>
<td>6-8: 2-AP-16</td>
<td>Impact of Computing</td>
<td></td>
</tr>
<tr>
<td>6-8: 2-AP-17</td>
<td>Algorithms + Programs</td>
<td></td>
</tr>
<tr>
<td>6-8: 2-CS-03</td>
<td>Algorithms + Programs</td>
<td></td>
</tr>
<tr>
<td>6-8: 2-IC-20</td>
<td>Algorithms + Programs</td>
<td></td>
</tr>
<tr>
<td>6-8: 2-IC-23</td>
<td>Algorithms + Programs</td>
<td></td>
</tr>
<tr>
<td>6-8: 2-NI-05</td>
<td>Algorithms + Programs</td>
<td></td>
</tr>
<tr>
<td>9-10: 3A-AP-13</td>
<td>Algorithms + Programs</td>
<td></td>
</tr>
<tr>
<td>9-10: 3A-AP-18</td>
<td>Algorithms + Programs</td>
<td></td>
</tr>
<tr>
<td>9-10: 3A-IC-24</td>
<td>Algorithms + Programs</td>
<td></td>
</tr>
</tbody>
</table>

### Common Core

<table>
<thead>
<tr>
<th>CCSS.MATH.CONTENT</th>
<th>CCSS.MATH.PRACTICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.OA.A.1</td>
<td>Addition + Subtraction</td>
</tr>
<tr>
<td>1.OA.A.1</td>
<td>Addition + Subtraction</td>
</tr>
<tr>
<td>HSG.M.G.A.3</td>
<td>Geometry</td>
</tr>
<tr>
<td>6.G.A.3</td>
<td>Geometry</td>
</tr>
<tr>
<td>4.OA.C.5</td>
<td>Patterns</td>
</tr>
<tr>
<td>3.OA.D.9</td>
<td>Patterns</td>
</tr>
<tr>
<td>MP1</td>
<td>Problem Solving</td>
</tr>
<tr>
<td>MP2</td>
<td>Reasoning</td>
</tr>
<tr>
<td>MP3</td>
<td>Reasoning</td>
</tr>
<tr>
<td>MP4</td>
<td>Mathematics</td>
</tr>
<tr>
<td>MP5</td>
<td>Tools</td>
</tr>
<tr>
<td>MP6</td>
<td>Precision</td>
</tr>
<tr>
<td>MP7</td>
<td>Structure</td>
</tr>
<tr>
<td>MP8</td>
<td>Reasoning</td>
</tr>
</tbody>
</table>
Student Engineering Log

Keeping Track of Your Work on Kano

We recommend that you use an Engineering Design Notebook (EDN) in your class to help students track their progress and be reflective of their time on Kano. You can use a notebook, binder, or folder for the EDN, as long as it is completely devoted to the work completed on the Kano. The EDN will hold any notes, drawings, handouts, or assignments that are related to Kano.

Each student will be responsible for their own EDN. It will be their responsibility to organize, update, and monitor. As the educator, you can use the EDN as an assessment piece and collect it to track student progress. We recommend that you require each student to keep their notebook's information in chronological order.

Included is a daily log your students can use to track their progress as they work on Kano.

Your personal Engineering Design Notebook will include:

- Class handouts
- Daily logs and notes
- All sketches, plans, and drawings
- Notes from design reviews
- Calculations relevant to your project
- Documentation of the evolutionary changes of your project
- All completed and returned assignments
- Final (turned-in) version of any individual assignments that are due
Student Engineering Log

What was your challenge?

How did you solve this challenge?

What worked? What didn’t work?

How can we make it better? How can we change it for next time?

In Class Reflection
Surveys + Kano

We always want to make sure that we are collecting information from our students and educators on how Kano is doing in your learning environment.

We would love to hear from you on any feedback you may have for Kano. Below are two surveys you and your students can fill out:

Teacher Survey

Student Survey

Also, on the following page, you will find a sample survey you can use in your classroom to access your students’ experiences with the Kano computer kits!
Kano Student Survey

We really want to know what you thought of Kano. By completing this survey you're helping us to make Kano even better for other students!

Name: __________________________

Class: __________________________

School: _________________________
For the below questions please indicate how strongly you agree or disagree with the following statements. Please circle only one answer.

<table>
<thead>
<tr>
<th>Statement</th>
<th>Agree</th>
<th>Unsure</th>
<th>Disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kano is fun</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kano is easy to use</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kano is helping me learn more</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kano lets me explore topics I’m interested in.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I feel like I can be creative on the Kano</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>What I am learning on the Kano will help me later in life</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I like to work with a partner on the Kano</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I feel like I can learn at my own pace on the Kano</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I feel like I understand programming languages more</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I feel like I understand hardware and software more</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Circle your favorite apps on the Kano:

- Kano Code
- Hack Minecraft
- Make Light
- Make Snake
- Make Pong
- Make Art
- Terminal Quest
- Story Mode
- Other

Circle your least favorite apps on the Kano:

- Kano Code
- Hack Minecraft
- Make Light
- Make Snake
- Make Pong
- Make Art
- Terminal Quest
- Story Mode
- Other

Circle all the adjectives that describe a STEM job:

- Fun
- Make the world a better place
- Exciting
- Repetition
- High paying
- Creative
- Flexible
- Hard
- Challenging
- Easy
- Boring
- Team-work
- Working alone
Share anything else about your Kano experience:
Teacher Reflection Guide

1. What went well in this lesson? Why?

2. What problems did I experience? Why?

3. What could I have done differently?

4. What did I learn from this experience that will help me in the future?

5. Preparation and research - What could I have done differently?

6. Assessment - Does my method(s) of assessment measure what I want? How did the class do? What should I change for next time?
Reviewing the Impact Questions

At the end of each worksheet, you should have seen a section called the “Impact Question.” These are more in-depth questions around the role of technology in: culture, society, politics, and our own personal relationships.

These questions are to be used as a reflection piece. Responses should be more than three sentences, with deeper discussion turning potentially into an essay. Learners will use their responses later in the “Impact of Technology” lesson to have a debate with their classmates and think collectively on the role technology has had on our past, present, and will have on our future.

You do not need to evaluate these responses, but they serve as great talking points to move forward! Take some time before or after a lesson to discuss thoughts around the impact questions!
ANSWER KEY:

Explorer Questions:
1. Memory Card: This gives the Kano powers and can hold thousands of songs, games and ideas. The operating system lives in here
2. Power Pieces: Gives power to the Kano
3. HDMI Cable: Lets the Kano display images
4. Keyboard + Mouse: Is the input device you can type into
5. Sound Sensor: Lets you listen to the world around you
6. Raspberry Pi: The brain of the Kano Computer
7. DIY Speaker: Gives the Kano a voice. Is made of four parts (power, speaker, circuit board, and sound)
8. USB Hub: Gives your Kano connection powers
9. Power Board: Gives the Kano a power button

1. Graphics Chip
2. HDMI
3. USB

Programmer Questions
1.

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raspberry Pi, Memory Card, Power Pieces, Speaker, HDMI Cable, Keyboard, Case, Screen</td>
<td>Make Art, Kano Code, Terminal Quest, Scratch, Make Snake, Make Pong, Make Light, Hack Minecraft</td>
</tr>
</tbody>
</table>

2. Inputs: keyboard                        Outputs: Screen

Creator Question

Let the learners be creative on what kind of computer they want to create. Some key components they should have though:
- The operating system (memory card, hard drive, etc.)
- A display (screen, TV, etc.)
- Power of some sort
- A way to input information (keyboard, touch screen, etc.)
Explorer Questions

What are some examples of coding languages?
Responses will vary. Possible answers include: python, Java, JavaScript, CoffeeScript, C++, HTML, etc.

When you did the drawing activity what happened?
Responses will vary. Learners should be able to elaborate on what they did in the drawing activity, why it was easy or hard, and why they think they did the activity.

When you did the robot activity what happened?
Responses will vary. Learners should be able to elaborate on what they did in the robot activity, why it was easy or hard, and why they think they did the activity.

Programmer Questions

Why is it important to be detailed when you give instructions?
Possible response: It is important to be detailed when you give instructions because sometimes people and computers get confused. Even though you (the storyteller or programmer) may know what comes next, the person/computer listening or reading doesn’t. If they are confused or do not know what to do then you may not get the result you wanted.

If you wanted to draw the image below, what instructions would you give someone?
Answers will vary. However, specific instructions should be given for each portion of the drawing. Possible answers include:
- Draw an oval
- Starting from the center of the oval at the top draw a curved line going down towards the right. The line should not be drawn outside of the oval
- 2 cms down from the first curved line draw another
- Repeat the lines four more times
- Starting from the center of the oval at the top draw a curved line going down towards the left. The line should not be drawn outside of the oval
- 2 cms down from the first curved line draw another
- Repeat the lines four more times
- At the top of the oval draw three bundles of leaves that lean to the left
- At the top of the oval draw three bundles of leaves that lean to the right

Creator Question

For this, learners do not need to explain how to go around the obstacle course, just that they have created one. However, they should explain how you start and end the course.
ANSWER KEY:

Explorer Questions:

Programmer Questions

Text and block based code are very similar in that you are creating code to create something. With text based code you have to type line for line of the code so the program is executed exactly as you want. With block based you can create code that may not need to be in an exact order or completely correct in syntax.

Creator Question

Let learners be creative! At the minimum you should see the following:

- A drawing of the app logo
- At least two sentences about what the app does
- At least one sentence about what you create with the app
- A statement on if it uses block or text based code
Story Mode Answer Key

**Explorer Questions**

Find the names on the Map

1. Power Path
2. Power Port
3. HDMI Lake
4. Fibreglass Plains
5. Vector Village
6. Museum of Make Art
7. Python Jungle
8. Block Mines
9. Ether St Station
10. Port Ether
11. HD Hill

Find the names on the CodeX

1. Binary
2. Power
3. Energy
4. LED
5. Voltage
6. Current
7. Pixels
8. HDMI
9. Resolution
10. Speaker
11. Sound Wave
12. Mono
13. Stero
14. Frequency
15. Vector
16. Terminal
17. If Statement
18. Coordinates
19. Logic Gate
20. Python
21. CoffeeScript

**Programmer Question**

Possible Answer: Hardware is the physical side of computers and the software is the programmed side (can’t touch) side of computers. Software is programmed to do something, which tells the hardware to do something (save a file, create a file, turn something on/off)

**Creator Question**

Let learners be creative! Make sure there is at least a drawing of a CodeX and an explanation on how they would get the CodeX (playing an app, talking to so many people, etc)
Explorer Questions
1. The gray space is used to type out your code, the white space is the canvas that displays the code in image form.
2. This block brings you back to the gray coding space.
3. This block shows you the code to create shapes.
4. This block shows you the code to draw lines.
5. This block shows you the code to move your coded objects on the canvas.
6. This block shows you the code to add text.
7. This block shows you the code to create loops or how to get a random number.
8. This block shows you the code to add or change color.

Programmer Questions:
What does Syntax error mean? How do you fix it?
Syntax error means that in your code there is an error that is due to spelling, capitalization, or incorrect parameters. When you see this, Make Art will tell you where the syntax is. Go to the line where the error is and see if you can figure it out. You can use the buttons on the left to show what the correct code syntax should be. Also, if you click hint on the top right it will tell you what the correct code will be in a challenge.

In the challenge “Sunny Day” you have the completed code below. In the below code, explain which words can be changed to modify the outcome and which things have to stay the same.

In this code you can change “blue,” “yellow,” and “150” with “background,” “color,” and “circle” staying the same. You could also change “circle” to another shape if you wanted to change from a circle.

Creator Question
Make sure that each shape (square, circle, and rectangle) is represented in the drawing. Some learners may be able to write out the correct code to create each shape as well!
Explorer Questions

<table>
<thead>
<tr>
<th>Hack Minecraft Categories</th>
<th>What Do These Blocks Do?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Craft</td>
<td>These blocks creates and sets the Minecraft blocks.</td>
</tr>
<tr>
<td>Blocks</td>
<td>These blocks let you select a random block, wool, or assign a specific Minecraft block.</td>
</tr>
<tr>
<td>Text</td>
<td>These blocks let you create words or sentences with Minecraft blocks.</td>
</tr>
<tr>
<td>Position</td>
<td>These blocks let you modify the x, y, or z position of the Minecraft blocks.</td>
</tr>
<tr>
<td>Player</td>
<td>These blocks let you modify the position of the player in Hack Minecraft.</td>
</tr>
<tr>
<td>Control</td>
<td>These blocks allow you to create loops.</td>
</tr>
<tr>
<td>Logic</td>
<td>These blocks allow you to create conditional statements (if, else, then).</td>
</tr>
<tr>
<td>Math</td>
<td>These blocks allow you to use math numbers and operations.</td>
</tr>
<tr>
<td>Variables</td>
<td>These blocks allow you to create variables (words-strings, numbers, etc.).</td>
</tr>
</tbody>
</table>

Programmer Questions

1. In this image you are changing the x, y, and z for the block's position. The graph helps you see where the block is placed.

2. These code blocks say:
   - When "Q" is pressed
   - Set a 3 x 3 floor
   - To blocks that are flowers

3. This is the menu that comes up when you want to change the letter that is pressed to do something in Hack Minecraft.

Creator Question

Let learners be creative in what they decide to build. In order to build a house in the sky they would either need to be in the sky already or use the position blocks to move them around. Both solutions are acceptable!
Snake + Pong Answer Key

Start Ups

Color

Logic

Events

Actions

Numbers

Getters

Setters
Snake + Pong Answer Key

Explorer Question
--lives argument to change the number of lives

-s (s, m, f) argument to change the speed

f when used with the argument -s will make the speed fast

t argument to change the game theme

e enter editor mode

--help shows the help message

-b (s, m, l) argument to change the board size

Programmer Question
1. If the ball is hit, the ball size will increase by 30
2. If either player scores a goal then the ball size will be set to 15

Creator Question
Below is a rubric you can use to evaluate projects:

<table>
<thead>
<tr>
<th></th>
<th>All Learners</th>
<th>Most Learners</th>
<th>Some Learners</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Should have an image of the game drawn</td>
<td>• Should have an image of the game drawn</td>
<td>• Should have an image of the game drawn</td>
</tr>
<tr>
<td></td>
<td>• At least a three sentence explanation of the rules</td>
<td>• At least three to five sentence explanation of the rules</td>
<td>• At least a five plus sentence explanation of the rules</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• One to two labels explaining the game</td>
<td>• Three plus labels explaining the game</td>
</tr>
</tbody>
</table>
Explorer Question
For each Make Art challenge, the learner should clearly identify what (the word or code) and where (which line) they changed and how it affected their Make Art creation.

Programmer Question
The image below is what is coded in Make Art

Creator Question
Let learners be creative! You should see, at a minimum, the following:
- A simple drawing with at least three shapes (can be the same shape used multiple times)
- Three to five lines of code written out to create the image
Abstraction Answer Key

**Explorer Question**
For this learners should have a simple map drawn that does not have much detail. For example, a learner may draw a map, but they did not include all the details that would be in a level of their world.

**Programmer Question**
A model of a tiger would include the following:
- Tigers have two small ears
- Tigers have orange or white fur with black stripes
- Tigers have four legs
- Tigers have tails
- Tigers have whiskers
- Tigers have two eyes

**Creator Question**
Learners can draw a variety of tigers. Some can be white or orange. Ear and tail length can vary as well. Let them be creative and have fun!
ANSWER KEY

Explorer Questions

background tan: change the background to the color tan
stroke 0: set stroke size to 0
moveTo 80, 250: move to the position 80, 250 on the grid
color olivedrab: change the color to olivedrab
circle 30: create a circle with a radius of 30
for [1 .. 7]: create a for loop that will run 7 times
  move 50, 0: move to horizontal distance of 50 and the vertical distance of 0 (over 50 on the x axis and the same place on the y)
color olivedrab: change the color to olivedrab
circle 30: create a circle with a radius of 30
square 30: create a square size 30 on top of the circle in the same position
color darkgreen: change the color to darkgreen
circle 20: create a smaller circle with a radius of 20 on top of the square
square 20: create a smaller square size 20 on top of the square
moveTo 70, 240: move to the position 70, 240 on the grid
color black: change the color to black
ellipse 10, 5: create an ellipse with a width of 10 and a height of 5
color salmon: change the color to salmon
move -15, 20: move to horizontal distance of -15 and the vertical distance of 20
rectangle -20, 4: create a rectangle with a width of -20 and a height of 4

Programmer Question:

stroke black
color white
moveTo 0,0
#to create one row of squares
for [0 .. 10]
  square 50
  move 50,0

Creator Question:

color white
for x in [0 .. 10]
  for y in [0 .. 10]
    moveTo 50 * x, 50 * y
    square 50


ANSWER KEY

Explorer Question
The learner would need to decompose this into a few different steps:
Creating a 2 player game
Creating a condition to win after 10 points
Creating feature 1
Creating feature 2
After they have identified the features, they can explain what code blocks they used to create the feature.

Programmer Question
What crime was committed?
When was the crime committed?
Where was the crime committed?
What evidence is there?
Were any witnesses?
Have there recently been any similar crimes?

Creator Question
Learners should have some sort of drawing that is made up of multiple shapes. Below or next to it, they can have the shapes used to create the drawing.
Variables Answer Key

Explorer Question

Make Art Challenge: Variables

```plaintext
background navy
stroke 0
color red
ghostsize 100
move 0, -50
circle ghostsize
move ghostsize 0
square ghostsize 2
move ghostsize 2
color white
circle 20
color blue
circle 10
move ghostsize 0
color white
circle 20
color blue
circle 10
```

Make Pong Challenge: Crazy Ball
Programmer Question
In Make Art and Make Pong, you used variables but for different reasons. For each code set, can you explain what the variable was used for?

<table>
<thead>
<tr>
<th>Code Section</th>
<th>What was the variable used for?</th>
</tr>
</thead>
<tbody>
<tr>
<td>background navy stroke 0 color red ghostsize = 100 move 0, -50 circle ghostsize move -ghostsize, 0 square ghostsize * 2 move ghostsize / 2</td>
<td>The variable “ghostsize” was created and set to 100. It was used as a place holder for a number that was modified throughout the code.</td>
</tr>
<tr>
<td>Ball size Ball position X▼</td>
<td>The variable “ball size” was set to whatever the ball position was on the x-axis</td>
</tr>
</tbody>
</table>
### Variables Answer Key

**Creator Question**
Below is a rubric you can use to evaluate projects:

<table>
<thead>
<tr>
<th>All Learners</th>
<th>Most Learners</th>
<th>Some Learners</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Have one drawing that shows their game</td>
<td>Have one drawing that shows their game</td>
<td>Have one drawing that shows their game</td>
</tr>
<tr>
<td>• five sentences explaining their game</td>
<td>One to three labels in the drawing explaining their game</td>
<td>Four plus labels in the drawing explaining their game</td>
</tr>
<tr>
<td>• Sentences should clearly articulate the why and how of the project (why the game the picked and how they coded it)</td>
<td>Five sentences explaining their project</td>
<td>Five sentences explaining their project</td>
</tr>
<tr>
<td></td>
<td>Sentences should clearly articulate the why and how of the game (why the design the picked and how they coded it)</td>
<td>Sentences should clearly articulate the why and how of the game (why the design the picked and how they coded it)</td>
</tr>
<tr>
<td></td>
<td>One to two sentences will reference explicitly the code used</td>
<td>Three plus sentences will reference explicitly the code used</td>
</tr>
</tbody>
</table>
Numbers Answer Key

Name: ____________________________________________ Date: _____________________________

ANSWER KEY

Explorer Questions

Hack Minecraft Challenge: Treemendous

Make Pong Challenge: Make It Noisy
## Programmer Question

In Make Art and Make Pong, you used variables but for different reasons. For each code set, can you explain what the variable was used for?

<table>
<thead>
<tr>
<th>Code Section</th>
<th>What was the variable used for?</th>
</tr>
</thead>
</table>
| ![Random Numbers](image1.png) | X is a random number between 40 and 460  
Y is a random number between 150 and 250  
This is being used to move random around the grid |
| ![Ball Settings](image2.png) | Size of ball is 90  
Speed of the ball is 90 |

## Creator Question

Answers will vary. Some potential variables they may need:

- Temperature (high and low for the day)
- Wind speed
- Humidity
- Chance of rain (precipitation)
- Dew Point
Make Art Challenge: Hiking

```
stroke 0
color (opacity white, .1)
for j in [0 ... 250]:
    x = random 0, 500
    y = random 300, 500
    moveTo x, y
    circle 100
moveTo 250, 350
color red
bold true
italic true
font 'Bariol' 40
Text 'Hike the Blue Mountains at'
font 20
move 0, 90
Text 'Camp Kano!' 
```

Make Light Challenge: Scrolling Text

```
mytime = timenow()
message = 'Time: ' + mytime
light.scroll(message)
```
### Strings Answer Key

<table>
<thead>
<tr>
<th>Code Section</th>
<th>What was the variable used for?</th>
</tr>
</thead>
<tbody>
<tr>
<td>color red</td>
<td></td>
</tr>
<tr>
<td>bold true</td>
<td></td>
</tr>
<tr>
<td>italic true</td>
<td></td>
</tr>
<tr>
<td>font 'Bariol', 40</td>
<td>To set the font to Bariol</td>
</tr>
<tr>
<td>text 'Hike the Blue Mountains of'</td>
<td>To add text that said “Hike the Blue Mountains of”</td>
</tr>
<tr>
<td>font 90</td>
<td>To add text that said “Camp Kano!”</td>
</tr>
<tr>
<td>move 0, 90</td>
<td></td>
</tr>
<tr>
<td>text 'Camp Kano!'</td>
<td></td>
</tr>
<tr>
<td>mytime = timenow()</td>
<td>To add &quot;Time : &quot; to message that would show the current time</td>
</tr>
<tr>
<td>message = 'Time : ' + mytime</td>
<td></td>
</tr>
<tr>
<td>light.scroll(message)</td>
<td></td>
</tr>
</tbody>
</table>

### Creator Question

Let learners be creative! Each learner should:
- Draw an example of the code
- Have two to three labels or sentences explaining the modifications to the code and what it did
Hack Minecraft Challenge: Whammy Walk

Make Art Challenge: Rainbow Gradient
## Loops Answer Key

### Programmer Question

<table>
<thead>
<tr>
<th>Code Section</th>
<th>What was the loop used for?</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Code Section" /></td>
<td>The loop was used for numbers 0 - 32. Meaning, for a loop that ran 32 times, it created a circle that had a radius 10 times the numbers (0 - 32)</td>
</tr>
<tr>
<td><img src="image2" alt="Code Section" /></td>
<td>The loop was used for numbers 0 - 25. Meaning, for a loop that ran 25 times, it allowed the loop to use integers 0 - 25 up to 25 times in a loop for multiplication.</td>
</tr>
</tbody>
</table>

### Creator Question

Let learners be creative and think of unique ways to create art with loops. It is fine to look at loops that were already created in Make Art and modify them for their project!
**Conditionals Answer Key**

Learners should be able to start writing sentences that say “if this happens [ ] then do [ ].

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Conditional Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Level 7</strong></td>
<td>If the W key is pressed the ball is instantly rebounded</td>
</tr>
<tr>
<td><strong>Level 8</strong></td>
<td>If the W keys is pressed Player 1 instantly wins</td>
</tr>
<tr>
<td><strong>Level 9</strong></td>
<td>If the ball is hit Player 1, 2, and the ball will change a random color</td>
</tr>
<tr>
<td><strong>Level 10</strong></td>
<td>If Player 1's score = 10, Player 1 wins</td>
</tr>
<tr>
<td><strong>Level 11</strong></td>
<td>If the ball is hit make a plink sound</td>
</tr>
<tr>
<td><strong>Level 12</strong></td>
<td>If the ball position on the x axis is less than the middle point of x, rebound</td>
</tr>
<tr>
<td><strong>Level 13</strong></td>
<td>If Player 1 scores, the paddle size of Player 2 will be halved</td>
</tr>
<tr>
<td><strong>Level 14</strong></td>
<td>If the ball is hit, the speed of the ball will increase by 10</td>
</tr>
<tr>
<td><strong>Level 15</strong></td>
<td>If the ball is hit, the ball size will increase by 30. If either Player 1 or 2 score the ball size will be set to 15</td>
</tr>
<tr>
<td><strong>Level 16</strong></td>
<td>If W key is pressed Player 2 moves up. If S key is pressed move Player 2 down.</td>
</tr>
</tbody>
</table>

**Programmer Question**
For these questions learners should start identifying that the IF area is the condition that needs to be met and ELSE is what is done IF the condition is NOT MET. You will need to go one by one through these examples to see what learners will create!

**Creator Question**
Again, at this point learners should be able to identify IF as the key word for measuring a condition and ELSE occurs when the condition is not met. Look through each example to make sure that this concept has been met!
Booleans Answer Key

Explorer/Programmer Questions

Code Blocks

What Does the Boolean Evaluate?

If hit == true; if the ball is hit

If Player 1 score == 10, Player 1 wins

If the W key == pressed, Player 1 wins

If ball position x < middle point x, rebound

Creator Question

Let learners be creative! They can show a drawing of the modified Pong game. At a minimum there should be 2 - 3 labels explaining how they would code the modification.
## Debugging Answer Key

### Explorer Question

<table>
<thead>
<tr>
<th>Code Blocks</th>
<th>What the code is supposed to do</th>
<th>Why it doesn’t work</th>
</tr>
</thead>
</table>
| ![Code Blocks Image](image1.png) | The blocks should be the following:  
  - When “space” key pressed  
  - Turn 90 degrees  
  - Turn 90 degrees  
  - Turn 90 degrees  
  - Turn 90 degrees  
  
  $90 \times 4 = 360$, so when you hit the space key it moves so quickly it appears to stay still. | |
| ![Code Blocks Image](image2.png) | The blocks should be the following:  
  - When green flag clicked  
  - Say “Meow, meow, meow! For 2 secs”  
  - Repeat 3  
  - Play sound meow  
  
  Say “Meow” needs to be inside the repeat loop. | |
| ![Code Blocks Image](image3.png) | The blocks should be the following:  
  - When the sprite is clicked  
  - Move across the screen  
  - If it hits the wall it will reverse directions and repeat  
  
  Add a “set rotation style, left-right” and it will only turn left and right when it touches the edge. | |
Debugger Question
Debugging is the process of identifying errors and fixing them! In our lives we debug all the time. Examples can include when you edit a paper, when you are trying to improve your kicking, or even when you are debating an issue with a friend!

Creator Question
Creator Question
For this creative question learners can either:
1. Draw a picture and label it with explanations on how you would code it
2. Write step by step how the character would move
Learners should have at least five steps or instructions to make the character dance.
Explorer Questions
Fill out each section by reflecting on your Design Challenge Project!

For this section learners should have:
• A defined problem they are trying to solve
• A community (local, state, national, international) they want to work with
• An articulated idea on how they can solve the problem
• A list of a few ideas they had to solve the problem
• The winning idea and a prototype design for the idea
• A clear understanding of how they would test their prototype to get feedback

Programmer Question
When you created your prototype you had to code a portion of the project on Kano. What did you code? Why did you use that coding environment? What were the limitations of that coding environment for your project?

Below is a rubric you can use to evaluate projects:

<table>
<thead>
<tr>
<th>All Learners</th>
<th>Most Learners</th>
<th>Some Learners</th>
</tr>
</thead>
<tbody>
<tr>
<td>One sentence explaining what was coded</td>
<td>Two to three sentences explaining what was coded</td>
<td>Three plus sentences explaining what was coded</td>
</tr>
<tr>
<td>One sentence explaining what Kano App they used</td>
<td>Two to three sentences explaining what Kano App they used</td>
<td>Three plus sentences explaining what Kano App they used</td>
</tr>
<tr>
<td>One sentence explaining the pros and cons of that Kano App</td>
<td>An explanation of the sequence the code follows</td>
<td>An explanation of the sequence the code follows</td>
</tr>
<tr>
<td></td>
<td>Two to three sentences explaining the pros and cons of that Kano App</td>
<td>Three plus sentences explaining the pros and cons of that Kano App</td>
</tr>
<tr>
<td></td>
<td></td>
<td>An explanation for why their code is more efficient than other coding environments that could be used</td>
</tr>
</tbody>
</table>

Creator Question
Allow learners to be as creative as they want. There needs to be a drawing with clear labelling and details for improvements to the project.