Teaching Develop in Swift

Tips and Tricks

Version 1.0.3
October 2022

Author:
Matt Hanlon
Matt is a longtime software developer, mentor, teacher, and coach. He's written software that lives on your laptops, phones, iPads, televisions, and in your browsers. He wants to make sure that you can fulfill the promise that #EveryoneCanCode, just with a bit of work and perseverance. And it's one of his main goals to help you get there.

Code Hub 2022

What do your students want out of knowing how to code? 3

Why Use Apple's Content? 3

Introduction 5

Delivery Options 6

1. As a condensed bootcamp-style course 7
2. As a series of modules over the course of a series of terms 9
3. Theory-focused with hands-on elements 11
4. Explorations Bootcamp 12
5. Hands-on, project-based learning 14

Tools 16

For Your Learning Management System: Common Cartridge 17

How TUM Does It: Artemis 18

Automatic Homework Checking: GitHub Workflows 19

Certification - Students 20

Certification - Institution and Instructors 21

Resources 22

What do your students want out of knowing how to code?

- Do they want to get on the pathway to writing code for a living?
- They want to become better, more knowledgable product managers who understand what it takes to build software?
- Are they a designer, looking for another tool to add to their tool belt?

Why Use Apple's Content?

Apple has a long history of shipping products that delight and inspire people around the world.
Apple designers and engineers have thought long and hard about how a person might use their software or product; countless hours have been put into thinking about how someone might interact with an app and the wide variety of circumstances of those people using the apps. This knowledge has been distilled into design and coding lessons throughout the Apple developer material. Real-life lessons from award-winning apps have been baked into material like the App Design Workbook (available with the Develop in Swift Fundamentals book) and the Human Interface Guidelines (developer.apple.com/design/human-interface-guidelines/guidelines/overview). They will teach you how to build thoughtful, useful apps of your own, based on your own ideas.

They will also teach you how to ship those apps. It was one of the main drivers in all the teams I worked in in my time at Apple: how and when can we ship this? The guidance in the app design workbook that is woven throughout all the Develop in Swift material will help you plan and iterate over your apps and succeed in shipping, instead of getting bogged down in trying to do too much all at once. It will also help you get over app writer's block, and generate some ideas for how to move forward, whether you're in the middle of a project or just starting out.

This is why I use this material: you get the gold standard of advice on how to build apps that people will want to use, and the supporting resources, like the Keynote slides and labs, are all centered around the same stories: where are you going to use this in your coding career and why are we learning about it right now?

Introduction

Apple have a number of resources to help you teach students how to code. In this document we're going to focus on material that suits different learners and how it might be deployed to satisfy their different needs, or your institution's capacity and capabilities.

Apple have provided a free curriculum guide for Develop in Swift available in the Books application which outline their opinion on how the material might be delivered.

There are three books in the series:

Develop in Swift Explorations

http://apple.co/developinswiftexplorations

Teacher Guide: http://apple.co/developinswiftexplorationstg

Develop in Swift Fundamentals

http://apple.co/developinswiftfundamentals

Teacher Guide: http://apple.co/developinswiftfundamentalstg

Develop in Swift Data Collections

http://apple.co/developinswiftdatacollections

Teacher Guide: http://apple.co/developinswiftdatacollectionstg

You could use just these books and their associated materials to teach your students app development. There are labs for students to gain practice with Swift coding concepts, guided projects you can lead in class or have students complete on their own, quizzes to check for understanding, and resources to help guide students' thinking when approaching their own app ideas in order to help them successfully ship a product, at the end of the day. The teacher materials provide you with Keynote presentations you can use to teach concepts in the classroom by putting a good framework and story around each lesson.

If you're searching for an alternative pedagogical approach to educate your students, you might find Challenge Based Learning interesting.

https://www.apple.com/br/education/docs/CBL_Classroom_Guide_Jan_2011.pdf

https://www.apple.com/ca/education/docs/Apple-ChallengedBasedLearning.pdf

Delivery Options

There are many ways to deliver Develop in Swift to your students. We're going to outline a few ways you might teach the material provided, depending on the type of students you have.

For Traditional Coders (with or without a background in coding):

1) I don't have a lot of time, and I want to learn as much as I can, quickly
2) I want a traditional, in-depth, three semester series of courses
3) I already have some coding background, I want to know about Swift and iOS

For Designers, Product/Project Managers:

4) I don't have a lot of time, and I want to learn what I can do with coding and the basics
5) I want to see how coding applies to real-life examples and projects

1. As a condensed bootcamp-style course

"I don't have a lot of time, and I want to learn as much as I can, quickly"

You may choose this approach if you're following the model of Technical University Munich, in which you skill students up on Swift and app development, then spend the remainder of the term focused on building out real world projects with industry (or other) partners. Or you may simply not have an entire term to focus on Swift and app development, but want to offer students a special module to enhance their skills.
You can adjust the time table to fit the available space and resource constraints (rooms, instructors, &c.).
If you are building teams to work on projects following the bootcamp, this condensed time period can serve as a leveling gauge for assigning members to teams.
The sample material selected below for each week is based on a class which has had some background in coding in Swift or another programming language, but could be adjusted to accommodate students who have no coding experience by taking longer with the earlier units.

Prerequisites: Students do not have to have any prior coding knowledge or experience to begin this bootcamp, but this will have an impact on the amount of material you can deliver and how long it will take.

Schedule

Students meet online or in-person with the instructor five times a week for roughly four hours a day.

The instructor holds office hours on specified days and times (eg. Thursday evenings), and participants will be able to book half hour sessions to go over material or questions they may have while working through the course. The instructor also holds code reviews for the students during the week, in which material from the present week's lessons is discussed and critiqued. Ideally, these code reviews progress to student-led code reviews by the end of the camp.
The course will be designed to cover the most common tools students will need to develop an app, namely Swift fundamentals, UIKit and SwiftUI fundamentals, working with the web (JSON, HTTP Requests, async programming), and then additional, project-specific requirements, like ARKit or RealityKit, ResearchKit, CoreML…

Sample Structure(s):

Coders with some Experience:

This structure assumes some familiarity with coding and Swift, potentially, to be able to move quickly through the lessons and exercises and progress to practical, specific lessons in the critical topics for the app or projects for which the class is preparing, after the bootcamp.

Expectation: At the end of this course would be that students would be ready to tackle a project in a team.

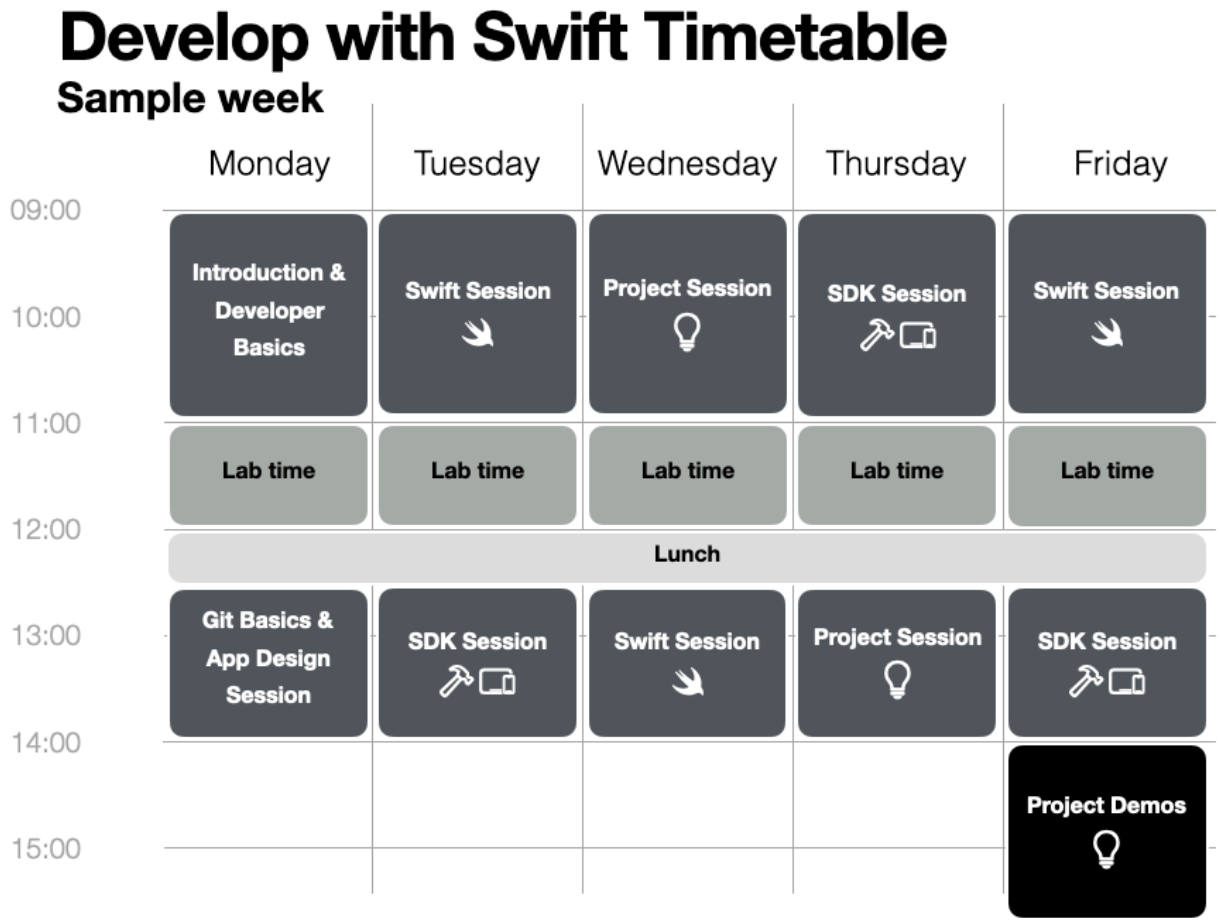
Week 1: Introductions, Developer Basics, Develop with Swift Fundamentals, Units 1 & 2

Week 2: Develop with Swift Fundamentals, Unit 3, Develop in Swift Data Collections Units 1 & 2
Week 3: Develop with Swift Data Collections, Units 2 & 3, SwiftUI
Week 4: App- or industry-specific framework focus, eg. ResearchKit, CareKit, HealthKit, ARKit, CoreML, Server-side Swift, app & feature prototyping

New Coders:

This next structure has been scaled back to address new coders. More time is spent in Develop in Swift Fundamentals, with a cursory look at advanced concepts like protocols, closures, extensions, working with the web, and generics from Develop in Swift Data Collections. Students would be expected to be able to create their own app and navigate documentation to investigate frameworks necessary for features beyond the scope of the four weeks of training. If you are holding this class with the aim of completing projects as a team the students would be capable of working in the project team with some senior guidance.

Simplified Week 1: Introductions, Developer Basics, Develop with Swift Fundamentals, Unit 1
Simplified Week 2: Develop with Swift Fundamentals, Unit 1 & 2
Simplified Week 3: Develop with Swift Fundamentals, Unit 2 & 3
Simplified Week 4: Develop with Swift Fundamentals, Unit 3, SwiftUI, Develop in Swift Data Collections Swift lessons



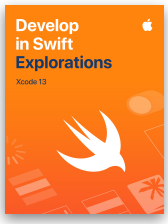
2. As a series of modules over the course of a series of terms

> “I want a traditional, in-depth, three semester series of courses”

This is the method outlined in the Develop in Swift Curriculum Guide. Over the course of three terms you would tackle the three Develop in Swift books.

Prerequisites: Students do not have to have any prior coding knowledge or experience to begin this series.

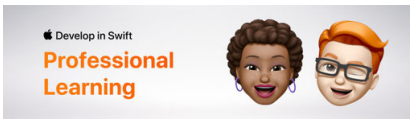
Term One



You would teach Develop in Swift Explorations over the first term, teaching students key computing concepts and getting hands on experience with Swift as a way of learning about the language and app development.

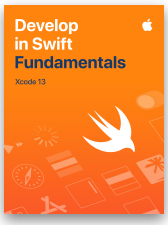
At the end of the first term students will have an array of apps they’ve finished building based on code and materials provided in the book.

Support: For advice and support in teaching Develop in Swift Explorations, Apple have provided an excellent professional learning resource, for which you can register for free here: apple.co/developinswiftexplorationspl



In-class: Class sessions will likely be an hour and a half to two hours, once or twice a week, discussing concepts, performing activities to emphasize the reading, and working through the labs in the classroom.

Term Two



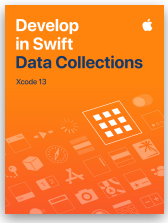
For the second term, students would tackle the more theory-based Develop in Swift Fundamentals. This course involves guided projects in which students build apps from scratch, following instructions in the book. There is ample opportunity for you or your students to design and implement features or components that go above and beyond the core project, to address concerns or a certain theme you may have developed around the class.
By the end of this term students will be able to create an app of their own, based on the design lessons and the fundamental skills and concept they’ve learned.

Prerequisites: Students do not **need** to have any prior coding knowledge or experience to begin this series, but it will make the class pace go faster if they do. If they have done the first term, they should be in good shape for this term.

Support: The teacher materials included with Develop in Swift Fundamentals (https://education-static.apple.com/fundamentals/xcode13/teacher.zip) provide you with Keynote files to present the concepts for almost every lesson, as well as solutions to all of the labs and completed versions of the guided projects. The teacher guide will also give you ideas for delivering the material to a variety of audiences, from new coders to more experienced ones, as well as tips for encouraging collaboration amongst your students.

In-class: Class sessions will likely be an hour and a half to two hours, once or twice a week, comprised of lectures and working through the labs or projects in the classroom.

Term Three



In the third term, students will learn skills related to common iOS app features, like being able to display a group of related items, manage and display data that has been fetched from over the internet and decoded into a useable format. They also learn more sophisticated patterns and tools for advanced development.
These materials lend themselves to a less linear approach as students begin to pick and choose concepts as and when they need them for their own projects. If there is no central theme or driving concepts, students wish to pursue the lessons can be taught in a linear fashion and students will come away with a really rich set of tools to build nearly anything they can dream up.

Prerequisites: Students should have an in-depth grasp of the Swift programming language. If they have completed Term Two they should have the requisite knowledge to attend this course.

Support: The teacher materials included with Develop in Swift Data Collections (https://education-static.apple.com/data-collections/xcode13/teacher.zip) provide you with Keynote files to present the concepts for almost every lesson, as well as solutions to all of the labs and completed versions of the guided projects. The teacher guide will also give you ideas for delivering the material to a variety of audiences, from new coders to more experienced ones, as well as tips for encouraging collaboration amongst your students.

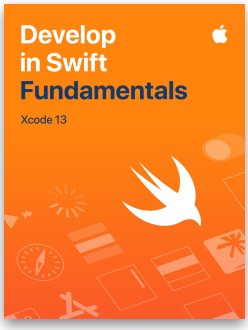
In-class: Class sessions will likely be an hour and a half to two hours, once or twice a week, comprised of lectures and working through the labs or projects in the classroom.

3. Theory-focused with hands-on elements

"I already have some coding background, I want to know about Swift and iOS"

If you want to teach a traditional, lecture-based course, Develop in Swift Fundamentals makes it incredibly easy to do so.

You would focus on Term Two of the second approach, and possibly elements from Term Three, with Develop in Swift Data Collections.



You could rely heavily on the slides from the teacher materials and assign the labs as homework.

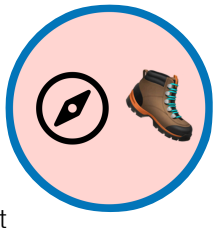
Prerequisites: Students do not have to have any prior coding knowledge or experience to begin this course, but this will have an impact on the amount of material you can deliver and how long it will take.

Schedule

Class sessions will likely be an hour and a half to two hours, once or twice a week, comprised of lectures and working through the labs or projects in the classroom.
In the early units, you may be able to fit two lessons into one class period.

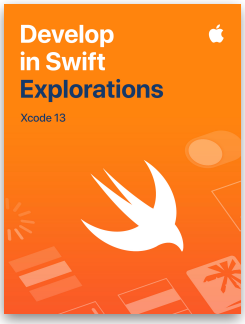
Expectations: By the end of this semester, students should feel comfortable developing their own apps and know where to go to learn more, if their apps require the use of frameworks or patterns beyond the scope of Develop in Swift Fundamentals. They should also be able to pass the App Development with Swift Certified User exam from Certiport.

4. Explorations Bootcamp



"I don't have a lot of time, and I want to learn what I can do with coding and the basics"

You may choose this approach if your students want a quick introduction to Swift and what it takes to ship an app. Relying on the Develop in Swift Explorations material, students will get hands-on experience writing code in Xcode Playgrounds and be able to work on real-world apps.
You can adjust the time table to fit the available space and resource constraints (rooms, instructors, &c.), but ideally you'd have at least two weeks for this bootcamp.



Prerequisites: Students do not have to have any prior coding knowledge or experience to begin this bootcamp, but this will have an impact on the amount of material you can deliver and how long it will take.

Schedule

Students meet online or in-person with the instructor five times a week for roughly four hours a day.

The instructor holds office hours on specified days and times (eg. Thursday evenings), and participants will be able to book half hour sessions to go over material or questions they may have while working through the course. The instructor also holds code reviews for the students during the week, in which material from the present week's lessons is discussed and critiqued. Ideally, these code reviews progress to student-led code reviews by the end of the camp.
The course will be designed to cover many coding concepts you would encounter in professional programming and give your students ample material to practice on and projects to which they can apply their knowledge.

Sample Structure(s):

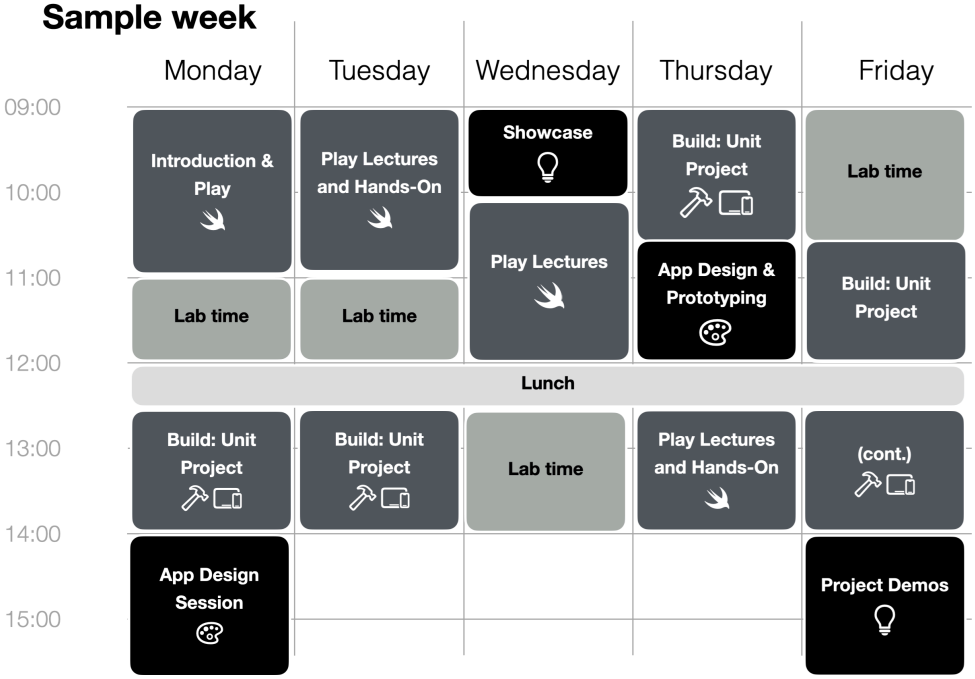
This structure alternates between sessions in which the Get Started and Play sections where the coding concepts are introduced and lab time to practice in the playgrounds provided in the student materials. After lab time for the ideas to digest, there are dedicated times to tackle the guided projects in each unit, from PhotoFrame to QuestionBot to BouncyBall and all the apps in Unit 4. Some sessions have been scheduled to introduce app design and prototyping from the end of each unit. You would ideally have two weeks for this boot camp, at least. If you only had one you could get up to the end of Unit 2 and QuestionBot.

Expectation: At the end of this course would be that students would be able to help out in a project and be able to analyze problems in a project or app and contribute to solutions.

Week 1: Develop with Swift Explorations, Units 1 & 2 Play and Build and Design sections
Week 2: Develop with Swift Explorations, Units 3 & 4 Play and Build and Design sections

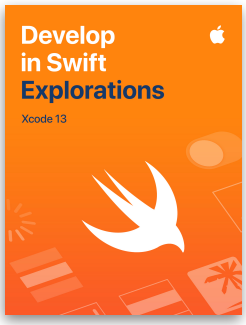
Develop with Swift Explorations Timetable

Sample week



5. Hands-on, project-based learning

"I want to see how coding applies to real-life examples and projects"

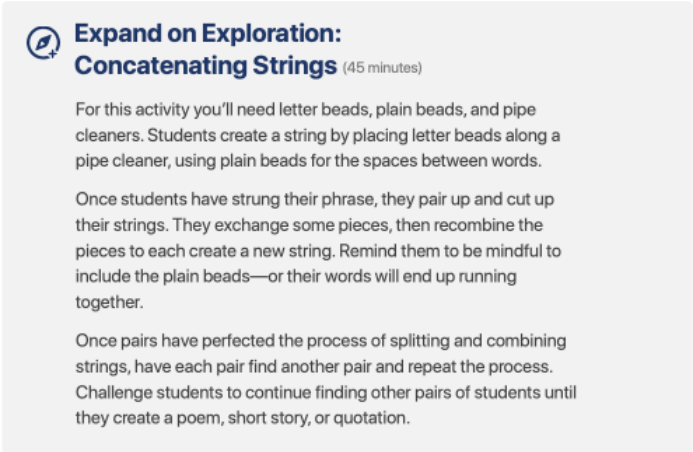


If you prefer teaching a lesson based on doing, Develop in Swift Explorations offers an intensive set of labs, in which students learn the material. This is the first term of the three term approach.

You would focus on Term One of the second approach.

Prerequisites: Students do not have to have any prior coding knowledge or experience to begin this course.

Class periods can be spent covering the activities in the Get Started section of the book. General time guidelines are provided for each activity.



Expand on Exploration: Concatenating Strings (45 minutes)

For this activity you'll need letter beads, plain beads, and pipe cleaners. Students create a string by placing letter beads along a pipe cleaner, using plain beads for the spaces between words.

Once students have strung their phrase, they pair up and cut up their strings. They exchange some pieces, then recombine the pieces to each create a new string. Remind them to be mindful to include the plain beads—or their words will end up running together.

Once pairs have perfected the process of splitting and combining strings, have each pair find another pair and repeat the process. Challenge students to continue finding other pairs of students until they create a poem, short story, or quotation.

Schedule

Class sessions will likely be an hour and a half to two hours, once or twice a week, comprised of working through labs and possibly watching videos from the professional learning resource.

Expectations: By the end of this semester, students should feel comfortable reading code and adapting code to suit their needs. They should feel confident reading the material in Develop in Swift Fundamentals to

take the next step and begin creating their own apps from scratch. They should also be able to pass the App Development with Swift Certified Associate exam from Certiport.

Tools

There are a few platforms and tools that may make teaching app development with Swift easier. These online platforms offer ways to present the material to your students, or to collect and even compile homework assignments.

There are also options for you if your institution already has its own learning management system that supports standards like Common Cartridge.

For Your Learning Management System: Common Cartridge

All three of the books mentioned above are available as Common Cartridge downloads, which you can install in your own Learning Management System. You need a free Canvas account to log in and download these resources.

If you use Canvas to teach your classes, you can import these modules directly into your classroom and assign the quizzes at the end of each lesson to your students to check their understanding. There are no quizzes in the Develop in Swift Explorations material, but there are for both Fundamentals and Data Collections.

Develop in Swift Explorations: https://lor.instructure.com/resources/68832bf58e8e41518839bcd4b707aaf3
Develop in Swift Fundamentals: https://lor.instructure.com/resources/fcddf3e0a8044d57bf4723a273cabee4
Develop in Swift Data Collections: https://lor.instructure.com/resources/ff75923a2be343d8b443c6150cf48474

Use this if:

- Your students have limited access to Macs or Apple devices
- You want to be able to assign the quizzes from the books to be submitted for grading
- You want better insight into the progress of your students with the material
- Your institution has a learning management system and you want to consolidate group discussions around the material

How TUM Does It: Artemis

Technical University of Munich uses their own platform to assign and grade programming exercises and quizzes, with support for manual and semi-automatic grading of other assignment types.
This platform is available as open source on Github, and there is documentation for setting up your own instance.

The platform is configured for TUM's environment, but the documentation lays out what key configurations you need to change, and also what key pieces of infrastructure you need in place to run it. It is a non-trivial task to install and run Artemis and its attendant components, but is a very successful piece of software as running a class like Develop in Swift.

Github: https://github.com/ls1intum/Artemis
Documentation: https://docs.artemis.ase.in.tum.de

Use this if:

- You have limited staff and some technical know-how
- You have the time to set up the system or enlist a consultant to do so
- You want to check student's homework assignments
- You want better insight into the progress of your students with the material

Automatic Homework Checking: GitHub Workflows

If you want to introduce your students to source control and pull requests, you can use the labs from the book in Github. The labs are from the Xcode 12 edition of the material, which do not differ greatly from the Xcode 13 edition (until the Develop in Swift Data Collections labs for async lessons in Unit 2).

There are instructions for using the Github repository at the link below.

One extra feature of the Github repository is the use of Github Actions to build the submissions when pull requests are created. You can use this same action for any Xcode playground assignments you might assign to your students, or use it as a challenge for students to work around the playgrounds that import UIKit (the verifier action only supports code that uses Foundation, not the UIKit framework).

Develop in Swift Fundamentals and Data Collections labs in Github: https://github.com/Teaching-Develop-in-Swift/Labs
The Xcode Playground Verifier Github Action Workflow: https://github.com/mhanlon/xcode-playground-verifier-action

Use this if:

- You have limited staff and some technical know-how
- You have the time to set up the system or enlist a consultant to do so
- You are assigning the labs via Github and want to check students work at a very high level (does it compile?)

Certification - Students

You can offer certification for your students through Certiport.
There are two exams students can take: App Development with Swift Associate and Certified User.
The Associate certification is aligned with the Develop in Swift Explorations course.
The Certified User certification is aligned with the Develop in Swift Fundamentals course.

To become a Certiport Test Center you should see the following link: https://certiport.pearsonvue.com/Educator-resources/Get-started.aspx

You can find out more about the certifications here:
https://certiport.pearsonvue.com/Certifications/Apple/App-Dev-With-Swift/Overview

Certification - Institution and Instructors

Apple offers free training and certification of instructors and institutions through the Apple Authorized Training Center for Education programme.
You can find out more information about this programme here: https://training.apple.com/aatce

Resources

Develop in Swift Explorations
http://apple.co/developinswiftexplorations
Teacher Guide: http://apple.co/developinswiftexplorationstg

Develop in Swift Fundamentals
http://apple.co/developinswiftfundamentals
Teacher Guide: http://apple.co/developinswiftfundamentalstg

Develop in Swift Data Collections
http://apple.co/developinswiftdatacollections
Teacher Guide: http://apple.co/developinswiftdatacollectionstg

Develop in Swift Explorations Professional Learning
http://apple.co/developinswiftexplorationspl

Inclusive App Design Activity
https://appleteacher.apple.com/#/home/rp/T040803A-en_EMEIA

Celebrating You Activity
https://education-static.apple.com/geo/uk/education/swift-playgrounds-guides/everyone-can-code-celebrating-you.pdf

The Code Hub: Map to Coding - different routes to learning to code
https://www.thecodehub.ie/news/map-to-coding/

The Code Hub: App Design
https://www.thecodehub.ie/news/app-design/

Teaching Swift: a Slack community
https://thecodehub.ie/teaching-swift-slack

Other institutions for inspiration and support:

iOS Developer University Program
https://developer.apple.com/programs/ios/university/

TUM: Research Group for Applied Software Engineering
https://ase.in.tum.de/lehrstuhl_1/projects/all-projects

Apple Developer Academy: Naples
https://www.developeracademy.unina.it/en/

Apple Developer Academy: Michigan State University
https://developeracademy.msu.edu/