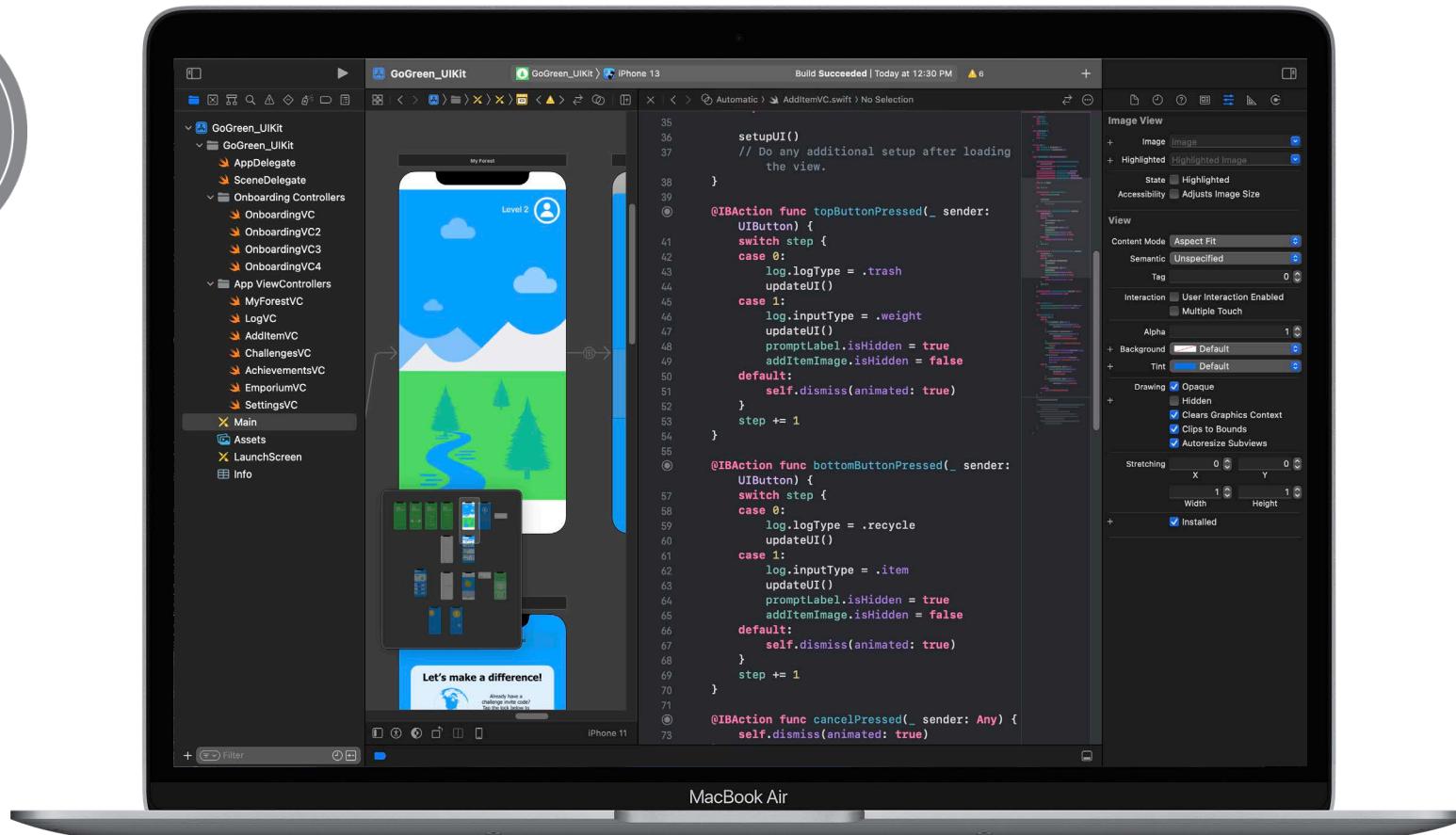




# Develop in Swift AP® CS Principles

## Curriculum Guide

2021–2022 School Year



# Develop in Swift

Develop in Swift is a comprehensive coding offering intended for students in grades 9 and above. The curriculum prepares students for college or a career in app development using the Swift programming language, and is complemented with free online professional learning for educators. Swift is designed for Mac—which supports all major programming languages—making it the ideal device for teaching and learning code.

As students move from Develop in Swift Explorations or AP® CS Principles to more advanced concepts in Fundamentals and Data Collections, they'll explore designing and building a fully functioning app of their own—and can even earn AP® credit or an industry-recognized certification. And for out-of-school coding, the App Design Workbook, App Showcase Guide, and Swift Coding Club help students design, prototype, and celebrate their app ideas.



# Develop in Swift AP® CS Principles Course Overview

The Develop in Swift AP® CS Principles course helps students learn key computing concepts and build a solid foundation in programming using Swift while preparing for the AP® Computer Science Principles Exam. Students will learn about the impact of computing and apps on society, the economy, and our culture as they explore iOS app development. Lessons take students through the app design process: brainstorming, planning, prototyping, and evaluating an app of their own.

## Learn coding concepts and build apps

Using Xcode—the same integrated developer environment (IDE) used by professional developers to create apps on the App Store—students learn the basics of Swift and app development. They'll learn key programming concepts as they write Swift code in Xcode playgrounds, an interactive coding environment that lets them test Swift code and see instant results.



### ChatBot

#### What You'll Build

- A data source object to keep track of a conversation between a user and a chat bot.

#### What You'll Learn

- How apps are built up from instances of different types working together.
- How data sources are used in iOS apps.

#### Key Vocabulary

- Cell**
- Data source**
- Table view**

#### Introduction

ChatBot is similar to the project you worked on in **QuestionBot**. Instead of asking a question, pressing a button, and getting an answer, you'll build up a conversation, similar to a chat in the Messages app.



569

## Explore our connected world

Students are presented with an illustrated story that follows a group of students in a film club and their use of technology to arrange a viewing party, order food, and post to social media. Through these interactive scenarios, they'll gain insight into the technology behind activities such as how computers digitize images and how online activity can be stored and tracked.

**Episode 1: The TV Club | Searching for Content**

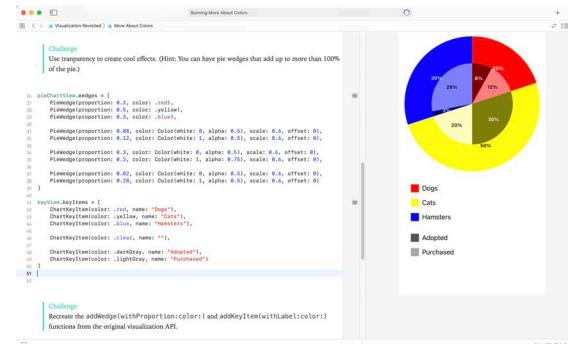
**Searching for Content**

As two of TV Club's officers, Ahlam and Lucas are responsible for recruiting new members and alerting existing members about club updates. Last year they created their own TV Club social media page. It's used to attract new members and allows everyone in the club to post pictures and comments about their favorite shows. This year, the club is focusing on *Finding Alyssa*, a show playing exclusively on a streaming platform called Streaming Plus.

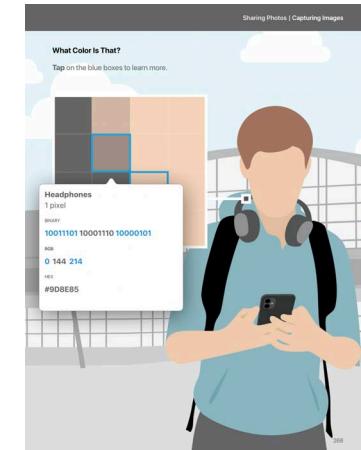
153

# Key Features

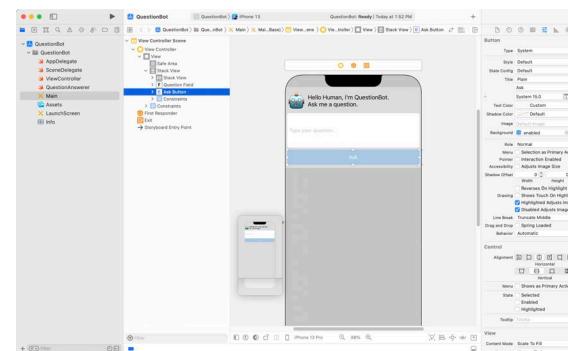
**Xcode playgrounds.** Students learn programming concepts as they write code in playgrounds—interactive coding environments that let them experiment with code and see results immediately.



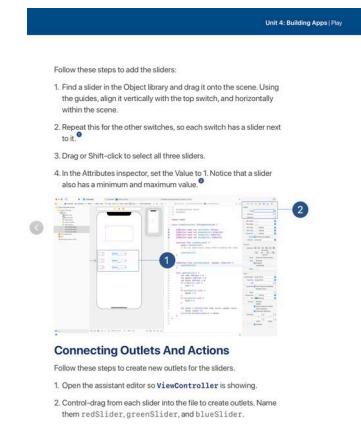
**Interactive widgets.** Using photo galleries, interactive images, and Keynote files, students examine everyday activities and tools—from searching on the web and taking photos to interacting on social media—while exploring the technology behind them and their impact on society.



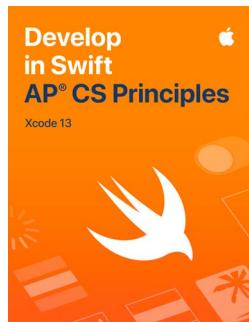
**Guided app projects.** Using the included project files, students can try out certain parts of code without having to build an app from scratch. Supporting images and videos challenge them to apply their knowledge.



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

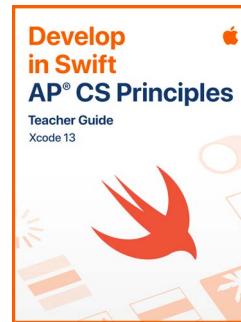


# Course Materials and Outline



## Student guide

Includes access to downloadable Xcode playgrounds and project files, an integrated glossary, and design process activities.



## Teacher guide

Includes additional activities, discussion questions, and activities aligned to the College Board curricular framework for the AP® Computer Science Principles course.

### Unit 1: Values

4 weeks

- 1.1 Get Started with Values
- 1.2 Play with Values
  - 1.2.A Playground Basics
  - 1.2.B Naming and Identifiers
  - 1.2.C Simulation
  - 1.2.D Strings
  - 1.2.E Constants and Variables
  - 1.2.F Word Games
- 1.3 Build a PhotoFrame App
- 1.4 Design for People

### Episode 1: The TV Club

1 week

- E1.1 Searching for Content
- E1.2 Sharing Personal Information
- E1.3 Ordering Online

### Unit 2: Algorithms

6 weeks

- 2.1 Get Started with Algorithms
- 2.2 Play with Programs
  - 2.2.A Functions
  - 2.2.B Types
  - 2.2.C Parameters and Results
  - 2.2.D Making Decisions
  - 2.2.E BoogieBot
  - 2.2.F Data Visualization
- 2.3 Build a QuestionBot App
- 2.4 Design an Experience

### Episode 2: The Viewing Party

1 week

- E2.1 Accessing the Show
- E2.2 Streaming on the Network

### Unit 3: Organizing Data

6 weeks

- 3.1 Get Started with Organizing Data
- 3.2 Play with Complex Data
  - 3.2.A Instances, Methods, and Properties
  - 3.2.B Arrays and Loops
  - 3.2.C Structures
  - 3.2.D Enums and Switch
  - 3.2.E Testing Code
  - 3.2.F Processing Data
  - 3.2.G Pixel Art
  - 3.2.H Visualization Revisited
  - 3.2.I Password Security
- 3.3 Build a BouncyBall App
- 3.4 Design a Prototype

### Episode 3: Sharing Photos

1 week

- E3.1 Capturing Images
- E3.2 Posting on Social

### Unit 4: Building Apps

10 weeks

- 4.1 Get Started with App Development
- 4.2 Play with App Components
  - 4.2.A Color Picker
  - 4.2.B ChatBot
  - 4.2.C Rock, Paper, Scissors
  - 4.2.D Meme Maker
- 4.3 Build an ElementQuiz App
- 4.4 Design for Impact

# AP® Computer Science Principles Curriculum 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 five Big Ideas that encompass foundational concepts of computer science.

## Big Ideas

1. Creative Development
2. Data
3. Algorithms and Programming
4. Computing Systems and Networks
5. Impact on Computing

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 will learn the Essential Knowledge statements to demonstrate an understanding of a particular Learning Objective. In general, each Learning Objective covers about one class period worth of material.

## Computational Thinking Practices

In addition, the AP® Computer Science Principles course contains six Computational Thinking Practices that 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: Computational Solution Design.** Students will engage in this practice as they define and develop their own solutions to exercises in the Xcode playgrounds that accompany the lessons. Choice and collaboration with others guide students' creation and evaluation of computational solutions.

**P2: Algorithms and Program Development.** Students will use language and pseudocode to represent algorithms in various unplugged activities. They'll implement and apply algorithms in Swift throughout the course, particularly in the Create Task portion in which they use Xcode to create an original app or playground.

**P3: Abstraction in Program Development.** Students will engage in abstraction throughout the course. Unit 2: QuestionBot App particularly focuses on the development of functions as a strategy to manage the complexity of code.

**P4: Code Analysis.** Students will troubleshoot and debug sections of their code that don't work. They'll also use Xcode playgrounds to see the effects of their changes in real time, which will help them analyze and fix problems.

**P5: Computing Innovations.** A number of Design activities ask students to discuss the impacts of computing innovations with partners. Students will also explore collecting, analyzing, and visualizing data in Unit 3: BouncyBall App.

**P6: Responsible Computing.** Students explore safe and ethical uses of computing devices, particularly in the episodes between each lesson. The episodes follow the story of a student film club and the various ways they use apps, networks, and social media to plan an event.

## The AP® Computer Science Principles Exam

The AP® Computer Science Principles Exam assesses student understanding of the computational thinking practices and learning objectives outlined in the course framework. The exam consists of the Create performance task and an end-of-course multiple-choice exam.

### Create performance task

The through-course performance task for this course is worth 30 percent of the overall AP® score and requires students to create a computational artifact through programming. As with the AP® Exam, the performance task is 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.

### End-of-course exam

The AP® Exam is given onsite, typically in early May. It consists of 70 multiple-choice questions that test students' knowledge of the Learning Objectives for the course. The AP® Exam is worth 70 percent of the overall AP® score.

## College Board resources

- [College Board curriculum framework](#)
- [College Board course description](#)
- [College Board exam description](#)
- [Assessment overview and performance tasks for students](#)
- [Create performance task and scoring guidelines](#)
- [College Board recruitment strategies](#)
- [College Board AP® Teacher Community \(login required\)](#)
- [Professional development by College Board-Endorsed Providers](#)

# Teaching Code with Apple

When you teach code, you're not only teaching the language of technology. You're also teaching new ways to think and bring ideas to life. And Apple has free resources to help you bring code into your classroom, whether you're just starting or ready to get your students certified in Swift. The [Everyone Can Code](#) curriculum introduces students to coding through a world of interactive puzzles and playful characters with the Swift Playgrounds app. The [Develop in Swift](#) curriculum introduces students to the world of app development by making it easy for them to design and build a fully functioning app of their own design. And Apple supports educators with professional learning offerings to help you get started bringing Everyone Can Code and Develop in Swift offerings to students.

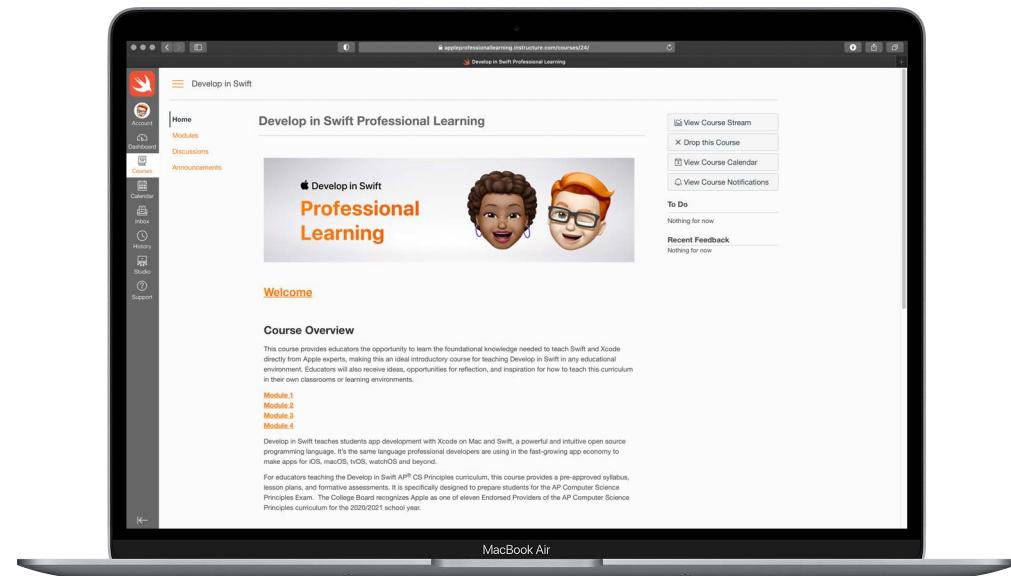
## Free self-paced online professional learning

The Develop in Swift AP® CS Principles course is available through Canvas by Instructure. Participants will learn the foundational knowledge they need to teach Swift and Xcode directly from Apple education experts, making this the ideal introductory course for teaching Develop in Swift in any educational environment. Course content includes an overview of the AP curriculum framework and preparation to teach the Create performance task. Learn more at [apple.co/developinswiftexplorationspl](http://apple.co/developinswiftexplorationspl).

## Bring an Apple Professional Learning Specialist to your school

For educators interested in going further, Apple Professional Learning Specialists organize multiple-day training engagements designed to provide hands-on, immersive learning experiences to help faculty develop innovative instructional practices that engage students.

To learn more about Apple Professional Learning,  
email [AppleProfessionalLearning@apple.com](mailto:AppleProfessionalLearning@apple.com).

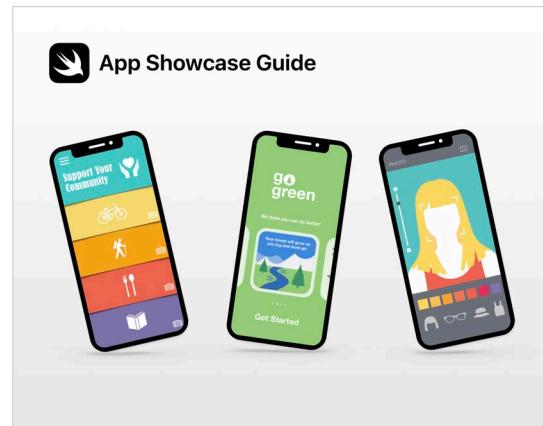


# Additional Resources



## App Design Workbook

The App Design Workbook uses a design thinking framework to teach students app design—a fundamental skill of iOS app development. They'll explore the relationship between app design and coding in Swift through each stage of the app design cycle to bring their app ideas to life. Download: [apple.co/developinswiftappdesignworkbook](http://apple.co/developinswiftappdesignworkbook)



## App Showcase Guide

Celebrate student ingenuity by encouraging students to share their coding achievements with community events, such as project demonstration events or app showcases. The App Showcase Guide provides practical support to help you host an in-person or virtual app showcase event. Download: [apple.co/developinswiftappshowcaseguide](http://apple.co/developinswiftappshowcaseguide)



## Swift Coding Club

Swift Coding Clubs are a fun way to design apps. Activities are built on learning Swift programming concepts in Xcode playgrounds on Mac. Students collaborate with peers to prototype apps and think about how code can make a difference in the world around them. Download: [apple.co/swiftcodingclubxcode](http://apple.co/swiftcodingclubxcode)

# K–20 Coding Resources

Whether students are beginning coders or are ready to build their first apps, Apple has programs to support teaching and learning with Swift, the same programming language used by professional developers to build some of the world's most powerful apps. Everyone Can Code introduces students to the world of coding and app design through interactive puzzles, playful characters, and activities for both in and out of the classroom using the Swift Playgrounds app. Develop in Swift provides students with practical experience designing and developing apps with Swift in Xcode to prepare them for college, career, and even industry-recognized certification in app development.

ELEMENTARY SCHOOL

MIDDLE SCHOOL

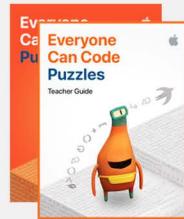
HIGH SCHOOL

HIGHER EDUCATION

## Everyone Can Code



16 hours

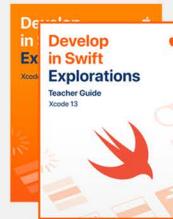


45 hours

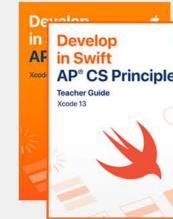


iPad only  
45 hours

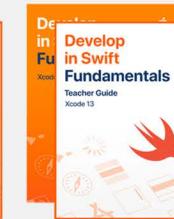
## Develop in Swift



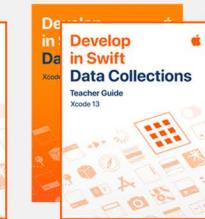
180 hours  
One semester



180 hours  
One semester



180 hours  
One semester



180 hours  
One semester



Swift Playgrounds on iPad and Mac



Xcode on Mac





AP is a registered trademark of the College Board and is used with permission. Features are subject to change. Some features may not be available in all regions or all languages. © 2021 Apple Inc. All rights reserved. Apple, the Apple logo, Mac, MacBook Air, Swift, the Swift Logo, Swift Playgrounds, and Xcode are trademarks of Apple Inc., registered in the U.S. and other countries. App Store is a service mark of Apple, Inc., registered in the U.S. and other countries. IOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license. Other product and company names mentioned herein may be trademarks of their respective companies. Product specifications are subject to change without notice. This material is provided for information purposes only; Apple assumes no liability related to its use. December 2021