First App

Part 1 - New Project
Now that you’re getting more comfortable with playgrounds, you might be wondering how to build an app you can use on your iOS device, or even your Apple Watch. A lot of moving parts need to work together to make an app run, and Xcode is the best tool for putting them all together.

In this three-part lesson, you’ll build SinglePhoto—a simple iOS app that displays a single photo. In the first exercise, you’ll create an app project from scratch. Then you’ll use Xcode to explore your project and learn to navigate your coding environment.

You can customize every part of your app—from its icon on the Home screen to the way its buttons behave. There are panels and controls in Xcode that display the many options available to you. You’ll practice using the Xcode Interface Builder to continue customizing your first app.

In the final step, you’ll add an image to your project and edit the user interface. You’ll also get an introduction to Interface Builder—a powerful component of Xcode where you create the user interface of your app. At the end of the second exercise, your app will look like this, but it will display a photo of your own choosing.
Everyone Can Code

Technology has a language. It’s called code. And we believe coding is an essential skill. 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. We think everyone should have the opportunity to create something that can change the world. So we’ve designed a program with tools and resources that let anyone learn, write, and teach it.
# Apple Coding Curriculum

Apple’s coding curriculum includes a range of resources that take students all the way from no coding experience to building their first apps. The table below provides an overview of all the free teaching and learning resources available.

<table>
<thead>
<tr>
<th>Curriculum</th>
<th>Device</th>
<th>Audience</th>
<th>Prerequisites</th>
<th>Overview</th>
<th>Learning materials</th>
<th>Support resources</th>
<th>Number of lesson hours included</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kindergarten through grade 2</td>
<td>None</td>
<td>Begin to think like coders with hands-on explorations of coding concepts using visual-based apps.</td>
<td>codeSpark Academy app lessons, Tynker Space Cadet course</td>
<td>Get Started with Code 1: Teacher Guide</td>
<td>30 hours, including Teacher Guide and app lessons</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grades 3 through 5</td>
<td>None</td>
<td>Explore fundamental coding concepts and practice thinking like coders using visual-based apps.</td>
<td>Tynker Dragon Spells course</td>
<td>Get Started with Code 2: Teacher Guide</td>
<td>36 hours, including Teacher Guide and app lessons</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle school and up</td>
<td>None</td>
<td>Learn fundamental coding concepts using real Swift code.</td>
<td>Swift Playgrounds app, Learn to Code 1 &amp; 2 lessons, iTunes U course</td>
<td>Learn to Code 1 &amp; 2: Teacher Guide, Apple Teacher Learning Center Swift Playgrounds badges</td>
<td>Up to 85 hours, including Teacher Guide and Learn to Code 1 &amp; 2 lessons</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle school and up</td>
<td>Learn to Code 1 &amp; 2</td>
<td>Expand coding skills and start thinking more like an app developer.</td>
<td>Swift Playgrounds app, Learn to Code 3 lessons</td>
<td>Learn to Code 3: Teacher Guide</td>
<td>Up to 45 hours, including Teacher Guide and Learn to Code 3 lessons</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High school</td>
<td>None</td>
<td>Students get practical experience with the tools and techniques needed to build basic iOS apps while helping to prepare them for the AP® Computer Science Principles exam.</td>
<td>AP® Computer Science Principles with Swift book and project files</td>
<td>AP® Computer Science Principles with Swift: Teacher Guide</td>
<td>180 hours</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High school and college</td>
<td>None</td>
<td>Get practical experience with the tools, techniques, and concepts needed to build a basic iOS app from scratch.</td>
<td>Intro to App Development with Swift book and project files</td>
<td>Intro to App Development with Swift: Teacher Guide</td>
<td>90 hours</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High school and college</td>
<td>None</td>
<td>Build a foundation in Swift, UIKit, and networking through hands-on labs and guided projects. Students can build an app of their own design by the end of the course.</td>
<td>App Development with Swift book and project files</td>
<td>App Development with Swift: Teacher Guide</td>
<td>180 hours</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Welcome to AP® Computer Science Principles with Swift. This course is designed to help students build a solid foundation in programming using Swift. It gives students practical experience with the tools and techniques they’ll need to build basic iOS apps from scratch while helping prepare them for the AP® Computer Science Principles exam. Students will improve their problem-solving skills and develop their own ways of breaking down problems into manageable pieces. And as they gain more practice as programmers, they’ll get better at designing clever and efficient solutions to coding challenges.

Using Xcode—the same tool that professional developers use to create the apps on the App Store—students will gain experience in designing, planning, and testing different algorithms to get the logic of their programming to do what they want it to. They’ll also work in Xcode playgrounds, which serve as interactive sketchbooks for testing out Swift code and instantly seeing the results. And as students debug, test, and improve the apps they create, they’ll practice the important skills of collaboration, communication, and analysis. Finally, they’ll learn the app design cycle, a powerful model for designing, testing, and iterating on app ideas.
Course Materials

AP® Computer Science Principles with Swift
AP® Computer Science Principles with Swift course materials have been approved by the College Board for the 2019–2020 school year. Apple is one of many curriculum providers of AP® Computer Science Principles endorsed by the College Board.

Download the following AP® Computer Science Principles with Swift course materials, available for free on Apple Books:

AP® Computer Science Principles with Swift
AP® Computer Science Principles with Swift: Teacher Guide

Course syllabus, curriculum units, and mapping of units to the AP® Computer Science Principles curricular framework are available in the Teacher Guide.

AP® Computer Science Principles with Swift Professional Learning Camp
Apple will hold an AP® Computer Science Principles with Swift Professional Learning Camp for teachers confirmed to teach AP® Computer Science Principles with Swift in the 2019–2020 school year. The camp will be July 21-26, 2019.

Educators can apply to participate in this five-day, face-to-face professional learning camp. The camp includes hands-on activities, immersive experiences, and independent projects for teachers intending to teach AP® Computer Science Principles. The five days of training, teaching materials, meals during training, and online support throughout the 2019-2020 school year will be provided at no cost. Additional travel costs are not covered.

If you’re interested in attending the camp, please apply online. Applicants must confirm plans to teach AP® Computer Science Principles with Swift for the upcoming 2019–2020 school year with a recommendation by a school administrator. This event is limited to 100 participants. Register now >

College Board Curricular Framework
College Board curriculum framework
College Board course description
College Board exam description
Assessment overview and performance tasks for students
Explore task rubric and scoring guidelines
Create task rubric and scoring guidelines
College Board recruitment strategies for underrepresented students*
College Board AP® Teacher Community (login required)
Professional development standards

The College Board offers both face-to-face and online professional development workshops, available for new and experienced AP® teachers as well as AP® coordinators and administrators. Led by experienced instructors, subject-specific AP® workshops provide educators with the support and training needed to develop effective strategies for teaching an AP® course.

*The College Board states: "All students should have equitable access to this new course. Care should be taken to ensure that students taking the course are demographically representative of the school's population before confirming their enrollment. Therefore, we encourage you to create policies that promote and enable diversity in the course and to not create barriers that would discourage underrepresented groups from participating."
AP® Computer Science Principles Curricular Framework

About the Course

The AP® Computer Science Principles course is designed to be the equivalent of a first-semester introduction to computer science course at the college level. It focuses on seven Big Ideas that encompass foundational concepts of computer science.

Big Ideas
1. Creativity
2. Abstraction
3. Data and Information
4. Algorithms
5. Programming
6. The Internet
7. Global Impact

Each Big Idea is divided into a series of Enduring Understandings that students should understand about that Big Idea. These are broken up into specific Learning Objectives, which are linked to one or more Essential Knowledge statements. Students must know the Essential Knowledge statements to demonstrate an understanding of a particular Learning Objective. In general, each Learning Objective covers about one class period's worth of material.

Computational Thinking Practices

In addition, the AP® Computer Science Principles course contains six Computational Thinking Practices, which describe how students will explore the different Learning Objectives. Students can think of the Learning Objectives as the what and the Computational Thinking Practices as the how.

P1: Connecting Computing. Students will engage in this practice primarily in the Topic in Focus sections on privacy and the Internet, as well as in the Explore Task Practice sections that focus on the global implications of technology on society.

P2: Creating Computational Artifacts. Students will create computational artifacts primarily as a means of assessment for the How the Internet Works section of the course, as well as for the Create Task Practice sections in which they use Xcode to create interactive prototypes.

P3: Abstracting. Students will engage in this practice throughout the course, but Unit 2: Functions and Abstraction particularly focuses on the development of functions as a strategy to manage the complexity of code.

P4: Analyzing Problems and Artifacts. Throughout the curriculum, students will troubleshoot and debug sections of their code that don’t work. They’ll also be able to use Xcode playgrounds to see the effects of their changes in real time, which will help them analyze and fix problems.

P5: Communicating. A number of activities ask students to discuss questions and topics with partners. Students will also test their prototypes by observing other people navigating their apps, then interviewing the testers to ask them about their observations.

P6: Collaborating. The curriculum offers many opportunities to practice pair programming. And the Apply and Extend sections of the Teacher Guide often include additional activities that require communication and collaboration.
The AP® Computer Science Principles course asks students to show that they understand the fundamental computer science concepts. It also asks them to apply the computational thinking practices defined for the course to prepare them for the AP® Computer Science Principles exam. The course doesn’t require them to learn a particular language.

The AP® Computer Science Principles with Swift curriculum from Apple focuses on teaching students how to design and build apps using Xcode and the Swift programming language. The curriculum also goes in depth on cybersecurity, data privacy issues, and how the Internet works. Students will engage in research assignments and create computational artifacts using a range of Apple creativity tools, culminating in the through-course assessment.

The AP® Computer Science Principles End-of-Course Exam
The AP® Exam is given onsite, typically in early May. It consists of 74 multiple-choice questions that test students’ knowledge of the Learning Objectives for the course. The AP® Exam is worth 60 percent of the overall AP® score.

Through-Course Performance Tasks
The through-course performance tasks for this course require students to explore the impacts of computing and create computational artifacts both digitally and through programming. As with the AP® Exam, the performance tasks are designed to gather evidence of student learning associated with the learning objectives. Performance tasks assess student achievement in broader ways than time permits on the exam. Learning objectives are more effectively measured through a real-world performance task.

How to use this Teacher Guide
This teacher guide will help you bring AP® Computer Science Principles with Swift into the classroom. You can use it in a standalone class or as part of any app development or computer science program. The course will align to various national and international curriculum standards for computer science. (See the Curriculum Correlations chapter for more information and current alignments.)

Throughout the course, this guide will suggest how much time you’ll want to allocate for each activity. If you teach a less-structured class, you might choose to divide up a lesson into several sections. All in all, you can expect the course to take a semester to complete.

For each lesson in the course, this guide supports your instruction with concepts, information, and ideas.

The AP® Computer Science Principles with Swift: Teacher Guide includes additional activities, discussion questions, and activities aligned to the College Board curricular and resource requirements for the AP® Computer Science Principles course.
Course Outline

AP® Computer Science Principles with Swift

AP® Computer Science Principles with Swift is designed to introduce beginning programmers to all aspects of app development while helping prepare them for the AP® Computer Science Principles exam. Students will explore how apps affect our society, economy, and culture. They’ll also build a solid foundation in programming fundamentals using Swift as the language. Throughout this course, students will get practical experience with the tools, techniques, and concepts needed to build a basic iOS app. Lessons take students through the process of designing an app, including brainstorming, planning, prototyping, and evaluating apps of their own. And they’ll learn user interface design principles, which are essential to programming and making great apps.

Unit 1: Introduction to Coding (4 weeks)
1.1 Topic in Focus: How Computers Work
1.2 Explore Task Practice 1
1.3 Create Task Practice 1
1.4 Playground Basics
1.5 Naming and Identifiers
1.6 Strings
1.7 Hello, world!
1.8 First App

Unit 2: Functions and Abstraction (6 weeks)
2.1 Topic in Focus: How the Internet Works
2.2 Explore Task Practice 2
2.3 Create Task Practice 2
2.4 Functions
2.5 BoogieBot
2.6 Constants and Variables
2.7 Types
2.8 Parameters and Results

Unit 3: Algorithms (6 weeks)
3.1 Topic in Focus: Privacy and Cybersecurity
3.2 Making Decisions
3.3 Instances, Methods, and Properties
3.4 QuestionBot
3.5 Arrays and Loops
3.6 Defining Structures

Unit 4: Building Apps (10 weeks)
4.1 Topic in Focus: Cryptography
4.2 Explore Task (8 classroom hours)
4.3 ChatBot
4.4 Actions and Outlets
4.5 Adaptive User Interfaces
4.6 Enumerations and Switch
4.7 Final Project
4.8 Create Task (12 classroom hours)
Key Features

**Playgrounds.** Students learn programming concepts as they write code in playgrounds—an interactive coding environment that lets them experiment with code and see the results immediately.

**Sample projects.** Using the included project files, students can try out certain parts of code without having to build an entire app from the beginning.

**Step-by-step instructions.** Detailed instructions with images and videos guide students through all the steps of building an app in Xcode.

**Study tools.** Students can check their understanding and apply what they’ve learned with review questions, key vocabulary, links to documentation, and more.
Additional Information

Download the Swift Playgrounds resources
- Learn to Code 1 & 2: Teacher Guide
- Learn to Code 3: Teacher Guide
- Swift Playgrounds app

Download the App Development with Swift guides
- Intro to App Development with Swift
- Intro to App Development with Swift: Teacher Guide
- App Development with Swift
- App Development with Swift: Teacher Guide

Additional resources
- Learn more about the Everyone Can Code program.
- Connect with other educators in the Apple Developer Forums.
- Get App Development with Swift Level 1 certification.

About Swift
Swift is the powerful and intuitive programming language created by Apple for building apps. It makes programming easier, more flexible, and more fun. Swift is not only great for getting you started with coding, it’s also super powerful. It’s designed to scale from writing the simplest program, like “Hello, world!” to the world’s most advanced software.

Learn more about Swift.

About Xcode
Xcode is the Mac app used to build every other Mac app and every iOS app, too. It has all the tools you need to create an amazing app experience. And it’s available as a free download from the Mac App Store.

Learn more about Xcode.

Apple Professional Learning
Apple Professional Learning Specialist offerings are organized into multiple-day engagements over a period of time and are designed to provide:
- Leadership visioning and planning to help administrators prioritize learning objectives
- Guidance from an Apple Professional Learning Specialist who’s dedicated to your project
- Customized, research-based professional learning plans to match learning goals
- Hands-on, immersive learning experiences to help faculty develop innovative instructional practices that engage students
- An opportunity to work with your professional learning leaders to build sustainability

Let Apple Professional Learning Specialists help provide your faculty with the best personal support for innovating their instructional practices in ways that will engage students and enable personalized learning.

To learn more, contact your Apple account executive or send an email to apls@apple.com.