

{developthefuture}

# Middle Year Student Project Resources

Designed for Grades 6–8



# Contents

## Introduction to the Program and Resources

- 4 **What is Develop the Future?**  
What is taught in the program?  
The Design Thinking Process Framework  
Who teaches the program?
- 5 **Develop the Future Timeline**
- 6 **Workshop Weeks**  
Workshop Timeline  
Preparing for Workshops
- 7 **Navigating the Pages Resources**
- 8 **Navigating the Keynote Resources**
- 9 **App and Playground Downloads**

## Foundations of App Design and Computational Thinking

### WEEK 1

- 10 **{design} What Are Apps For?**  
Handout: What Are Apps For?
- 13 **{code} Computational Thinking**
- 14 **Content Connection and Glossary**

### WEEK 2

- 15 **{design} Problem-finding**  
Handout: A Day in the Life  
Handout: Conducting Empathy Interviews
- 19 **{code} Commands**
- 20 **Content Connection and Glossary**

### WEEK 3

- 21 **{design} App Concept Ideation**  
Handout: App Concept Ideation
- 23 **{code} Functions**
- 24 **Content Connection and Glossary**

### WEEK 4

- 25 **App Idea Proposal**  
Handout: App Idea Proposal – The 5 Ws
- 28 **{code} Loops**
- 29 **Content Connection and Glossary**

## Designing App Features and Coding Personalization

### WEEK 5 – Workshop Week

- 30 **{design} Keynote Fundamentals**  
Workshop Logistical Prep
- 31 **{code} Catch-Up Week and Content Connection**

### WEEK 6 – Workshop Week

- 32 **{design} Initial Wireframing**  
Handout: Initial Wireframe Planning
- 34 **{code} About Me Workshop**  
Workshop Logistical Prep
- 35 **Content Connection and Glossary**

### WEEK 7

- 36 **{design} Feature-Finding**  
Handout: Feature-Finding
- 38 **{code} Variables**  
Workshop Logistical Prep
- 39 **Content Connection and Glossary**

### WEEK 8 – Workshop Week

- 40 **{design} Feature-Building in Keynote**  
Workshop Logistical Prep
- 41 **{code} Conditional Code**
- 42 **Content Connection and Glossary**

# Contents

---

## Sharing the Journey with Others

Coming  
Soon

### WEEK 9 – Workshop Week

- # **Presentation Skills**  
Workshop Logistical Prep
- # **{code} Optional Code Explorations**

### WEEK 10

- # **Presentation Preparation**  
Lorem ipsum

### WEEK 11

- # **Midterm Check-in**  
Midterm Check-in Evaluator Feedback Form  
Midterm Check-in Student Feedback Form

### WEEK 12

- # **{design} Accessibility and Inclusivity**  
Accessibility and Inclusivity Student Handout

## Looking Back and Moving Forward

Coming  
Soon

### WEEK 18

- # **Student Reflections**  
Student Reflection Form
- # **Stakeholder Feedback**  
Stakeholder Feedback Form

## Final Showcase Celebration

Coming  
Soon

### WEEK 13

- # **Showcase Overview & Preparation**  
Lorem ipsum

### WEEK 14-15

- # **Showcase Final Feedback Loops**  
Lorem ipsum

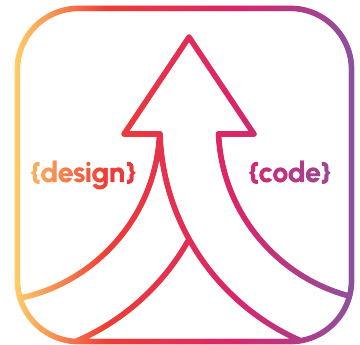
### WEEK 16-17

- # **Dress Rehearsal and Final Showcase**  
Lorem ipsum

# What is Develop the Future?

---

Develop the Future is a 17-week program designed for middle school students to explore the basics of app development using Swift Playgrounds and Keynote on iPad. Through engaging activities and hands-on projects, students will learn the basic principles of coding and prototype and present their own original app ideas. This program focuses on building essential skills in computational thinking, basic coding principles, UX/UI design, and effective presentation techniques, laying a strong foundation for future learning and creative problem-solving.



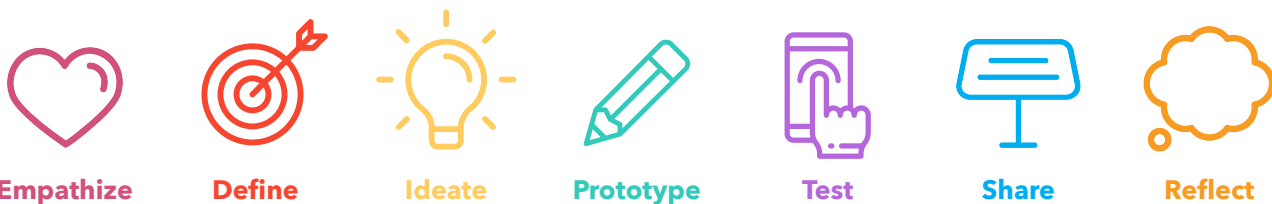
## Instruction

### What is taught in the program?

The Develop the Future program starts off split up into two skill-building streams: **{design}** and **{code}**. The initial weeks of the program help to form foundational understandings to the design process, computational thinking, and basic coding principles. The **{design}** and **{code}** streams eventually converge as students explore the impacts of design on code, and vice versa.

The entirety of the program is based heavily on the Design Thinking Process, which is a standard design framework that underpins technology curricula across Canada, such as the [Applied Design, Skills, and Technology curriculum in British Columbia](#) and the [Science and Technology Engineering Design Process in Ontario](#).

### The Design Thinking Process Framework



### Who teaches the program?

Develop the Future is nearly exclusively taught by the classroom teacher. If you are signed up for the CEC-supported program, you will have access to the supplemental workshops facilitated by outside presenters affiliated with the CEC. These workshops mark key milestones and follow the project's suggested timeline.

If you are not signed up for the CEC-supported program, the supplemental workshops will not be offered to you, but feel free to move through the project at your own pace and explore the provided resources to aid with instructional practice.

# Develop the Future Middle School Timeline

## {design}

## {code}

### Foundations of App Design and Computational Thinking



#### WEEK 1–4

What Are Apps?  
Problem-finding  
App Concept Ideation  
App Idea Proposal

#### WEEK 1–4

Computational Thinking Overview  
Commands  
Functions  
Loops

### Designing App Features and Coding Personalization



#### WEEK 5–8

Workshop: Keynote Fundamentals  
Initial Wireframing  
Feature Finding  
Workshop: Advanced Keynote – Feature Building

#### WEEK 5–8

Catch-up Week  
Workshop: All About Me  
Variables  
Conditional Code

### Sharing the Journey with Others



#### WEEK 9–12

Workshop: Presentation Skills  
Presentation Preparation  
Midterm Check-in  
Accessibility and Inclusivity

#### WEEK 9–11

Optional Swift Extras  
Logical Operators  
Functions With Parameters  
While Loops  
Expand All About Me

### Final Showcase Celebration



#### WEEK 13–17

Showcase Overview and Preparation  
Dress Rehearsals and Final Showcase

### Looking Back and Moving Forward



#### WEEK 18

Student Reflections  
Stakeholder Feedback

# Workshop Weeks

---

On particular weeks throughout the program, classes that are supported by the Career Education Council (CEC) will have the opportunity to participate in virtual workshops facilitated by industry experts. This can include Apple Distinguished Educators, Apple Professional Learning Providers, or Apple Retail (in-person field trip experience), depending on what each educator decides together with their CEC workshop coordinator.

The goal of guest speaker workshops is to:

- Develop critical foundational skills that support their app development journey
- Provide hands-on opportunities for student learning on iPad
- Link students to the perspectives of different industry experts, including a focus on design, code, and business

## Workshop Timeline

Week	Workshop Title	Required Apps
5	Keynote Fundamentals	Keynote
6	About Me in Swift Playgrounds	Swift Playgrounds with "About Me" playground downloaded inside of the app
8	Advanced Keynote – Feature-Building	Keynote
9	Presentation Skills	Pages

## Preparing for Workshops

There are a few things to consider when prepping for a virtual workshop. The checklist below outlines the items you should prepare, schedule, and check prior to each of your workshops:

- ☐ Confirm the dates and times of your workshops with your CEC coordinator.
- ☐ Confirm the video-conferencing platform you will be hosting the virtual call on (I.e. Zoom, Teams, Meet, etc). Ensure you have the appropriate app downloaded, logged in, and you have the call link readily available.
- ☐ Plan to host the session in a location that has a dependable internet connection. If this location requires booking, do so in advance.
- ☐ Ensure the space has the ability to project your computer onto a screen with full audiovisual capabilities. Test the set-up in advance to ensure both visuals and sound will be suitably adjusted for when students are together in the space.
- ☐ Book a class set of iPads for each workshop well in advance. Ensure the iPads are fully charged in the day leading up to the workshop.
- ☐ Download the latest version of the required app for the workshop in advance.

# Navigating the Pages Resources

## Overview

Provides a clear outline for the goal of the lesson or activity, and how it will be achieved

## Required Materials

Materials that will be needed in order to complete the listed learning activities

## Goals for Understanding

Lists specific learning outcomes to target within the lesson or activity

## Lesson Resources

Digital supplements to the lesson that can be used as anchoring visuals or guides as you progress

## Bolded Text

All text in a bold typeface will be defined in a glossary at the end of each week's content

## {design} What Are Apps For?

Week 1

### Overview

In a hands-on, small group activity, students will identify appropriate categories and sort apps to better understand the existing app market landscape. They will then explore the **purpose** behind commonly-used apps while linking their understandings to the **Design Thinking Process**, which is a strong influence within Canadian curriculum and modern teaching practices.

### Goals for Understanding

By the end of the lesson, students should be able to:

- Understand different categories that apps can be placed into (eg. shopping, health, social media, education, gaming, utilities, etc.)
- Identify the early phases of the Design Thinking process (**Empathize** and **Define**)
- Discuss the problems that various apps solve and possible app use cases (user scenarios)

### Required Materials

- Scrap slips of paper or sticky notes
- iPads (for exploration in small groups or individually)
- "What Are Apps For?" handout for students

### Lesson Resources

- Week 1 Develop the Future Middle Years slides resource (Keynote – slides 2-11)
- "What Are Apps For?" handout

### Learning Activities

Time	Activity
15 min	<b>Brainstorm and Categorize</b> (5 min) In small groups of 3-4 students, have students use scrap slips of paper or sticky notes to brainstorm as many apps as they can. Let them know that there should only be one app name per slip. (10 min) Prompt students to rearrange the slips in order to sort their app ideas into categories. Have them create category headers for their sort and lay them out on the table so others can read them. Afterward, do a short gallery walk or whole-group discussion so other groups can agree or disagree with the categories that students identified, and explore whether any were missed by creating a master list of categories together.
5 min	<b>Introduce the Design Thinking Process</b> Project the Design Thinking Process framework on Slide 6. Let them know that as you embark on this new project, you'll be moving through the phases: Empathize, Define, Ideate, Prototype, Test, Share. The next few weeks will be focused on <b>empathizing</b> with app users and <b>defining</b> problems that apps solve.
20 min	<b>Model and Explore</b> In a whole-class discussion, have students consider the Clock app for iPhone or iPad. Discuss different <b>use cases</b> of this app: how might people in different contexts use this app in unique ways? For example, what features would someone who is traveling for work rely on most? What about a professional baker? Brainstorm use cases together and use the ideas to model a rich and detailed response in the "What Are Apps For?" handout. Provide students with time to research and identify purpose and audience for at 2-4 apps, and complete the handout. Steer students away from social media and gaming to get them to expand their thinking.
5-10 min	<b>Discuss</b> Bring students together for final sharing. Either in small groups or as a whole class, check for understanding by having the students share out their responses for app purposes, <b>audience</b> , and unique features. Extend the conversation by having other students identify alternate use cases and/or purposes the app in question could have. Discuss a few examples together.  Close the lesson with this quote from Ellen Lipton on Slide 11: "Design is art that people use." A well-designed app will have a clear purpose and target audience.

## Learning Activities

These are the suggested activities for meeting the learning objectives at the top of each page. The Learning Activities include suggestions for instructional approaches (whole class, small group, partnered, individual), timing, and discussion prompts.

# Navigating the Keynote Resources

Each week's {design} and {coding} lessons Learning Activities align with a set of Keynote slide resources. These slides are designed to be student-facing and follow the suggested facilitation and pacing of the Learning Activities section for each lesson. The {design} and {coding} lessons for each week are embedded into the same deck for easier instructional organization and flow.

Teacher Instructions

1

What Are Apps For?

2

Design Activity Overview

3

Brainstorm

4

Categorize

5

Design Thinking Process

6

Consider The Click App

7

Think

8

What Are Apps For?

9

Discuss

10

"Design is art that people use."

11

Computational Thinking

12

## Teacher Instructions

Welcome to the Develop the Future Keynote resources! Each week's {design} and {code} lessons work to maximize the discussion links between the content, with the eventual convergence of the two streams later in the project.

Each set of Keynote slides are directly correlated to the lesson's Learning Activities for that week's {design} and {code} lessons.

Week 1 – Foundations of App Design & Computational Thinking

### Week 1 Slide Breakdown:

{design} Slides 2–11

{code} Slides 12–19

#### {design} What Are Apps For?

Week 1

**Overview**

In a hands-on, small group activity, students will identify appropriate categories and sort apps to better understand the existing app market landscape. They will then explore the **purpose** behind commonly-used apps while linking their understandings to the **Design Thinking Process**, which is a strong influence within Canadian curriculum and modern teaching practices.

**Goals for Understanding**

By the end of the lesson, students should be able to:

- Understand different categories that apps can be placed into (eg. shopping, health, social media, education, gaming, utilities, etc.)
- Identify the early phases of the Design Thinking process: **Imagine** and **Define**
- Discuss the problems that various apps solve and possible app use cases (user scenarios)

**Learning Activities**

Time	Activity
15 min	<b>Brainstorm and Categorize</b> (5 min) In small groups of 3-4 students, have students use scrap slips of paper or sticky notes to brainstorm as many apps as they can. Let them know that there should only be one app name per slip. (10 min) Prompt students to rearrange the slips in order to sort their app ideas into categories. Have them create category headers for their sort and lay them out on the table so others can see them. Afterwards, do a

**Required Materials**

- Scrap slips of paper or sticky notes
- Pads for exploration in small groups or individually
- "What Are Apps For?" handout for students

**Lesson Resources**

- Week 1 Develop the Future Middle Years slides resource (Worthington – slides 2-11)
- "What Are Apps For?" handout



# App and Playground Downloads

---

Students will need access to certain apps on the iPad in order to complete the {design} and {code} streams of the Develop the Future program. This page will help you to understand each app, its purpose, and any extra downloads or files needed prior to use.

---



## Keynote

This app is used to visually map out the user interfaces of students' app ideas, also known as a "wireframe." Students will learn to utilize several powerful tools within Keynote through two different virtual workshops within the program, and use their final wireframe prototype to share their app idea with others.

[Download here](#)

---



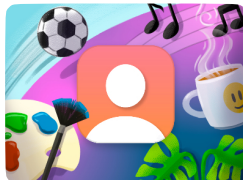
## Swift Playgrounds

This app plays host to several different coding challenges and puzzles called "playgrounds." Each playground needs to be downloaded inside of the app and is added to the main menu as its own file to be worked on. There are three playgrounds that need to be downloaded to be used for the program and the week they need to be do:

[Download here](#) \*



**Learn to Code 1**  
(Week 2-8)



**About Me**  
(Week 5-6)



**Learn to Code 2**  
(Week 7)

\* Need support with downloading individual playgrounds in the app? [Watch this video](#)

---



## Pages

This app is used in the Presentation Skills workshop for students to consider their talking points when telling their app story. It can also be used to have students complete digital handouts associated with various weekly lessons. The .pages files for the black-line masters can be found on the [Develop the Future website - Middle Year Curriculum](#).

[Download here](#)

### Overview

In a hands-on, small group activity, students will identify appropriate categories and sort apps to better understand the existing app market landscape. They will then explore the **purpose** behind commonly-used apps while linking their understandings to the **Design Thinking Process**, which is a strong influence within Canadian curriculum and modern teaching practices.

### Goals for Understanding

By the end of the lesson, students should be able to:

- Understand different categories that apps can be placed into (eg. shopping, health, social media, education, gaming, utilities, etc.)
- Identify the early phases of the Design Thinking process (**Empathize** and **Define**)
- Discuss the problems that various apps solve and possible app use cases (user scenarios)

### Required Materials

- Scrap slips of paper or sticky notes
- iPads (for exploration in small groups or individually)
- "What Are Apps For?" handout for students

### Lesson Resources

- Week 1 Develop the Future Middle Years slides resource (Keynote – slides 2-11)
- "What Are Apps For?" handout


### Learning Activities

Time	Activity
15 min	<b>Brainstorm and Categorize</b> (5 min) In small groups of 3–4 students, have students use scrap slips of paper or sticky notes to brainstorm as many apps as they can. Let them know that there should only be one app name per slip. (10 min) Prompt students to rearrange the slips in order to sort their app ideas into categories. Have them create category headers for their sort and lay them out on the table so others can read them. Afterward, do a short gallery walk or whole-group discussion so other groups can agree or disagree with the categories that students identified, and explore whether any were missed by creating a master list of categories together.
5 min	<b>Introduce the Design Thinking Process</b> Project the Design Thinking Process framework on Slide 6. Let them know that as you embark on this new project, you'll be moving through the phases: Empathize, Define, Ideate, Prototype, Test, Share. The next few weeks will be focused on <b>empathizing</b> with app users and <b>defining</b> problems that apps solve.
20 min	<b>Model and Explore</b> In a whole-class discussion, have students consider the Clock app for iPhone or iPad. Discuss different <b>use cases</b> of this app: how might people in different contexts use this app in unique ways? For example, what features would someone who is traveling for work rely on most? What about a professional baker? Brainstorm use cases together and use the ideas to model a rich and detailed response in the "What Are Apps For?" handout. Provide students with time to research and identify purpose and audience for at 2–4 apps, and complete the handout. Steer students away from social media and gaming to get them to expand their thinking.
5–10 min	<b>Discuss</b> Bring students together for final sharing. Either in small groups or as a whole class, check for understanding by having the students share out their responses for app purposes, <b>audience</b> , and unique features. Extend the conversation by having other students identify alternate use cases and/or purposes the app in question could have. Discuss a few examples together.  Close the lesson with this quote from Ellen Lipton on Slide 11: "Design is art that people use." A well-designed app will have a clear purpose and target audience.


Name: \_\_\_\_\_

**Instructions:** Use an iPad to find two apps (or more, if you choose to!). Add the app name and logo, and take time to consider multiple possible use cases for the app. Fill in the app's purpose and intended audience (there can be several!), and list the features or abilities that set this app apart from its competition.

### App #1


App Logo and Name	Purpose (what does your app do?)	Audience (who is your app intended for?)
	_____	_____
	_____	_____
	_____	_____
	_____	_____
	_____	_____
<b>Originality</b> (what sets this app apart from others?)		
_____		
_____		

### App #2


App Logo and Name	Purpose (what does your app do?)	Audience (who is your app intended for?)
	_____	_____
	_____	_____
	_____	_____
	_____	_____
	_____	_____
<b>Originality</b> (what sets this app apart from others?)		
_____		
_____		

Name: \_\_\_\_\_

### App #3

App Logo and Name	Purpose (what does your app do?)	Audience (who is your app intended for?)
	_____	_____
	_____	_____
	_____	_____
	_____	_____
Originality (what sets this app apart from others?)		
_____		
_____		

### App #4

App Logo and Name	Purpose (what does your app do?)	Audience (who is your app intended for?)
	_____	_____
	_____	_____
	_____	_____
	_____	_____
Originality (what sets this app apart from others?)		
_____		
_____		

### Overview

The purpose of this lesson is to introduce students to the basics of **computational thinking**. Students will learn how to break down complex problems, recognize patterns, create simple algorithms, and understand how computational thinking relates to coding. This lesson sets the foundation for problem-solving, logical reasoning, and understanding how computers process information.

### Goals for Understanding

By the end of the lesson, students should be able to:

- Decompose a task into smaller, manageable parts (**decomposition**)
- Identify recurring patterns or similarities in tasks (**pattern recognition**)
- Understand the basics of creating step-by-step instructions to solve problems (**algorithm design**)

### Required Materials

- Slips of paper or sticky notes
- Classroom objects to make a mock PB&J sandwich, like paper (bread), sticky notes (spreads), a pen holder or basket (jars), a whiteboard marker (knife)

### Lesson Resources

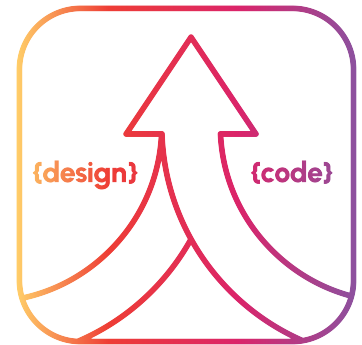
- Week 1 Develop the Future Middle Years slides resource (Keynote – slides 12-19)

### Learning Activities

Time	Activity
10 min	<b>How to Make a PB&amp;J</b> Making a sandwich is a task that takes quite a few smaller steps to achieve the final goal. This learning activity will mimic some initial aspects of computational thinking including breaking down a larger task into smaller parts. In small groups, prompt students to come up with the steps that they would need to take to make a peanut butter and jelly sandwich. They can collaborate to write each step on a slip of paper or a sticky note.
10 min	<b>How a Robot “Makes” a PB&amp;J</b> Bring students back together and flip to Slide 15 on the Keynote resource, which outlines a sample of steps to make a PB&J. Let students know you are acting as a robot, meaning you need to follow these instructions <i>exactly</i> , using objects in the classroom to stand in for the making of the sandwich. Have a student read you the instructions on the screen as you complete the actions. Try not to “fill in gaps” according to how the instructions are written so you can demonstrate how each step should be more literally worded in order to complete a computational task. For example, for the first two steps, you might: <ol style="list-style-type: none"><li>1. <i>Get bread, peanut butter, jelly, and a knife</i> - model gathering the materials</li><li>2. <i>Spread peanut butter on one side of bread</i> - rub the “peanut butter jar” against the “bread”</li></ol> And so on. Make it entertaining—and frustrating!—so there is a clear demonstration of how the instructions need be very clear and encompass <b>all</b> actions taken by the robot. You may choose to have students view the “PB&J Exact Instructions” video resource to drive the point home.
20 min	<b>The Students’ Turn</b> After you’ve modelled this, instruct each group to join up with another group. Have them gather some materials to make their mock sandwiches, and have one student from each group act as the “robot” who will go through the instructions from the other group. If there is time, allow them to adjust their instructions and try again.
10 min	<b>Debrief and Wrap-Up</b> Bring students back together to share their experiences. How did breaking down the task make it easier for the robot to complete it? What were some successes? Frustrations? Round out the discussion by linking the concept of computational thinking to the task. Point out how all of the technology we use - including the apps that were explored in the {design} activity - are coded using step-by-step instructions to complete a task.

## Connecting {design} and {code}

Exploring apps requires students to recognize that apps operate through logical processes. By applying computational thinking to break down tasks into manageable steps, students can eventually design apps with clarity and purpose, ensuring every element and user interaction is logically planned.



## Glossary

### {design}

#### Audience

The intended group of people that an app is designed for

#### Define

The portion of the Design Thinking Process in which designers identify specific problems or pain points for potential users, and use the problem as a foundation toward designing a new solution

#### Design Thinking Process

An iterative and non-linear process in which designers identify problems and create meaningful solutions through empathizing with potential users. This process forms the basis of BC's ADST curriculum

#### Empathize

The portion of the Design Thinking Process in which designers attempt to understand the perspectives of others and their experiences as a foundational cornerstone for the design process

#### Purpose

Identifying a significant reason for app design that promotes real-world problem-solving rather than entertainment

#### Use Cases

The different ways that an app can be utilized, often dependent on the user's unique perspective or role

### {code}

#### Algorithm Design

Creating step-by-step instructions to solve a problem or complete a task

#### Computational Thinking

A way of solving problems by breaking them down, finding patterns, and designing solutions that a computer can execute

#### Decomposition

Breaking down a complex problem into smaller, easier-to-manage parts

#### Pattern Recognition

Identifying repeated elements or trends to help solve problems or predict outcomes

### Overview

When designing a mobile technology solution, **finding a problem** to design for is critical. People experience a series of small problems on a daily basis, sometimes without even realizing it. In this lesson, students will utilize the **"Five Whys" questioning technique** to identify the root causes of day-to-day challenges that themselves or others may encounter. They will conduct various **"empathy interviews"** with people they know while recording their responses.

### Goals for Understanding

By the end of the lesson, students should be able to:

- Use critical thinking and questioning skills to identify a problem to design for
- Utilize the "Five Whys" questioning technique to strengthen their ability to identify root causes to problems
- Practice using the technique by interviewing peers
- Be prepared to conduct empathy interviews with others that they know and record their findings

### Required Materials

- iPad with Pages app installed  
or "Problem-finding" handout (3 pages)

### Lesson Resources

- Week 2 Develop the Future Middle Years slides resource (Keynote – slides 2-16)
- "Problem-finding" handout

### Learning Activities

Time	Activity
10 min	<b>Problem-finding in a Typical Day</b> Provide students with a copy of the "Day in the Life" handout either digitally or on paper. Give students 5 minutes to jot down the details of a typical day in their life including their activities and the time of day they occur. When complete, get them to star common parts of their day where they encounter some kind of struggle. This could be activities taking too much time, obstacles to completing their goals, getting side-tracked, etc.
10 min	<b>The Five Whys Questioning Technique</b> Have students get in partners or small groups and look over each other's "Day in the Life" outlines. Their goal will be to find out the root cause of their partner's struggles. Provide them with the <b>"Five Whys" interrogation technique</b> prior to launching them into discussion. In this technique, each answer that a student provides should be met with a "why" from the other. Through this activity, students will essentially complete a mini oral interview with one another to <b>define</b> the problem the other is experiencing. Give each group has ~5 minutes each to share.
5-10 min	<b>Why Ask Why?</b> With the whole group, discuss the discoveries that different groups made about each other's problems. Discuss: how does the Five Whys technique help us to identify deeper problems? Who else could we talk to in order to find new problems? Discuss how understanding why problems exist help us to <b>empathize</b> with others and to better define their problems, which are two key steps of the Design Thinking Process.
15 min	<b>Empathy Interviews for Perspective-taking</b> Explain that students are going to use the "Five Whys" technique with 2-3 more people in their lives to identify typical problems they experience. These root problems will support the identification of areas that their app concepts could focus on later.  Students should use the "Conducting Empathy Interviews" pages for the next part of the activity. On the handout, they will need to indicate who they will be interviewing and then create a starting question to guide the conversation. This question could focus on broad daily experiences like they did in the "Day in the Life" activity, or they can hone in on a specific area such as a special hobby or interest. They will write down the perceived problem of that person and use the "Five Whys" to dig deeper into the root cause of the problem.

Name: \_\_\_\_\_

### Warm-up Activity: A Day in the Life

**Instructions:** Jot down the activities/events you encounter in a typical day. Include the time and some details about what is going on during that activity or event. For example, you might wake up at 7:30AM, eat breakfast from 7:35-7:50AM, brush your teeth at 7:55AM, and so on.

When you're finished, place a star ★ beside the items that are challenging. This can be for any reason — perhaps they take too long, you encounter obstacles, you tend to get side-tracked, or you tend to avoid the task all together! The starred items will help to spark a conversation later on.

Time	Activity

Time	Activity

**Next:** Find a partner. Review their Day in the Life outline and identify the “starred” events in their day. Use the **Five Whys questioning technique** to attempt to find the root cause of their problems.



Name: \_\_\_\_\_

## Conducting Empathy Interviews

**Instructions:** You will be interviewing 2-3 people in your life to identify the challenges they experience on a day-to-day basis. Your interviews can focus on everyday living or you can hone in on a special interest area such as an activity or hobby. You can use the “Five Whys” technique to dig into the root cause of each challenge.

First, identify who you will be interviewing. Next, consider what starting question to use to frame your conversation and figure out what their perceived problem is. Finally, have the conversation and use “Five Whys” to boil their problem down to the root cause! **Hint:** You might not use ALL five of your “why” questions.

Example
<b>Interview #1:</b> Mr. Johal
<b>Relationship to You:</b> Soccer coach
<b>Starting Question:</b> What is the main challenge you experience as a youth soccer coach?
<b>Perceived Problem:</b> Mr. Johal has difficulty figuring out which drills he should run in his hour-long weekly practices.
<b>The Whys:</b>  1 Why? – He’s not sure how to narrow down the drills that he can find online.  2 Why? – It is overwhelming, even when searching for drills on the soccer association site.  3 Why? – The soccer association resources aren’t organized well.  4 Why? – The website does not organize information on drills by skill and age.  5 (This “why” was not needed in this case)
<b>Actual Problem:</b> The soccer association needs to provide an organized and searchable selection of soccer drills for volunteer coaches.

<b>Interview #1:</b>
<b>Relationship to You:</b>
<b>Starting Question:</b>
<b>Perceived Problem:</b>
<b>The Whys:</b>  1  2  3  4  5
<b>Actual Problem:</b>

Name: \_\_\_\_\_

### Conducting Empathy Interviews

Interview #2:
Relationship to You:
Starting Question:
Perceived Problem:
The Whys:
1
2
3
4
5
Actual Problem:

Interview #3:
Relationship to You:
Starting Question:
Perceived Problem:
The Whys:
1
2
3
4
5
Actual Problem:

### Overview

How can we use code to have our devices perform actions for us? In this lesson, students will be introduced to the basics of coding using **Swift Playgrounds**. They will explore the fundamental programming coding concept of **commands** and how they can relate to **computational thinking** through explorative gameplay and puzzles by controlling an interactive character called Byte.

### Goals for Understanding

By the end of the lesson, students should be able to:

- Use commands in the correct order to work through coding puzzles (**sequencing**)
- Improve **problem-solving skills** and identifying errors (**debugging**)
- Create and use **commands** like `moveForward()` for efficient coding (**task execution**)
- Approach programming with **logical thinking**

### Required Materials

- iPads with the [Swift Playgrounds](#) app installed
- Learn to Code 1 Playground downloaded in the Swift Playgrounds app

### Lesson Resources

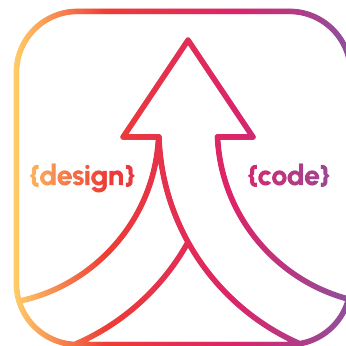
- [Everyone Can Code Puzzle Book](#) (pages 8–17)
- Week 2 - Develop the Future Middle Years slides resource (Keynote - slides 17-26)

### Learning Activities

Time	Activity
5 min	<b>Commands</b> Begin the lesson with a recap of the PB&J activity, discussing why it was necessary to move from more basic to more detailed instructions as the activity progressed. Reveal that students were actually practicing the concept of “commands” – a basic coding principle – which they’ll explore on the iPads using Swift Playgrounds and Apple’s Everyone Can Code resource.
35 min	<b>Swift Playgrounds</b> Distribute iPads to students and have them open Swift Playgrounds. Have them download the Learn to Code 1 playground by adding it (+) to the main page of the app. To save time, you may wish to download this to the iPads beforehand (a tutorial video can be found <a href="#">here</a> ). If you are in a shared device environment, you may also wish to have students document which iPad they use for continuity in the next lesson.  For this section, the accompanying Keynote slides resource provides a very valuable, sequential visual guide: <ul style="list-style-type: none"><li>• Use Slide 21 to introduce the concept of commands, which are instructions that are written in a specific sequence in order to carry out a task.</li><li>• On Slide 22, play the accompanying video, which demonstrates how to write and sequence commands in Swift Playgrounds. This video connects commands to computational thinking while explaining how to solve puzzles within the Swift Playgrounds interface. Encourage students to follow along on their iPads, guiding them to start the first puzzle.</li></ul> Encouraging Exploration: After the video, prompt students to explore additional puzzles within the commands section of Learn to Code 1. Circulate the room to assist and ensure each student understands how to access and work through the playgrounds independently.
5 min	<b>Reflect and Discuss</b> Bring students back together to reflect on their app creation process. How do coding commands relate to identifying and solving problems? What were some successes they experienced? What challenges did they face? Conclude the discussion by emphasizing the connection between computational thinking and app development, highlighting how coding involves creating step-by-step instructions to address specific problems. Use the questions on Slide 25 to guide the conversation.

## Connecting {design} and {code}

Just as commands drive actions in coding, problem finding helps define how an app should respond to user actions. Students translate this into wireframes, linking identified problems to specific app features and commands, ensuring every element and user interaction is logically planned.



## Glossary

### {design}

#### Define

The portion of the Design Thinking Process in which designers identify specific problems or pain points for potential users, and use the problem as a foundation toward designing a new solution.

#### Empathy Interviews

Inquisitive discussions between two people, where the interviewer works to better understand the experience of the interviewee.

#### Empathize

The portion of the Design Thinking Process where the designer attempts to immerse themselves into the thoughts, feelings, and viewpoints of another person in order to better understand their perspective.

#### Finding a Problem

Using questioning techniques to identify the root problems within individuals' personal experiences, by which technology could potentially provide a solution.

#### Five Whys Questioning Technique

An interrogation technique that uses a series of "why?" questions to get to the root of a perceived problem. Starting with an initial problem, the questioner asks "Why?" to each new response, up to five times, with the goal of uncovering what is going on beneath the surface issue.

### {code}

#### Commands

Instructions that tell the computer to perform specific actions.

#### Problem-solving

The process of finding solutions to coding challenges.

#### Sequencing

The arrangement of commands in the correct order for execution

#### Computational Thinking

A problem-solving process that involves breaking down complex problems into manageable parts, using logic and reasoning to develop algorithms, and applying these methods to coding tasks.

#### Debugging

The process of identifying and fixing errors or bugs in code to ensure it runs correctly.

#### Swift Playgrounds

An interactive development environment created by Apple for learning and experimenting with the Swift programming language. It provides a user-friendly interface that allows users to write Swift code, see results in real time.

### Overview

This **ideation phase** calls for students to capture overviews of app concepts that may be worthy to pursue. No ideas are “bad ideas” here: some of the best app concepts come from peer collaboration, commentary, and discussions. Students will be pushed to focus on app ideas that solve **real-world problems** rather than social media or games. By the end of this session, students should identify the app concept that they want to take on as their project.

### Goals for Understanding

By the end of the lesson, students should be able to:

- Generate multiple app concept ideas, including defining the problem the app idea will solve for, **identifying the audience** it's designed for, and brainstorming **app functionalities**
- Give and receive feedback through peer and teacher collaborative discussions
- Refine and narrow down ideas

### Required Materials

- “App Concept Ideation” handouts (single-sided - 2-3 copies per student)
- Sticky notes
- Students’ completed Empathy Interview handouts

### Lesson Resources

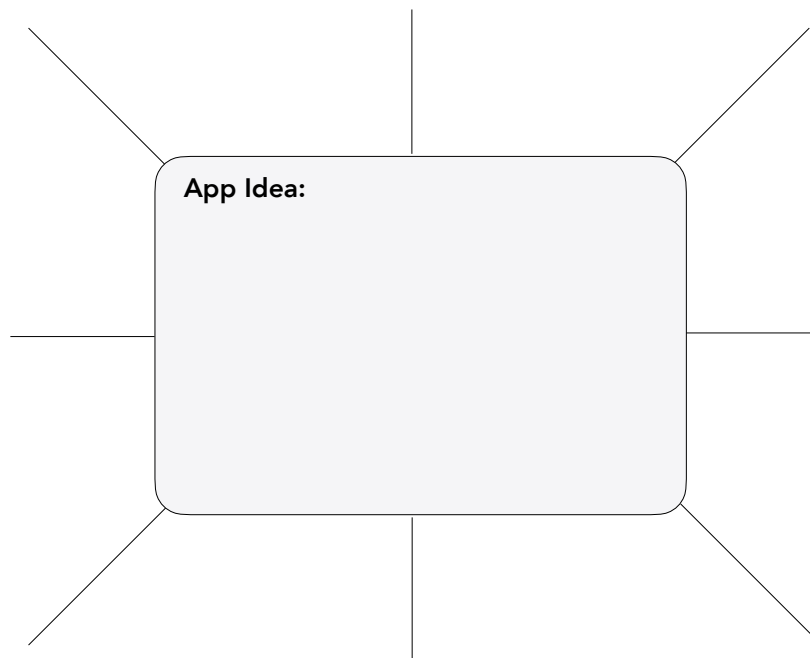
- Week 3 Develop the Future Middle Years slides resource (Keynote – slides 2-10)
- “App Concept Ideation” handout

### Learning Activities

Time	Activity
5 min	<b>Defining Problems, Ideation, and the Design Thinking Framework</b> Have students gather their completed Empathy Interview notes from Week 2. Hold a short discussion about the root problems they found through their interviews. Explain that you will be using those problems to brainstorm ideas for new apps. The empathy interviews helped to <b>define</b> a problem, and now they are moving to <b>ideation</b> in the Design Thinking Framework. Hand out a single-sided copy of the “App Concept Ideation” handout.
30 min	<b>Brainstorm</b> Encourage students to use their empathy interview notes to support the ideation process - what root problems did they identify through their interviews? What problems might they be able to solve for someone through technology? They will be asked to <u>brainstorm 3 or more app ideas</u> throughout this activity, including defining the problem that the app would solve for, and who the intended audience of the app would be. Let students know that even ideas that they think are “bad” should be jotted down as they may flourish into something bigger later on! There is no commitment to the ideas put on the pages in this activity. For the teacher, this is a key time to steer students away from social media or gaming ideas and to focus more on real-world problems.
10-15 min	<b>Gallery Walk and Comment</b> Once students have finished 2-3 ideation pages, have them lay the sheets out on their tables or desks so each idea is visible. Distribute sticky notes to students and let them know they are going to be going for a gallery walk to observe the ideas of their classmates. They will use the sticky notes to ask questions, make meaningful comments, or to suggest different features that the author may not have thought of. You may wish to scaffold in a discussion about what makes a meaningful question or comment versus a “dead-end” one, and how to keep the conversation positive - an exemplar can be found on Slide 7 in the Keynote slides for this week.
5–10 min	<b>Review Comments and Identify Idea to Pursue</b> Have students return to their seat to review the comments and questions that they received. You can close the lesson by giving them a few minutes to amend their ideas on their handouts by answering questions or adding feature ideas suggested by their peers.

Name: \_\_\_\_\_

**Instructions:** Use the mind map frame below to brainstorm ideas for an app concept. Jot down ideas you have about what the app will do, possible features, and different ways in which it could be used. In the question boxes below the mind map, identify the problem that the app would solve and the intended audience for the app.



What problem does this app solve?

Who is the intended audience for this app idea?

### Overview

This lesson introduces the concept of functions and how they help to organize code. Students will build on their understanding of **commands**, learning how to create reusable pieces of code that can perform specific tasks multiple times, called **functions**.

### Goals for Understanding

By the end of the lesson, students should be able to:

- Create reusable code blocks in order to repeat commands (functions)
- Apply learning to solve coding challenges effectively
- Practice breaking complex problems into smaller, reusable **components** (**modular thinking**)

### Required Materials

- iPads with Swift Playgrounds installed
- "Learn to Code 1" downloaded in the Swift Playgrounds app

### Lesson Resources

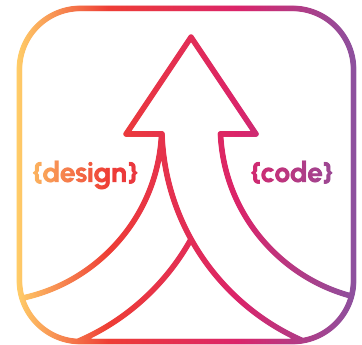
- [Everyone Can Code Puzzle Book](#) (pages 18–24)
- Week 3 - Develop the Future Middle Years slides resource (Keynote – slides 11-22)

### Learning Activities

Time	Activity
10 min	<b>Exploring the Concept of Functions</b> Begin the lesson by choosing a routine task, like brushing your teeth, and have students consider the commands a person would have to follow if they coded it. Write down their ideas on the board. Ask students whether it would be more efficient to list every single instruction every time they brush their teeth, or if it makes more sense to group instructions together into a single category called <code>brushTeeth()</code> . In coding, the grouping of multiple commands to complete a larger task is called a function. After the initial set-up, functions provide a singular line of code that has a larger group of steps categorized within it that could be repeated again at a later time.
40 min	<b>Swift Playgrounds</b> Distribute iPads to students and have them open Swift Playgrounds and open the Learn to Code 1 playground. If students were working on a specific device during the Commands lesson, have them return to that device and navigate to the Functions section (see Slide 18 for details).  For this section, the accompanying Keynote slides resource provides a very valuable, sequential visual guide: <ul style="list-style-type: none"><li>• Use Slides 14-15 to learn the concept of functions and link the content to discussions in the previous activity.</li><li>• On Slide 16, play the accompanying video, which reinforces the concept of functions. Encourage students to follow along on their iPads, guiding them to try the first functions puzzle.</li></ul> After the video, prompt students to explore additional puzzles within the Functions section of Learn to Code 1. Circulate the room to assist and ensure each student understands how to access and work through the playgrounds independently.  The playground saves student progress when they close out the app - however if they need to access a specific module, they can use the navigation on the top left of the playground and select the playground from the menu.
5-10 min	<b>Reflection</b> Use Slide 21 to support facilitation of a group reflection on the concept of functions. Focus the discussion on why grouping steps into functions is helpful and how they make code easier to write and understand, especially when steps are reused throughout a process.

## Connecting {design} and {code}

Functions in coding perform specific tasks, just like app features serve specific purposes. During ideation, students can visualize app features as components working together, much like functions in code solving a problem.



## Glossary

### {design}

#### Ideation Phase / Ideate

Part of the Design Thinking Framework where the designer brainstorms multiple ideas and approaches to create a solution for the problem

#### Real-world problems

Issues or challenges that present themselves in the "real world": this could be day-to-day life or within a specific activity or hobby

#### Define

Part of the Design Thinking Framework where the designer works to identify and break down a problem, issue, or challenge, to its root cause

#### Identifying the audience

Using perspective-taking to figure out who potential users would be within a given design or idea

#### App Functionalities

What an app can do; this can include app features (e.g. search, history, filters, etc.)

### {code}

#### Calling Functions

The process of invoking or executing a previously defined function in order to perform the specific task it encapsulates

#### Code Management

Techniques for organizing and maintaining code effectively

#### Commands

Instructions that tell the computer to perform specific actions

#### Components

Individual parts or elements of a program or system that work together to create a larger functionality; in coding, these can refer to functions, classes, or modules that contribute to the overall structure of the code

#### Functions

Reusable blocks of code that perform a specific task

#### Modular Thinking

Breaking complex problems into smaller, manageable parts



### Overview

By this point, students should have identified a meaningful app idea that they wish to pursue. This week will focus on proposing their idea through an analysis of the “5 Ws.” They will consider a series of questions about their idea so they are able to start imagining their **app design** and **app functionalities**.

### Goals for Understanding

By the end of the lesson, students should be able to:

- Identify if they are working individually or in a partnership (based on whether teacher wishes to create this offering)
- Finalize their app idea
- Write a **proposal** that clearly communicates the app’s purpose, audience, functionalities, and more

### Required Materials

- iPad with Pages app installed  
OR “App Idea Proposal - The 5Ws” handout

### Lesson Resources

- Week 4 Develop the Future Middle Years slides resource (Keynote Slides 2-9)
- “App Idea Proposal – The 5 Ws” handout

### Learning Activities

Time	Activity
10 min	<b>Review App Ideating Session</b> Start this session by reviewing students’ App Concept Ideation handouts from the previous week. By the end of the ideating session, they should have gone through one round of feedback (via a sticky note gallery walk) about the various app ideas that they came up with.  Let them know that it’s time to make a selection, and give them a few minutes to identify the idea that they wish to pursue for their project. If you are allowing partnered work, discuss the benefits and drawbacks of working individually versus in a partnership for a long-term project. If students wish to work in a partner pair, allow them to select their app focus together as they will need to discuss how to hone and combine their original ideas.
5-10 min	<b>Overview of the App Proposal Process</b> After students have made their selections, provide them with a copy of the “App Idea Proposal - The 5 Ws” handout. Take some time to go over the questions in depth, and encourage students to provide as much detail in their responses as possible. This assignment is often utilized as an assessable component of the project, so share with students if you intend to assess their app proposals (this may also result in better efforts!).
30 min	<b>Student Work Block</b> Provide students with ample time to complete their app proposals and offer feedback throughout the block to support the request of responding in detail. The more information that students have in their app proposals, the easier it is to support them on their app journey later on, as this assignment acts as the initial planning document for their project!
5 min	<b>App Proposal Submissions</b> If students are finishing up their proposals, allow them to submit their work and provide feedback if you wish. If they require more time, provide them with a due date sometime within the week. If you are a CEC-supported teacher, finalizing their app proposal will be a valuable asset as they move through the upcoming “Keynote for Wireframing & Design” workshop in Week 5.

Name: \_\_\_\_\_

**Instructions:** App development goes beyond learning to design, code, and market your app. It also sharpens your brainstorming and social entrepreneurship skills to help you be more successful in the future. Write clear and detailed responses as you consider the following questions that cover the what, why, where, when, who, and how of your app idea. This handout will serve as your formal proposal of your idea to your teacher.

## The 5 Ws

### Who is in your group?

Ask your teacher if you may work independently or in a partnership for this project.

### What is a name idea for your app?

If you have a name idea, add it here. If you don't, let your teacher know you're not sure yet!

### Why? - What purpose does this app serve and why do you want to create it?

Building something that solves a problem or makes a task easier makes a positive impact on the world. Consider: why should your app exist? Why does your audience need this app? What problem does it solve or what positive impact does it make on peoples' lives?

### What? - Tell us about your app idea.

What does your app do? Tell about the basic concepts of your idea - what's included?

Name: \_\_\_\_\_

### Who? - Describe the people who are most likely to use your app.

Knowing as much about your audience as possible will help you create a solution more people want with a better user experience. Is your audience of a certain age or gender affiliation? Do they live in a particular area, and/or face a similar challenge?

### Where? - Where are people going to be using your app?

Understanding when and where your audience will use your app will help you ensure it works as intended. Will your app be localized only to being used in your city or province? Or does it involve specific places like the grocery store or a movie theatre? Can you use your app anywhere or will it need a Wi-Fi connection?

### How? - How does your app work and what features does it have?

Thinking through the steps of how your audience will use and interact with your app will help you determine what iOS features are essential to include.

For example, will it require certain features such as GPS location, Wi-Fi connections, Bluetooth, speech recognition, use of the camera, notifications, etc? Answer based on what you know about the available features on an iPhone or iPad.

### Overview

In this lesson, students will dive into the concept of **loops** in programming using Swift Playgrounds. They will understand how loops help in repeating actions efficiently and apply this concept to control the LoopBot robot. The lesson aims to enhance students' problem-solving skills, efficiency in coding, and understanding of how loops simplify repetitive tasks in real-world applications.

### Goals for Understanding

By the end of the lesson, students should be able to:

- Repeat tasks efficiently and understand how **iteration** enables them to perform a set of instructions multiple times (loops)
- Solve problems that require repeated actions (**for loops**)
- Recognize how loops improve efficiency by making code more manageable

### Required Materials

- iPads with Swift Playgrounds installed
- "Learn to Code 1" downloaded in the Swift Playgrounds app

### Lesson Resources

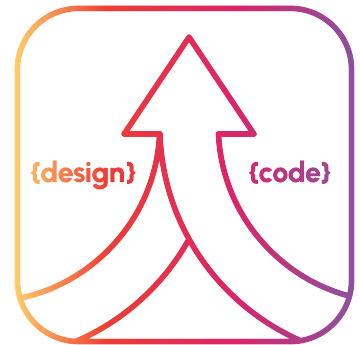
- [Everyone Can Code Puzzle Book](#) (pages 18–27)
- Week 4 – Develop the Future Middle Years slides resource (Keynote – slides 10-21)

### Learning Activities

Time	Activity
10 min	<b>Introduction to Loops</b>  Begin by introducing loops as a way to repeat a set of instructions without rewriting them multiple times. Provide a simple analogy: "If you want to take five steps forward, instead of saying 'step forward' five separate times, you can say, 'repeat step forward five times.'" Write these examples on the board to visually demonstrate how loops reduce repetition.  As a whole class, brainstorm everyday examples where repeated steps occur throughout the day (e.g. tying shoes, eating a meal, etc). Ask students to apply a number of times that each of these things might happen in a day. They may suggest "2" for tying shoes, "3" for eating a meal, and so on. Ask students why this might be useful in coding. Highlight that loops make code more efficient, easier to write, and less prone to errors.
40 min	<b>Swift Playgrounds</b>  Distribute iPads and have students open the "Learn to Code 1" playground. Direct students to the Loops section. If they've been working on specific devices in previous lessons, ensure they use the same device.  For this activity, refer to the Keynote slides: <ul style="list-style-type: none"><li>• Slides 12-15: Use these slides to explain the purpose and mechanics of loops, connecting it to the introductory activity.</li><li>• Slide 16: Play the video that demonstrates how to use loops in Swift Playgrounds and Guide them through the first puzzle in the Loops section.</li></ul> Encourage students to continue working through the loops puzzles independently after the video. Circulate the room to offer support and clarify questions. If students finish early, challenge them to explore the next set of puzzles or ask them to create their own loop-based task for a peer to solve.
5-10 min	<b>Reflection</b>  Explain that loops are common in coding because many tasks involve repeated steps. Using loops makes it easier to write and understand code while reducing the chance of mistakes from repeated manual steps. Celebrate that students progress made in writing efficient code!

## Connecting {design} and {code}

Loops handle repetition in coding, just as app wireframes may include repeated user tasks or actions. While learning Keynote, students can connect loops to designing recurring processes in their apps.



## Glossary

### {design}

#### The 5 Ws

The five predominant questions to ask when explaining a concept: “who, what, where, why, when” (and “how,” but “5Ws + 1H” just doesn’t sound as good!)

#### App design

The way in which students lay out the components of their app

#### App functionalities

The different things an app can or cannot do

#### Proposal

A preliminary written outline of a student’s plan for their app concept, design, and features; this assignment is often a major assessment as a core part of the planning process within the project

### {code}

#### For Loop

A type of loop that repeats a block of code a specific number of times, often used with a counter variable

#### Iteration

The repetition of a process or set of instructions

#### Loop

A programming construct that repeats a block of code multiple times

## Welcome to Workshop Week!

As a Career Education Council (CEC)-supported program, this week features the first of four different hands-on workshops led by Apple Distinguished Educators or Apple Professional Learning Providers.

### Overview

This session introduces students to the creative world of app design using Keynote, showcasing its powerful design features beyond traditional presentations. Students will learn to use shapes, typography, colour, and animation to create a working prototype of an app concept. Through hands-on exploration, they will view their app ideas through a visual design lens, developing skills to effectively communicate their vision.

### Goals for Understanding

- Understand how to use Keynote as a tool for app prototyping, focusing on shapes, typography, colour, and animation.
- Build foundational design thinking skills to prepare for future coding with Swift Playgrounds or Xcode.
- Refine the ability to pitch app ideas by mastering visual storytelling techniques.
- Empower students to create visually compelling, user-friendly apps that integrate both design and technical functionality.

### Workshop Logistical Prep

Here are some items you should prepare, schedule, and check prior to each of this virtual workshop:

- ☐ Confirm the date and time of your Keynote Fundamentals workshop with your CEC coordinator.
- ☐ Confirm the video-conferencing platform you will be hosting the virtual call on (i.e. Zoom, Teams, Meet, etc). Ensure you have the appropriate app downloaded, logged in, and you have the call link readily available on the date of the presentation.
- ☐ Plan to host the session in a location that has a dependable internet connection. If this location requires booking, do so in advance.
- ☐ Ensure the space has the ability to project your computer onto a screen with full audiovisual capabilities. Test the set-up in advance to ensure both visuals and sound will be suitably adjusted for when students are together in the space.
- ☐ Book a class set of iPads for each workshop well in advance. Ensure the iPads are fully charged in the day leading up to the workshop.
- ☐ Check that the latest version of Keynote is installed on the iPads you have booked.

### Required Materials

- Projector set-up & video conferencing app to connect with presenter (predetermined with CEC coordinator)
- iPads with Keynote app installed - one device per student or partner pair

### Lesson Resources

- Additional workshop details can be found in the [Develop the Future {ADE\\_Workshop\\_Guide}](#)

In this workshop, students may collaborate on one iPad with their project partner (if they have one), or independently if they would like to develop skills independently, then come back together later.

### Overview

This week is all about consolidating and catching up on all of the learning completed in the {code} stream so far. Teachers will have an opportunity to review content as required and students can be provided with time to work on and complete their Swift Playground Puzzles. When they are finished the puzzles, they should begin wireframing their app idea on Keynote.

### Goals for Understanding

By the end of the lesson, students should be able to:

- Review concepts learned so far in order to consolidate coding knowledge
- Work through and complete all Swift Playgrounds puzzles for commands, functions, and loops
- Begin wire-framing their app idea on Keynote, following the Keynote workshop

### Extension Activities

Students who have already completed all the Swift Playground Puzzles in the Commands, Functions, and Loops sections of the Learn to Code 1 Playground have a couple of options this week:

1. Begin wireframing their app idea on Keynote. Building the foundational elements of their app idea on Keynote is a critical step to the app development process. The {design} and {code} streams begin to come together as students consider layouts, colours, fonts, logos, and functionalities of their app idea.
2. If students are hungry for more coding challenges, they may complete any playgrounds from Learn to Code 1 they have yet to explore or they can dive into Learn to Code 2 and start with the “Types” module.

### Required Materials

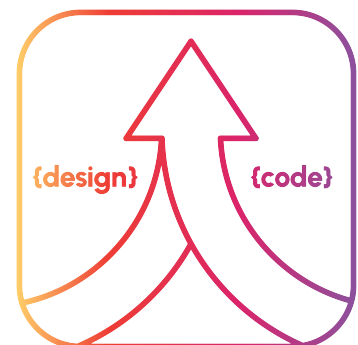
- iPads with the [Swift Playgrounds](#) app installed
- Students' Learn to Code 1 Playground file, with previous progress saved

### Lesson Resources

- [Everyone Can Code Puzzle Book](#) (pages 7-27)

## Connecting {design} and {code}

As students review content and work on completing their Swift Playground puzzles, they will also begin wireframing their app idea on Keynote. As they explore how to personalize user profiles or preferences in their app, they can apply similar design thinking in Keynote to create layouts and visual elements that reflect customization. This process allows students to integrate the design and code streams, considering aspects such as layouts, colours, fonts, logos, and functionalities for their app.



### Overview

Now that students have a grasp on how to use Keynote to wireframe an app, they can begin **prototyping** their own app design! Using their “5 Ws” app proposal and resources from the Keynote workshop as a guide, students can design their **app logo** and set up the initial slides that act as the backbone for their **app wireframe**.

### Goals for Understanding

By the end of the week, students should be able to:

- Create a rough mock-up of their app logo
- Develop at least 3-5 slides of their initial wireframe in Keynote, including **linking between slides**
- Begin to identify more **challenging features** that they do not yet know how to build in Keynote (l.e. drop-down menu, search function, collapsible filter, etc.)

### Required Materials

- iPad with Keynote app installed

### Lesson Resources

- Week 6 - Develop the Future Middle Years slides resource (Keynote - slides 2-10)

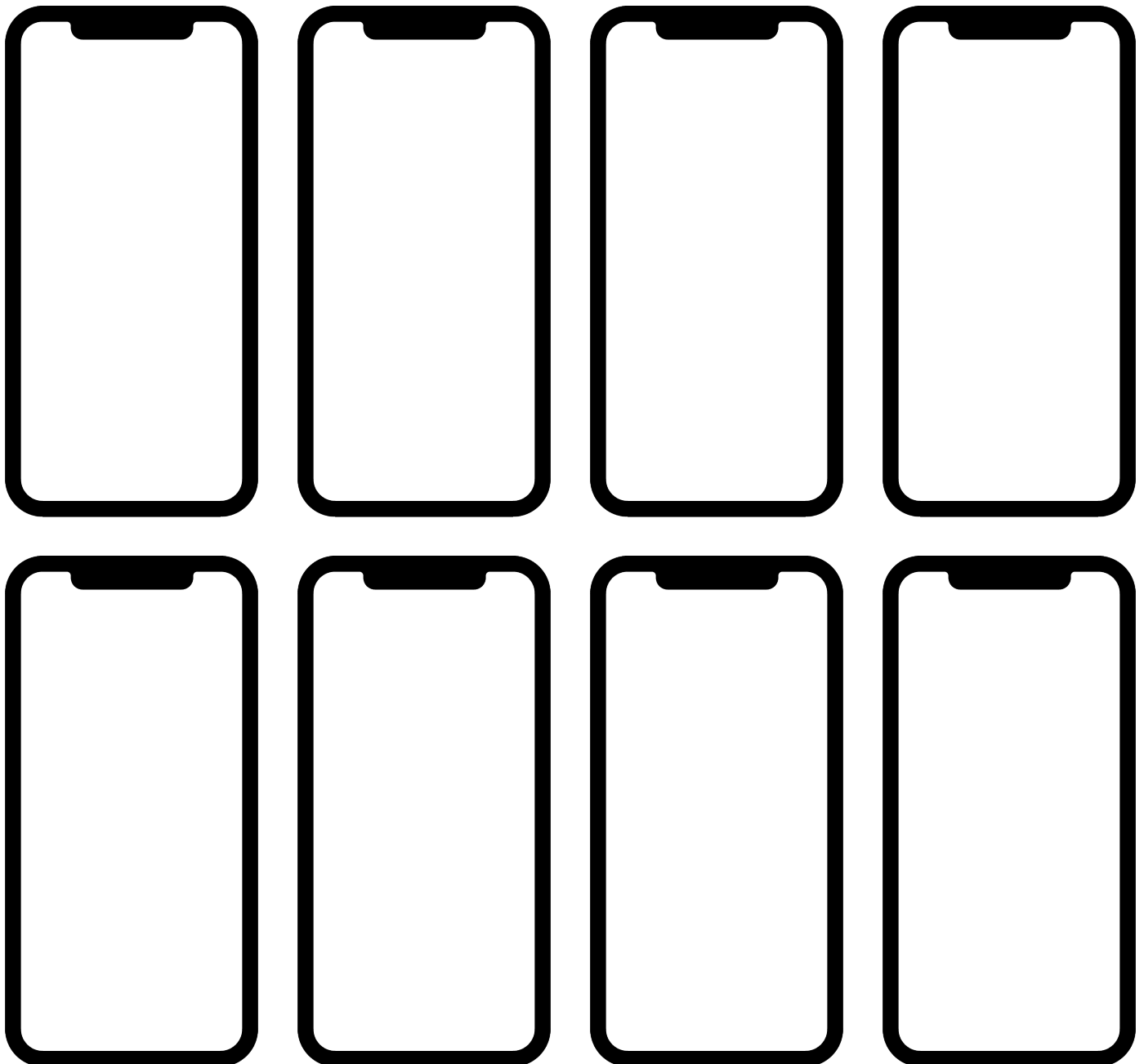
### Learning Activities

Time	Activity
5 min	<p><b>Design Thinking: The Prototyping Stage</b></p> <p>Begin the week’s work by sharing the Design Thinking Framework on the “Initial Wireframing” Keynote slides. Take a moment to have students reflect on the process that they have engaged in so far. They have spent time empathizing with potential users and defining a problem formally via their app proposal assignment – now it is time to prototype!</p> <p>The app wireframe that each student develops on Keynote will act as their prototype for their project. Take some time to discuss the evolving nature of a project of this scope - there will be multiple <b>iterations</b> of their work, and they should be mentally prepared that the first copy is not the final copy!</p>
10 min	<p><b>Creating an App Logo Mock-up</b></p> <p>Once students have iPads, it’s time to get to work! They can begin by opening a new Keynote file. Insert the phone shape, then ensure it is enlarged and locked on their screen. From there, they can select their brand colours and use shapes in order to design their app logo on Keynote. Don’t forget to add a “Get Started” button, too! You may want to complete an opening page in a guided fashion to do a quick assessment on student support needs.</p>
30 min+ <i>(extra work blocks can be offered this week as well)</i>	<p><b>Wireframing</b></p> <p>After students create their app logo, they can duplicate the slide in order to build out the opening pages of their <b>user interface (UI)</b>. Where should that next slide take the user? Allow students some time to imagine how their app might be set up. Different students will have different processes for this. Some will map out multiple slides with notes. Others will draw it out on paper. Others yet will create each slide one by one with the plan all in their head. There is no wrong way to move through this creative process! If you’d like to scaffold this out beforehand, that’s fine, too! You may use the optional “Wireframe Planning” handout from this week as a planning support.</p> <p><i>Note:</i> There may be some features students want to include but are unsure how to build or set up in their wireframe. Have them note their questions for the advanced Keynote workshop!</p>
15 min	<p><b>File Naming and File Back-up Recommendations</b></p> <p>Ensure students rename their Keynote file with their name(s) at the front of the file name. You may wish to back-up files by having students AirDrop their Keynote file to you each week, particularly if students access iPads that are shared across multiple classrooms. This prevents students losing their work in the event it gets deleted.</p>



Name: \_\_\_\_\_

**Instructions:** Wireframing is the process of mapping out an application or website. A good wireframe should demonstrate logical visual sequences, clearly showing the pathway where one page would lead to the next, and the next, and so on. Use this handout to sketch out your app wireframe. Include features you'd like to add such as buttons, images, videos, drop-down menus and more. You can use as many copies of this page as you need to as you complete your initial planning.



## Welcome to Workshop Week!

As a Career Education Council (CEC)-supported program, this week features the second of four different hands-on workshops led by Apple Distinguished Educators or Apple Professional Learning Providers.

### Session Overview

In this hands-on workshop, students will create a personal project that incorporates elements of design and code using the “About Me” guided playground in the Swift Playgrounds app. This activity encourages self-expression and creativity while reinforcing the coding skills that have been developed over the last 6 weeks.

### Goals for Understanding

- Apply learning about commands, sequences, and loops to design a project that showcases personal interests, hobbies, and aspirations
- Experiment with both code and design choices
- Observe how altering code directly impacts what appears on the screen as well as the user experience (UX)
- Sharing “About Me” creations with peers

### Workshop Logistical Prep

Here are some items you should prepare, schedule, and check prior to this virtual workshop:

- ☐ Confirm the date and time of your About Me workshop with your CEC coordinator.
- ☐ Confirm the video-conferencing platform you will be hosting the virtual call on (i.e. Zoom, Teams, Meet, etc). Ensure you have the appropriate app downloaded, logged in, and you have the call link readily available on the date of the presentation.
- ☐ Plan to host the session in a location that has a dependable internet connection. If this location requires booking, do so in advance.
- ☐ Ensure the space has the ability to project your computer onto a screen with full audiovisual capabilities. Test the set-up in advance to ensure both visuals and sound will be suitably adjusted for when students are together in the space.
- ☐ Book a class set of iPads for each workshop well in advance. Ensure the iPads are fully charged in the day leading up to the workshop.
- ☐ Check that the latest version of Swift Playgrounds is installed on the iPads you have booked, with the About Me playground downloaded inside of the app

### Required Materials

- Projector set-up & video conferencing app to connect with presenter (predetermined with CEC coordinator)
- iPads with Swift Playgrounds app installed - one device per student

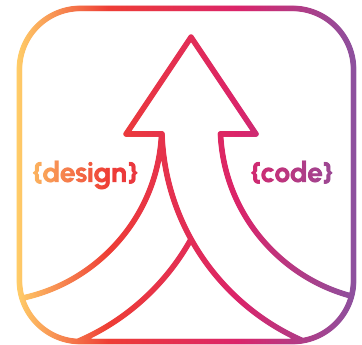
### Lesson Resources

- Additional workshop details can be found in the [Develop the Future {ADE\\_Workshop\\_Guide}](#)

In this workshop, it is best for students to work on their own iPad to independently to ensure a fulsome experience in understanding the convergence of design and coding principles.

## Connecting {design} and {code}

The All About Me workshop inspires students to use code and design principles that emphasize user interaction and personalization, encouraging them to think about how their app's interface can adapt to different user choices.



## Glossary

### {design}

#### App Logo

A symbol or icon that represents an app's overall identity, purpose, and brand

#### App Wireframe

The basic visual layout for a mobile app that demonstrates the structure, design, and functionality of the app's user interface

#### Challenging Features

Features that are not necessarily easy to visually recreate via a Keynote wireframe (I.e. search, drop-down menus, etc.)

#### Iterations

The cyclical process of refinement and improvement of a prototype based on feedback, evaluation, and reflection

#### Linking Between Slides

Connecting the visual components of an object on the user interface

#### Prototyping

Creating preliminary versions of a mobile app which can be used to test and refine the app's design, functionality, and usability prior to sharing with others

#### User Interface

The visual and interactive elements that allow a user to engage with an app's features and functionalities

### {code}

#### Creative Design

Using coding to express personal identity and stories through visuals.

#### Loops

Programming constructs that repeat code

#### Commands

Instructions that tell the computer to perform specific actions

#### Sequences

The order in which commands are executed in a program

#### Presentation Skills

The ability to effectively showcase and communicate work to others.

#### Self-Expression

Sharing personal identities and stories through coding.

### Overview

Students may be savvy with navigating apps but can they pick out the features that make an app work well? This lesson prompts students to find and identify various **app features**, known in the developing world as **UI elements** or **controls**, that make apps more user friendly. From buttons to drop-down menus to search filters, these tools make our lives easier, and they're critical for app designers to understand.

### Goals for Understanding

By the end of the week, students should be able to:

- Identify and define various app features within multiple apps
- Discuss how particular features may fit into their own **app concept** and design

### UI Element Examples

- Buttons
- Toggle switches
- Sliders
- Search bars
- Drop-down menus
- Search filters
- User history
- Interactive charts
- Chat interfaces
- Camera
- Insert photos/video
- Insert drawing
- ...and more!

### Learning Activities

Time	Activity
5 min	<b>What are App Features?</b> Open the lesson with a table talk discussion: what are app features and how many can you brainstorm with your group? Prompt them to think about apps they commonly use and what types of features they have. If students are struggling defining what app features are, show them some examples on Slide 5 and have them resume their discussion.
10 min	<b>Feature-Finding with the Weather App</b> Keep students in table groups or have them split off into partners with an iPad. Have them jump into the Weather app (or another simple app of your choice) and see if they can identify any features within the app and try to figure out why they were designed that way. Model how pressing the buttons (including tapping and holding!) and exploring different buttons, tools, and menus reveal different features of the app. Create a list on the board of the features you find and discuss their relevance to the app experience. Some examples are on Slide 5.
20 min	<b>Their Turn: Feature-Finding</b> Now it's the students' turn to do some feature-finding exploration on the iPad with other apps. Their task will be navigating these apps from a different lens than they're used to: what features can they find in the app that were an intentional design choice by the creators? You may wish to assign students specific apps to look into, or you might have them explore apps of their choice. Provide students with the Feature-finding hand-out so they can jot down app features as they explore.
5 min	<b>Share &amp; Discuss</b> Bring the class back together and have students share some of their findings in a whole-class discussion. Throughout the discussion, write on the board to start a master list of the different app features that the class found. If they're having trouble, steer them towards features like search bars, toggle switches, sliders, drop-down menus, filters, user history, media import tools, etc.  Finish the discussion by having students reflect individually or with their partner: Which app features are you curious about? Which would you like to integrate within your own app idea?

### Required Materials

- iPads for each partner pair or table group with access to various apps

### Lesson Resources

- Week 7 - Develop the Future Middle Years slides resource (Keynote - slides 2-10)

Name: \_\_\_\_\_

**Instructions:** This learning task is all about exploring, but from a different perspective! Take some time to look at common mobile apps from an **app design lens**. App features should make a user's experience more smooth. Consider these questions as you explore the apps your teacher assigned:

- What are the elements a user interacts with in this app? (I.e. buttons, drop-down menus, toggle switches, sliders, etc.)
- What did the designer intentionally include to make using the app easier or more enjoyable?

As you consider these questions, take time to write down any app features that you find, even if you don't intend to use them in your own app design.

App Name	Features Found (name the elements that make the user experience easy, interesting, or enjoyable)	
	<ul style="list-style-type: none"><li>•</li><li>•</li><li>•</li><li>•</li><li>•</li></ul>	<ul style="list-style-type: none"><li>•</li><li>•</li><li>•</li><li>•</li><li>•</li></ul>
	<ul style="list-style-type: none"><li>•</li><li>•</li><li>•</li><li>•</li><li>•</li></ul>	<ul style="list-style-type: none"><li>•</li><li>•</li><li>•</li><li>•</li><li>•</li></ul>
	<ul style="list-style-type: none"><li>•</li><li>•</li><li>•</li><li>•</li><li>•</li></ul>	<ul style="list-style-type: none"><li>•</li><li>•</li><li>•</li><li>•</li><li>•</li></ul>

### Overview

In this lesson, students will learn about **variables** in programming. They will understand how variables can store information and be used to control aspects of their code, applying this knowledge to create engaging patterns using **camelCase** for variable naming.

### Goals for Understanding

By the end of the lesson, students should be able to:

- Understand variables and how they store and manipulate data, and how to update variables
- Use variables to create dynamic code

### Required Materials

- iPads with the [Swift Playgrounds](#) app installed
- Learn to Code 2 Playground downloaded in the Swift Playgrounds app

### Lesson Resources

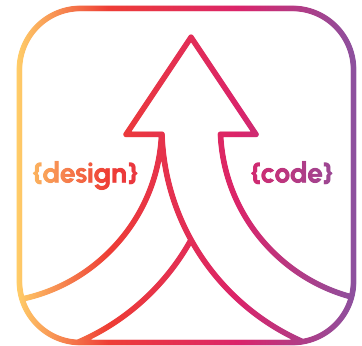
- [Everyone Can Code Puzzle Book](#) (pages 41-53)
- Week 7 - Develop the Future Middle Years slides resource (Keynote - slides 11-24)

### Learning Activities

Time	Activity						
10 min	<p>Imagine you're playing a basketball game and keeping score. Instead of writing it down every time, you keep track of it in your head. Let's say at the start of the game, your score is 0.</p> <p>This is your starting variable: score = 0</p> <p>Now, let's take some shots:</p> <table><tr><td>1. You take a shot and miss.</td><td>The score doesn't change: score = 0</td></tr><tr><td>2. You make a free throw (1 point).</td><td>Update the variable: score = 1</td></tr><tr><td>3. You make a two-pointer (2 points).</td><td>Update the variable again: score = 3</td></tr></table> <p>Each time something happens in the game, we check the score and update it. This is exactly how variables work in coding! Variables allow computers to <i>remember</i> things and update them as needed, just like we do when keeping score in a game. In coding, we use variables for everything—tracking scores, counting points, storing player names, and much more!</p> <p>Open the class by setting up a mock basketball activity, having a few students shooting a paper ball at a basket or garbage bin from different distances for each score. Another student can track the score, keeping a table of how many three-pointers, two-pointers, and free throws are made, and how the total score changes each time a basket is made.</p>	1. You take a shot and miss.	The score doesn't change: score = 0	2. You make a free throw (1 point).	Update the variable: score = 1	3. You make a two-pointer (2 points).	Update the variable again: score = 3
1. You take a shot and miss.	The score doesn't change: score = 0						
2. You make a free throw (1 point).	Update the variable: score = 1						
3. You make a two-pointer (2 points).	Update the variable again: score = 3						
35 min	<p><b>Swift Playgrounds</b></p> <p>Use Slide 15 to provide a concise overview of variables in Swift, connecting it to the earlier basketball example. Play the accompanying video on Slide 17, which introduces variables in coding and explains their role in making programs dynamic.</p> <p>Distribute iPads and have students open Swift Playgrounds. Direct them to the Learn to Code 2 module, specifically the section on Variables (refer to Slide 18 for navigation details).</p>						
5 min	<p><b>Reflection on Variables</b></p> <p>Bring students together for a brief discussion using Slide 21. Explain that variables are fundamental to coding because they allow programs to store, manipulate, and reuse data efficiently. Celebrate the students' progress in understanding how variables enable more powerful and adaptable programming.</p>						

## Connecting {design} and {code}

On an iPhone and in apps, variables help store and update information, making features work dynamically. Apps use variables to track progress (like steps in a fitness app), save preferences (such as dark mode), and adjust content in real time (like weather updates or shopping recommendations). Exploring these features helps reveal how variables control data and create a more interactive experience.



## Glossary

### {design}

#### App Concept

The core idea of how a mobile app addresses a specific need or problem

#### App Features

The specific functions, tools, or capabilities within an app that supports its overall purpose, which can include navigation, interactive UI components, or customization options; also known as controls or UI elements (see below)

#### Controls

How a developer might refer to a coded UI element, and what the user interacts with to navigate the app; interchangeable language with app features above

#### UI Elements

Visual elements that allow a user to navigate or use an app; interchangeable language with app features above

### {code}

#### Booleans

Data types that can hold true or false values

#### Camel Case

A naming convention where each word in a variable name is capitalized except for the first word (e.g., myVariableName)

#### Data Types

Categories that define the type of data a variable can hold, such as numbers or text

#### Double

A data type used to represent floating-point numbers, which can contain decimals

#### Integers

Whole numbers without decimal points

#### Strings

Sequences of characters used to represent text

#### Variables

Named storage locations in code that hold data values

## Welcome to Workshop Week!

As a Career Education Council (CEC)-supported program, this week features the third of four different hands-on workshops led by Apple Distinguished Educators or Apple Professional Learning Providers.

### Session Overview

This follow-up workshop to Keynote Fundamentals offers a deeper dive into how Keynote can create powerful mobile app wireframes and how it's actually a powerful tool to reinforce coding skills. By this point, students should have honed their foundational Keynote skills through initial wireframing, and are ready to build more complex features.

### Goals for Understanding

- Explore how to use Keynote's **builds, animations, transitions, and media options** to build more complex UI features into their app ideas
- Customize one or more complex app feature builds for their own wireframe
- Connect the idea of "feature-finding" to "**feature-building**"
- Critically analyze where, how, and why complex features could exist in their app's final wireframe

### Workshop Logistical Prep

Here are some items you should prepare, schedule, and check prior to each of this virtual workshop:

- ☐ Confirm the date and time of your Feature-building in Keynote workshop with your CEC coordinator.
- ☐ Confirm the video-conferencing platform you will be hosting the virtual call on (i.e. Zoom, Teams, Meet, etc). Ensure you have the appropriate app downloaded, logged in, and you have the call link readily available on the date of the presentation.
- ☐ Plan to host the session in a location that has a dependable internet connection. If this location requires booking, do so in advance.
- ☐ Ensure the space has the ability to project your computer onto a screen with full audiovisual capabilities. Test the set-up in advance to ensure both visuals and sound will be suitably adjusted for when students are together in the space.
- ☐ Book a class set of iPads for each workshop well in advance. Ensure the iPads are fully charged in the day leading up to the workshop.
- ☐ Check that the latest version of Keynote is installed on the iPads you have booked.

**Note:** As you enter the second half of the project timeline, it is recommended to provide 1-2 hours per week for dedicated app wireframing work blocks. This time is crucial for students to bring their ideas to life, give and receive feedback to one another, and have a working prototype for the upcoming midterm check-in (Week 11).

### Required Materials

- Projector set-up & video conferencing app to connect with presenter (predetermined with CEC coordinator)
- iPads with Keynote app installed - one device per student or partner pair

### Lesson Resources

- Additional workshop details can be found in the [Develop the Future {ADE\\_Workshop\\_Guide}](#)

In this workshop, students may collaborate on one iPad with their project partner (if they have one), or independently if they would like to develop skills independently, then come back together later.



### Overview

In this lesson, students will learn about conditional code and how it enables decision-making in programming. They will explore **if-else statements** and **logical operators** through engaging activities in Swift Playgrounds. This lesson helps students understand how to control the flow of their programs based on different conditions that present themselves as a program is used.

### Goals for Understanding

By the end of the lesson, students should be able to:

- Understand how **conditional statements** are used to make decisions in code.
- Explore **logical operators** like **AND**, **OR**, and **NOT**.
- Create interactive scenarios (**conditionals**)

### Required Materials

- iPads with the [Swift Playgrounds](#) app installed
- Learn to Code 1 Playground downloaded in the Swift Playgrounds app

### Lesson Resources

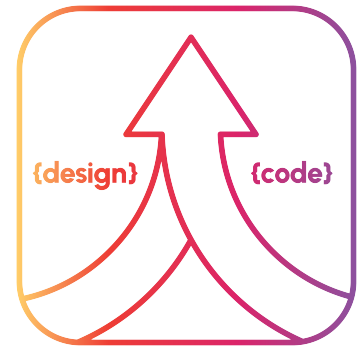
- [Everyone Can Code Puzzle Book](#) (pages 54-66)
- Week 8 - Develop the Future Middle Years slides resource (Keynote - slides 2-18)

### Learning Activities

Time	Activity
10 min	<p><b>Exploring Conditional Code with Relatable Analogies</b></p> <p>Start the lesson with a simple example on the board: “If it rains, then I’ll take an umbrella.” Discuss the alternative option: “If it’s not raining, I will take my sunglasses.”</p> <p>Pose a number of “IF” questions to students: What do you do if your phone battery is low? What do you wear if it’s snowing outside? Write student responses on the whiteboard, highlighting the “If ___ then ___” pattern. The key takeaway is that conditional logic is about making decisions based on different conditions — both in real life and in coding!</p> <p>Break students into small groups (3-4 per group). Each group will create a flowchart that represents a real-world decision based on a prompt from the slides.</p> <p>Example prompts for their flowcharts:</p> <ul style="list-style-type: none"><li>• <b>Basketball shot</b> (Slide 7): If you make the shot, add points; if you miss, the score stays the same.</li><li>• <b>Traffic light</b> (Slide 8): If the light is red, stop; otherwise, drive forward.</li></ul> <p>Walk around and guide groups by asking questions:</p> <ul style="list-style-type: none"><li>• What is the decision being made?</li><li>• What are the possible outcomes?</li><li>• Can you break it down into simple steps?</li></ul>
40 min	<p><b>Swift Playgrounds</b></p> <p>Use Slide 9 to introduce the syntax of if-else statements in coding, linking it to the earlier analogy.</p> <p>Play the accompanying video on Slide 10, which explains how to write and use if-else statements in Swift. Distribute iPads and guide students to open Swift Playgrounds, specifically the Learn to Code 1 module’s Conditionals section.</p> <p>Use Slide 12 and the accompanying video to guide students through the first puzzle in the Conditionals section, showing how conditional logic can dictate different outcomes in code.</p> <p>Encourage students to explore additional puzzles in the Conditionals section at their own pace. Pair students who need additional support with those who finish quickly.</p>
5 min	<p><b>Reflection on Conditionals</b></p> <p>Facilitate a group discussion using the guiding questions on Slide 15.</p> <p>Conclude by emphasizing how understanding conditionals improves logical thinking and problem-solving, which are essential skills in coding and everyday life.</p> <p>The final slides in the Keynote show that this week is the official ending to the Middle Years {code} journey in Swift Playgrounds. If students wish to continue challenging themselves with code, please refer to Slide 18.</p>

## Connecting {design} and {code}

As students identify and build key features for their app, they can consider how conditional code makes an app adapt to different user inputs. Similarly, they can design app features that change based on specific scenarios or actions.



## Glossary

### {design}

#### Animations (Keynote)

The option to add dynamic visual effects to objects on the screen in order to enhance user experience

#### Buils (Keynote)

Options for how and when an objects enters or exits the screen on a given slide

#### Feature-Building

Using Keynote's tools to develop more complex app features

#### Media Options (Keynote)

The ability to import or create media such as photos, videos, audio, or drawings that can be added to a Keynote slide

#### Transitions (Keynote)

The visual effects that occur as one slide moves to the next

### {code}

#### AND

A logical operator that returns true only if both conditions are true

#### Conditional Statements

Statements that execute code based on certain conditions

#### Flowchart

A visual representation of a process or algorithm, using shapes and arrows to illustrate the sequence of steps and decisions involved in a task or program.

#### If-Else Statements

A conditional statement that allows the execution of different code blocks based on whether a specified condition is true or false; the "if" block runs if the condition is true, while the "else" block runs if the condition is false

#### Logical Operators

Symbols used to connect two or more conditions in logic (e.g., AND, OR, NOT)

#### NOT

A logical operator that negates a condition, returning true if the condition is false

#### OR

A logical operator that returns true if at least one of the conditions is true