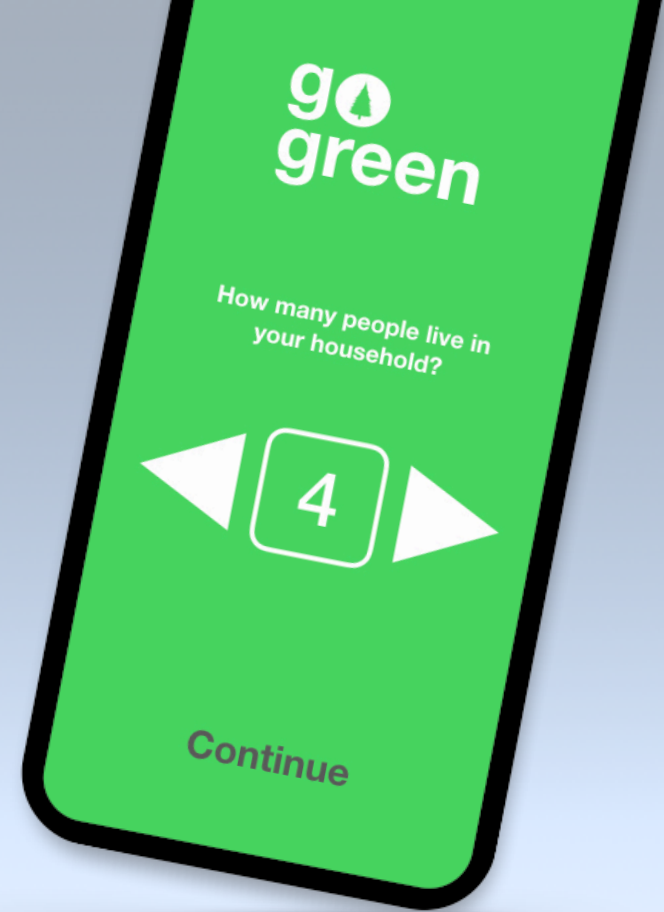




Swift Coding Club



# Xcode Kit

# Welcome to the Swift Coding Club!

Learning to code teaches you how to solve problems and work together in creative ways. And it helps you build apps that bring your ideas to life.

Swift Coding Clubs are a fun way to learn to code and design apps. Activities built around Swift, Apple's coding language, help you collaborate as you learn to code, prototype apps and think about how code can make a difference in the world around you.

You don't have to be a teacher or a coding expert to run a Swift Coding Club. The materials are self-paced, so you can even learn alongside your club members. And you can all celebrate your club's ideas and designs with an app showcase event for your community.

This kit is arranged in three sections:



## Get Started

Everything you need to launch a Swift Coding Club.



## Learn & Design

Tips and activities for designing club sessions.



## Celebrate

Helpful resources to plan and host an app showcase in your community.

## Swift Coding Clubs

### Block-Based Coding | Ages 8–11

Learn coding basics using visual apps on iPad.



### Swift Playgrounds | Ages 11+

Use Swift code to learn coding fundamentals with Swift Playgrounds on iPad.



### Xcode | Ages 14+

Learn to develop apps in Xcode on Mac.

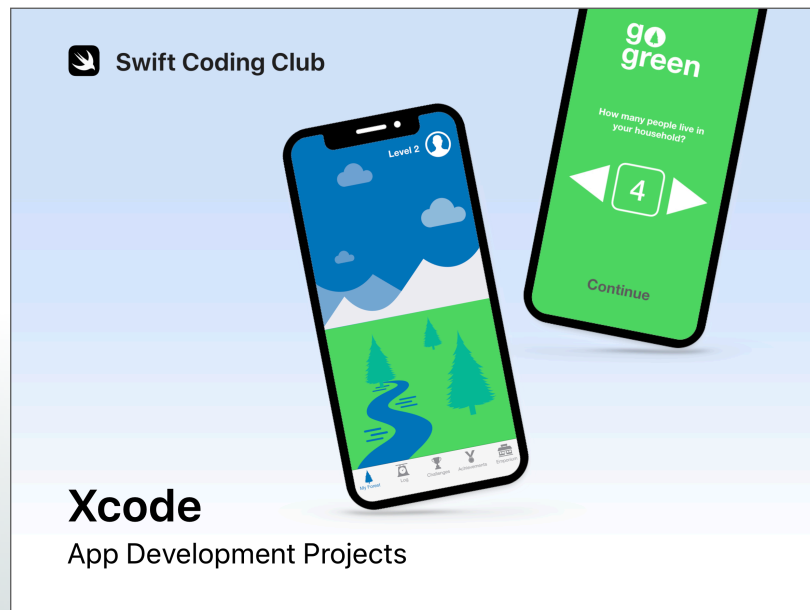


# Get Started



## 1. Download club materials.

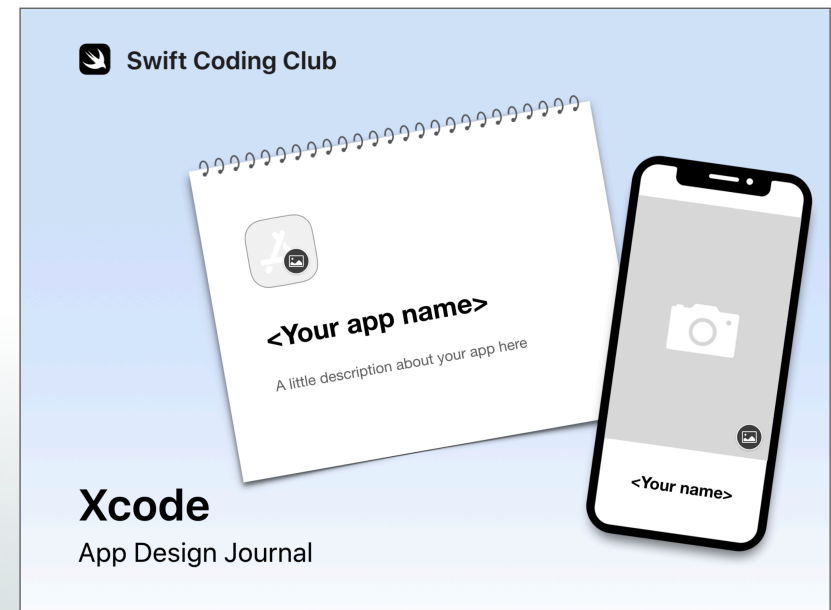
Use AirDrop to share these two guides with club members in your first club meeting. They're also included as part of this document.



### App Development Projects

Guided app development projects introduce the same tools, techniques and concepts that professionals use. Learn programming concepts as you write in Xcode playgrounds on Mac.

[Download Xcode App Development Projects >](#)



### App Design Journal

Explore the app design process with this Keynote journal. Brainstorm, plan, prototype and evaluate your club's app ideas.

[Download Xcode App Design Journal >](#)



## 2. Check your tech.

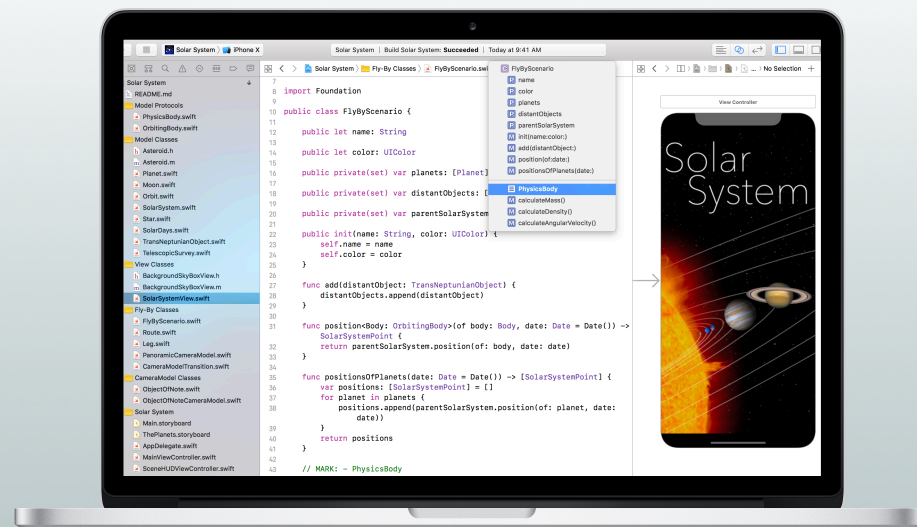
Before your first meeting, make sure you have:

- **Mac.** Club members will need Mac computers running macOS Mojave or High Sierra. It's recommended that they each have their own, but they can also share and code together.
- **Xcode.** This free Mac app from Apple is used to build every other Mac app and every iOS app too. It has all the tools for creating an amazing app experience. Xcode 9 is compatible with Swift 4, and Xcode 10 is compatible with Swift 4.2.
- **Intro to App Development with Swift.** This free resource from Apple guides beginners through eight coding projects.
- **Keynote.** You'll use the Keynote app on iPad for your app prototypes.
- **Swift Coding Club materials.**

## 3. Make a plan.

Here are some things to consider:

- Who are your club members? What are their interests? Do they have experience with coding or are they brand new to it?
- How often will your club meet? If you're planning to meet over the summer holidays, how many hours of coding activities will you have?
- What technology is available for the club?
- What are the goals of your club?







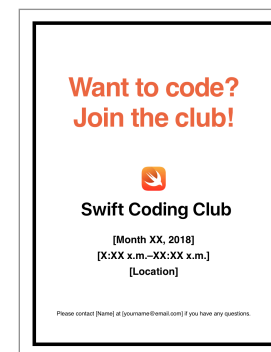
## 4. Spread the word.

Let people know about your Swift Coding Club. Here are some ideas and resources to attract new members to your club:

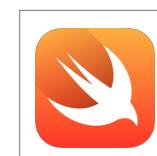
- **Announce your club.** Use email, social media, the web, flyers or word of mouth to let your community know about your club.
- **Host an informational meeting.** Ask potential club members about their interests and what types of apps they'd want to create. Talk about ideas for holding an app design showcase and how members can get involved. You can also share a short video about the club online.

These items can help you promote and personalise your Swift Coding Club:

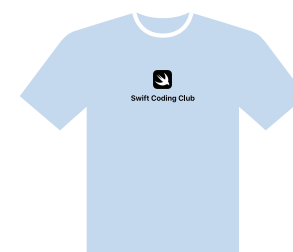
- **Posters.** [Download this free template](#), then personalise it to create your own poster. Print and display it, or make a digital poster to share online. Make sure you include details for when and where the club will meet and how to join.
- **Stickers and T-shirts.** Use these [Swift Coding Club stickers](#) to help promote your club. T-shirts are a great way to recognise members who participate in app showcase events. Download the [Swift Coding Club T-shirt template](#) to make shirts for your members.



Swift Coding Club poster



Swift Coding Club sticker

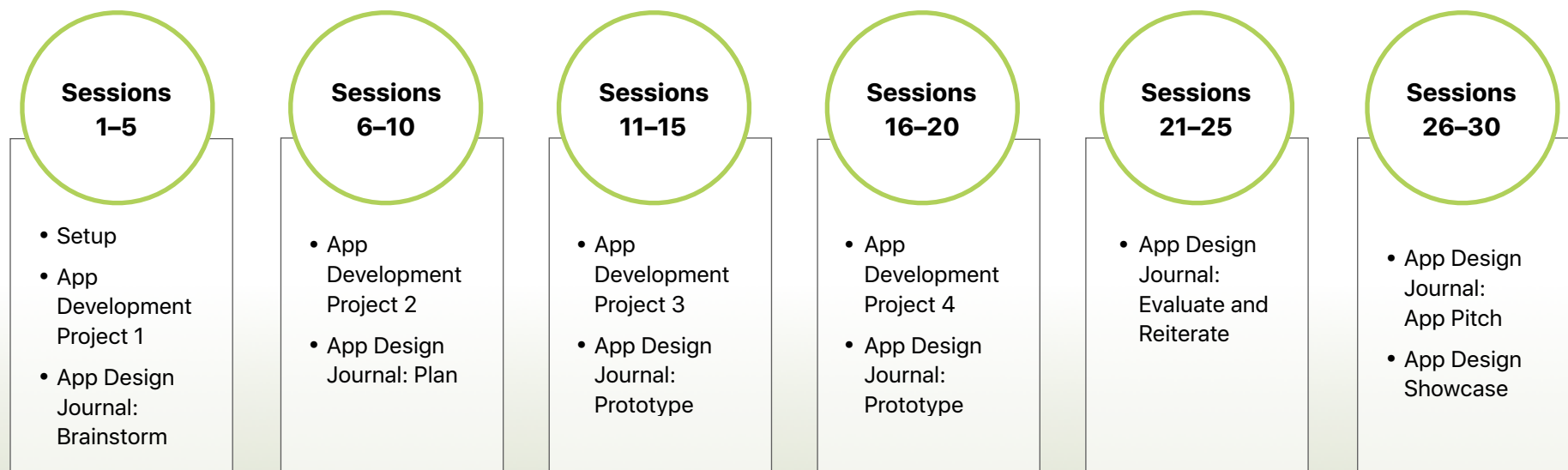


Swift Coding Club T-shirt



# Learn & Design

The club materials are designed for you to interweave coding and app design projects. You can also add sessions that support your members' interests. Below is a sample schedule for 30 one-hour club sessions.



Consider adding sessions to expand on app design and coding projects, like exploring augmented reality and virtual reality, and hearing from local app designers. To prompt app design brainstorming, you might even want to add guest speakers or excursions.

## Tips for Club Leaders



**Build a leadership team.** Having a group of members who help lead the club can make it much easier and more fun. Which club members have leadership potential? Think about adding club officers for events, coding, app design and more.

**Learn together.** Club leaders don't have to know everything. Help your members work on their own research and problem-solving skills, and encourage them to help others.

**Show off.** An app showcase event is a great way to promote your club, app ideas, and coding skills to friends, family members, teachers, and the community. It might even help you recruit more members. See page 11 to get tips for holding your own app showcase.

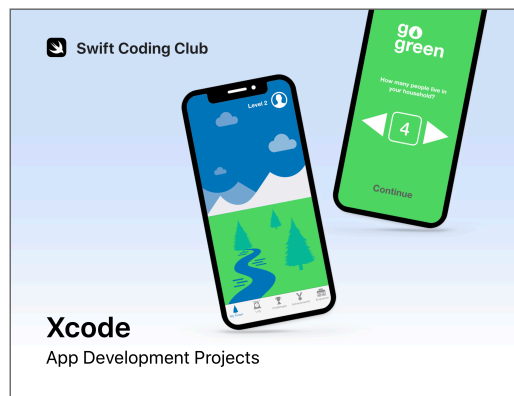


**Share ideas.** Some members will be interested in making games. Others might want to create apps to help people, learn Swift or control robots. Think about ways for members to work together on projects they care about.

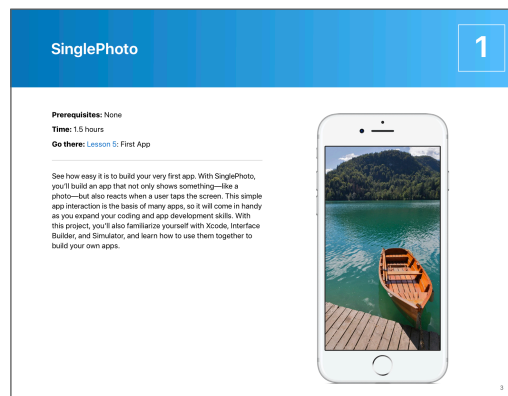
**Mix it up.** Sometimes members who are more advanced can leave others behind. See if those members can partner up with beginners for paired programming. Teaching someone else is a great way to learn!



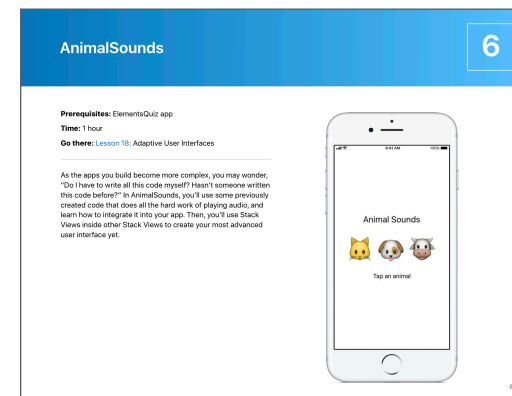
## Xcode App Development Projects




Eight projects guide a beginner through the basics of coding concepts and app development. Each project builds on the last and results in a working app, focusing on different aspects of apps so that by the end, students have the skills to build their very own apps.



Members aren't expected to complete all eight projects. The goal is to introduce them to different aspects of apps and inspire them to design their own apps.



Members will need the Intro to App Development with Swift course to do the projects. The prerequisites are a must!




Everyone Can Code

Intro to App Development with Swift

Xcode

Download the Intro to App Development with Swift >



Everyone Can Code

Intro to App Development with Swift Teacher Guide

Xcode

Need more information or want to go deeper?

Download the Intro to App Development with Swift Teacher Guide >

## Tips for Learning with Xcode



**Check the console to debug.** The console will sometimes have helpful information about what went wrong. The red highlight shows the line in the code where the error occurred.

**Stop and think.** Bugs are inevitable. Stop and think about the problem. What are its symptoms? Was it working fine until [X]?

**There's no one way to write code.** Members should review each other's code, provide feedback, and help each other debug.

**Set up a help desk.** Maintain a space where club experts can provide support to their peers.



**Explore Xcode preferences.** Set your text editing and other preferences by choosing Xcode > Preferences from the menu bar. You can add developer accounts, customise navigation or fonts, choose certain behaviours when events occur, and more.

### Master keyboard shortcuts.

Build the project: ⌘ B

Build and run the project: ⌘ R

Toggle comments on selected rows of code: ⌘ /

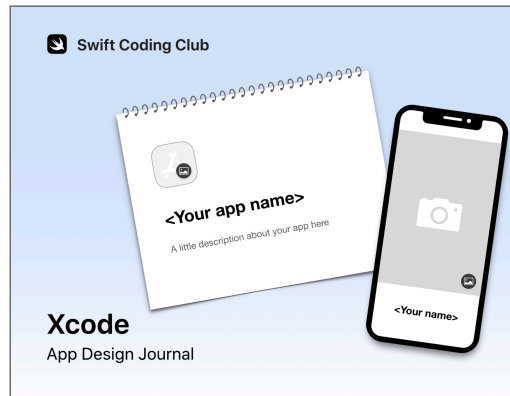
Shift the selected code left: ⌘ [

Shift the selected code right: ⌘ ]

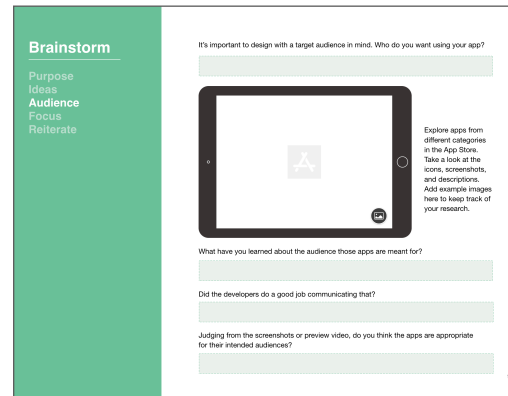
**Take it further.** Advanced members can work through the App Development with Swift course and focus more on coding aspects of their app prototypes in Xcode.



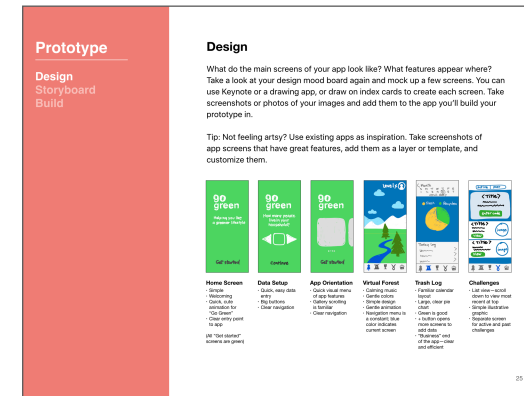
# App Design Journal



Coders use this Keynote journal to learn about app features and design an app to solve a community problem.

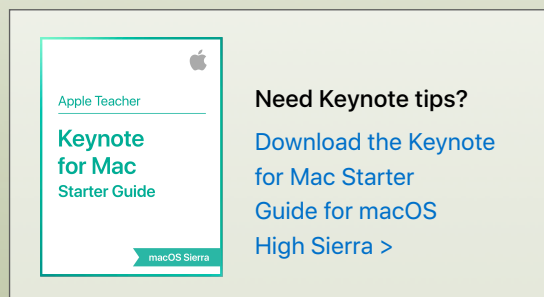


Club members work in small teams to brainstorm and plan the app solution, then create a working prototype of the app in Keynote.



The journal walks coders through the process of evaluating their designs and iterating on their prototypes — just like professional app designers.

Members create a three-minute app pitch presentation or video, and celebrate their work in an app design showcase.



# Celebrate



## App design showcase

The app design process and the showcase are powerful opportunities to involve the broader community and explore the potential of apps for solving contemporary problems. The showcase is also the perfect way to show off the talents of your club members!

**1. Plan the big event.** Set a date for the showcase and invite students, teachers, parents and community members to attend.

Allow time for each team to present their app pitch and to hold a short Q&A session. If you have a large group, you can split the club into two rounds where members can watch each other's pitches.

Consider finishing the event with a fun slideshow of photos taken throughout club sessions.

**2. Design awards.** Friendly competition can be a great motivator. Inspire club members by offering awards that recognise specific strengths in app design. Consider awards for:

- Best Engineering
- Best Innovation
- Best Design
- Best Pitch

You could also encourage audience participation with a People's Choice award.



You can download and modify this [certificate](#) for different awards.



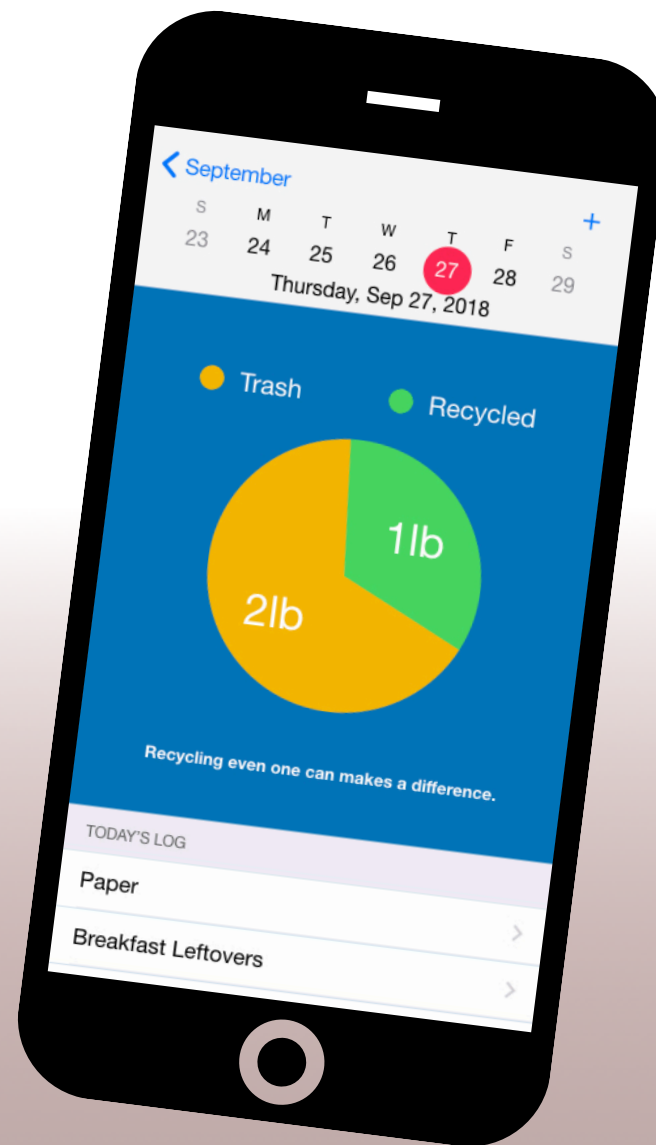


**3. Recruit judges and mentors.** Judges and mentors can be teachers or staff members, students with expertise in coding, experts from the developer or design industry, members of the school board, local community leaders or individuals who would benefit from the app idea.

Judges don't have to wait until the showcase to meet the club. Consider inviting them as guest speakers to share their expertise when your club members are in the brainstorming or planning phase of their app design.

**4. Pick a winner.** Judges can use the rubric on the next page to help them evaluate the app pitches and provide feedback. You could also share the rubric with coders before the showcase as part of the evaluation phase of the app design process.

**5. Share and inspire.** You may want to record the showcase presentations. Share them with the broader community and create a highlight reel to inspire future club members.



# Evaluation Rubric

[Download >](#)

	Novice (1 point)	Intermediate (2 points)	Proficient (3 points)	Mastered (4 points)	Points
<b>Pitch Content</b>	The pitch shares key information about the app, such as its purpose and target audience.	The pitch clearly explains how the app was designed and how it improves on existing apps that have a similar purpose and audience.	The pitch explains how the app was designed to meet a market demand and how its solution is unique.	The pitch provides evidence of market demand and explains how the app design process ensures app success with key stakeholders.	
<b>Pitch Delivery</b>	The team explains key points during its pitch.	The team delivers its pitch with confidence and enthusiasm.	The team delivers an engaging pitch using a range of audience engagement techniques.	The pitch is well articulated, creative, memorable and fluid across the team.	
<b>User Interface</b>	The UI design complements the purpose of the app and has thematically consistent screens.	The UI design applies familiar iOS interface elements, icons and text styles to achieve clarity and function. The prototype has interactive elements that demonstrate the app's behaviour.	The UI is elegant, concise and pleasing, with attention given to colour, layout and readability. It gives feedback to the user about their progress through the app, or the options available to them when they perform an action.	The app design defers to its content as the most important element. The UI design empowers the user to directly interact with and manipulate its content. It uses animation to provide additional feedback for interactions.	
<b>User Experience</b>	The prototype expresses a clear intent for the app and how users can interact with it to accomplish a goal.	The prototype uses consistent and standard navigation techniques to provide the user with a clear and intuitive path through its content.	The prototype enables users to view and interact with its content differently depending on their needs. The prototype addresses accessibility and includes features to protect user privacy or online safety.	The prototype innovates on best practices of similar apps, and caters to the needs of both new and experienced users. It exhibits a style and personality that set it apart from its peers.	
<b>Coding Concepts</b>	The team describes how their app design would relate to its code, such as the kind of data it stores or how it reacts to different user inputs.	The team explains how basic coding concepts like data types, conditional logic and touch events relate to the design of its app.	The team describes specific coding tasks that would be necessary to implement the app, as well as how that code powers its screens and/or interactions.	The team explains the algorithms at the heart of the app, and describes them conversationally or in pseudocode. They describe the app's functional parts and its data, and how they're structured and interrelated.	
<b>Code Implementation (Optional)</b> <i>Apply this metric if the team has extended their app prototype into Xcode and you're comfortable evaluating Swift and iOS code.</i>	Code runs with specific examples. It's basic with no abstraction.	Code runs without error in all cases. It's basic and there's some evidence of abstraction.	Code is organised with clear Swift naming conventions and there's high evidence of abstraction. It follows iOS API guidelines and UI conventions.	Code is well documented with comments. There's effective use of applicable Swift language and library features. It employs standard organisation principles, such as Model-View-Controller.	
<b>Comments</b>				<b>Total score</b>	



**Swift Coding Club**

Xcode

# Certificate of Achievement

Awarded to

For

---

Signature

---

Date

# Take It Further

Swift Coding Club is just the beginning of your coding journey. The Everyone Can Code curriculum provides fun, supportive resources to take coders from learning the basics on iPad to building real apps on Mac. App Development with Swift Certification is even available for students who have completed App Development with Swift.

And you don't have to stop at club activities. Comprehensive Teacher Guides also enable teachers to bring coding into the classroom, with step-by-step, curriculum-aligned lessons for students from kindergarten to university.

[See all the Everyone Can Code resources >](#)

