

Socially Just Coding: Develop in Swift Explorations Pt. 2

Socially Just Coding Pt. 2 is the second course in the four-course series. In this course, you will continue to learn core programming concepts through hands-on activities, Xcode playgrounds, and app projects. You will dig deeper into building your app development skills, through the development of a QuestionBot prototype app using programming in Xcode. Lastly, you will explore how you can personally leverage computing innovations and society through the decisions you make about your actions online and as an app developer.

In this course, students delve into essential computing concepts and establish a strong foundation in Swift programming. They will explore the profound influence of computing and applications on societal structures, economic trends, and cultural dynamics, all within the realm of iOS app development. Guided lessons walk students through the intricate stages of app design, from ideation and strategizing to prototyping and assessment. Even as they hone the skills to transition prototypes into fully-fledged applications, the emphasis on design fosters a deeper understanding of coding and underscores its significance.

HIGHLIGHTS

- **Earn while you learn!** Learners receive stipend upon completion of the microcredential program
- **Live weekly office hours** with course facilitators and student success coaches
- **Insights** from renowned course facilitators and student success coaches
- Generalized **knowledge** of Swift Coding and app development
- Topic-related **activities and assignments** reinforce understanding of concepts
- **Exposure to Apple** technology, applications, and productivity tools
- **Information** connected to existing knowledge and real-world tasks
- Personalized **feedback, support, and network development**

OBJECTIVES

- Demonstrate the ability to create custom types using structs.
- Construct arrays containing a specified number of items and apply loop techniques to process them.
- Identify and implement enums to represent specific sets of related values.
- Produce pixel art, extend a data visualizer tool, and design an algorithm that can categorize password strength levels (e.g., weak, medium, strong).
- Develop and present an interactive game app with at least three different colorful shapes.
- Evaluate and report on at least two computing innovations that have led to unintended consequences, detailing both positive and negative outcomes.
- Draft clear specifications for an app idea and construct an interactive prototype using Keynote, ready for user testing.
- Complete at least two guided projects in Xcode and Interface Builder, each resulting in a functional app.
- Implement user interface elements on a screen, establish coded connections for these elements, and demonstrate handling of at least three different user interaction events.
- Showcase the progression of an app's development through at least three different stages, with evidence of testing at each phase.

OVERVIEW

PLEASE NOTE: This course requires a minimum of 2 - 4 hours per week of engagement with course learning activities, resulting in a maximum of 32 hours to successfully earn your course badge. **It is recommended not to take this course with any other Propel course.**

MODULE 1: Instances, Methods and Properties

Build on your knowledge of types by learning about the methods and properties that make up an instance of that type. You learn the difference between a type and an instance, and learn how to use the Xcode documentation viewer to find information about unfamiliar types, properties, or methods.

Learning outcomes

- Create values without using literals.
- Describe the difference between a type and an instance.
- Describe the difference between a method and a property.
- Use documentation to find out information about unfamiliar types"

MODULE 2: Arrays and Loops

Learn how arrays provide a common way to group objects in an ordered list. You will learn how to create and work with arrays by adding and removing objects. Also, you learn how to use a for...in loop to work with each object in an array.

Learning outcomes

- Develop data abstraction using arrays to store multiple elements.
- Explain how the use of data abstraction manages complexity in program code. Understand and explain common use cases for arrays.
- Use arrays to manage collections of objects.
- Iterate through an array to perform a common action on each item in the collection.

MODULE 3: Structures

In Module 3 You will recognize that it is often useful to group related information and functionality into a custom type and learn how to define a custom type by using a structure with accompanying properties and methods. In practice, you will be creating a custom Song type that has related properties to define the title, artist, and duration of each instance.

Learning outcomes

- Understand and explain the importance of custom structures and their common use cases.
- Design a custom structure to group related data into one type.
- Define custom properties and methods for custom structures.

MODULE 4: Enums and Switch

Enumerations, or enums, are a way to define a named list of options. You will learn what enums are used for, how to define them, and common ways to work with them. Also, you will learn to use the switch statement to conditionally run blocks of code based on the value of an enum instance.

Learning outcomes

- Understand and explain the benefits and proper use cases of enumerations, or enums.
- Use a switch statement on an enum value to conditionally trigger specific code.”

MODULE 5: Testing Code

Due to the nature of computer technology, many kinds of basic values have limits. Students will discover the limits of two types—Int (and its various flavors) and Double. They’ll learn how to evaluate the way functions might fail, and think about the inputs and outputs of functions.

Learning outcomes

- How built-in types are not perfect representations of the data they represent.
- Why documentation and testing are important.
- How to test a function.

MODULE 6: Processing Data

Process a large data set from a hypothetical survey. Data transformation and data translation are the two key techniques used in this lesson. Regardless of the presence of errors (or lack thereof), both are common techniques when working with large amounts of data.

You will also encounter third-party code and use it to solve a problem. Using code that’s not your own is an almost universal practice—it’s hard to find a program that’s completely self-reliant. The key to using third-party code is to verify that it’s safe and sound, and to abide by any license terms that accompany it.

Learning outcomes

- Tally basic information in a large data set.
- Identify and correct simple errors using a variety of data transformation techniques.
- Use and cite third-party code.
- Translate data to derive statistical information.

MODULE 7: Bouncy Ball App

Build a physics-based game that uses touch interactions—a much more complex app than those in previous lessons. You build the app in stages, stopping along the way to verify that their code works. Also, you will refactor your code multiple times, reorganizing it to make it more flexible or to enable new functionality without changing how it works.

Learning outcomes

- Follow the incremental development process to build an app in small stages, testing their code after each step.
- Explain the benefits of incremental development.
- Refactor their code to keep it readable and organized.
- Describe how callbacks enable code to be triggered by events.
- Use functions as callbacks.

MODULE 8: Experiment with Building Apps Part 1

Complete a Rock/Paper/Scissors game from scratch. You will review a variety of concepts covered in Units 1-4, and build the user interface, the model data, and the controller objects that make up the entire application.

Learning outcomes

- Apply the Model-View-Controller (MVC) design pattern.
- Use an enum to model simple app state.
- Show and hide user interface elements as the app's state changes.
- Randomize game play.

MODULE 9: Experiment with Building Apps Part 2

You made it! This is your final project. You will complete an app for studying and quizzing with the periodic table of elements and apply the skills you've acquired throughout the course, using the incremental development process as they build the app from a simple interface into a full-featured app with two modes. Also, you learn about how to handle keyboard input and use the standard iOS alert dialog to present the user's score.

Learning outcomes

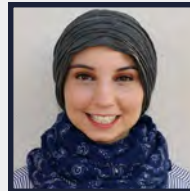
- Follow the incremental development process over the course of a complex app.
- Refactor their code multiple times as they add features.
- Track app state with multiple properties.
- Use the single-path UI update pattern to modify the user interface as the app's state changes.
- Handle keyboard input with a text field.
- Display an alert and handle its completion callback.

FACULTY



DR. KESHA MALLORY JAMES is the Director of Distance Education for Lawson State Community College and the LSCC Apple Tech Hub Program Manager. She led the LSCC Distance Education Program to rank #2 as one of the Best Online Community College Programs in 2019.

With her vision, LSCC became 1 of 10 Apple Tech Centers for HBCUs, and she has over 12 years of experience as an instructor in the Business and Information Technologies Department. Dr. James holds a Doctor of Philosophy from Auburn University and is a graduate of Alabama State University. Dr. James is a certified Apple Teacher and Swift coding instructor.



DR. YASMEEN RAWAJFIH is an Assistant Professor in the Computer Science (CS) Department at Tuskegee University. She received her PhD in Computer Science and Software Engineering from Auburn University. Dr. Rawajfih teaches many core courses in the CS curriculum at Tuskegee

University including Software Engineering, Design and Analysis of Algorithms, Statistics, and Data Analytics.