

iOS Mobile Application Programming

Fall & Spring 202X

Online Anytime

Instructor Contact Information

Instructor	Jarrett Lindsey	Office Phone	713-xxx-xxxx
Office	XXX	Office Hours	XX: XX
Email	XXX@gmail.com	Office Location	XXX

Please feel free to contact me concerning any problems that you are experiencing in this course. Emailing me is the best way to contact me. I'm available to meet you on Mondays/Wednesdays at Stafford campus. My schedule may change at some point during the semester so please email me to verify.

Instructor's Preferred Method of Contact

Please use the Canvas Inbox email tool. I will respond to emails within 24-48 hours, Monday through Friday; I do check my emails during the weekend, but if I don't get to your email, I will respond to them by Monday.

What's Exciting About This Course

The iOS Application Programming introduces iOS programming using Swift. Topics include Install and configure development tools, iOS development life cycle, develop iOS apps using Swift and API, test and deploy apps using simulator and physical devices.

My Personal Welcome

Welcome to iOS Application Programming. My name is Jarrett Lindsey. I have a master's degree in Applied Digital Learning. I worked as a teacher for almost 6 years before implementing this program.

This course introduces UIKit, View Controller Life Cycle, Navigations, Table Views and Data Persistence, and a touch of Swift UI.

Please read the rest of this syllabus for the course description, pre-requisites, student learning outcomes, required textbook, instructional material, course assignments/assessments, as well as other course policies (participation, makeup, etc.). Pay special attention to the Course Calendar section shown below for assignments/assessments due dates.

As the course progresses, you may encounter challenging ideas or difficulties completing your coursework. I am available to support you. The fastest way to reach me is through the Canvas Inbox e-mail. If, for any reason, you can't access Canvas, you can reach me at my e-mail. The best way to really discuss issues is in-person. I'm available during posted office hours to tackle the questions. My goal is for you to walk out of the course with a solid understanding of iOS programming, so please visit me, or contact me by email, whenever you have a question.

Prerequisites and/or Co-Requisites

While this is an introduction to mobile development, it is not an introductory programming course. Accordingly, students need to have:

- Must have completed 2 Computer Science Courses prior to taking this one, including at least 1 Fundamentals course and/or Advanced Placement (AP) course. If you have enrolled in this course having satisfied these prerequisites, you have a higher chance of success than students who have not done so.
- Working knowledge of at least one object-oriented programming language such as Java, JavaScript, or C++
- A firm understanding of how to compile code, use libraries, use a debugger, and

This is a class about Apple's iOS platform, so the student must have a Macintosh laptop running a current version of the operating system with the most recent version of Apple's XCode IDE installed. (Students will need to have Mac OS 10.15 "Catalina" installed). Please note that it is not possible to use a Windows or Linux computer for this course as code written on those platforms cannot be deployed to either an iOS simulator or device.

Canvas Learning Management System

This course will use [Canvas](#) to supplement in- class assignments, exams, and activities. HCCS Open Lab locations may be used to access the Internet and Canvas. **USE [FIREFOX](#) OR [CHROME](#) AS THE INTERNET BROWSER.**

MacBook Check-out Process

You need a MacBook with the latest XCode installed for this course. (Our Institution) will provides MacBook Air and iPad for students to check out if you do not have a MacBook. Details will be discussed in class.

Note: An **Apple ID** is required to install software on your MacBook. MacBook must be returned on the last day of the class. If you decide to withdraw from this class, you need to notify the instructor and return the MacBook to the library of the campus where class is held.

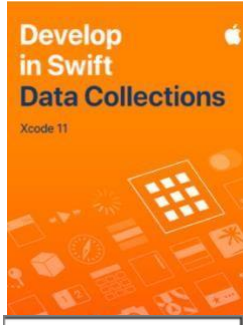
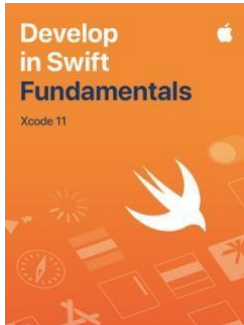
Scoring Rubrics, Sample Assignments, etc.

Look in Canvas for the scoring rubrics for assignments, samples of class assignments, and other information to assist you in the course.

Instructional Materials

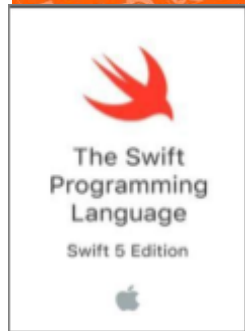
Textbook Information

The textbook listed below is required for this course.



"Develop in Swift Fundamentals" iOS 13 and XCode 11. It is free download from Apple Book Store.

"Develop in Swift Data Collections" iOS 13 and XCode 11. It is free download from Apple Book Store.



"The Swift Programming Language" Swift 5/5.1 Edition or higher by Apple. It is free download from Apple Book Store. Can also be found at swift.org.

MacBook Check-out Process

You need a MacBook with the latest XCode installed for this course. (Our Institution) will provides MacBook Air and iPad for students to check out if you do not have a MacBook. Details will be discussed in class.

Note: An **Apple ID** is required to install software on your MacBook. MacBook must be returned on the last day of the class. If you decide to withdraw from this class, you need to notify the instructor and return the MacBook to the library of the campus where class is held.

Course Overview

iOS Mobile Application Programming introduces iOS programming using Swift. Topics include Install and configure development tools, iOS development life cycle, develop iOS apps using Swift and API, MVC design pattern, test, debug and deploy apps using simulator and physical devices.

This course introduces the basics of contemporary mobile application development using Apple's iOS technology as the development platform. The objective of the course is to enable the student to build an iOS application using the standard Apple tool chain. Requirements for the course will be met by the student demonstrating an ability to develop an application which implements a coherent data model, a view with layout, drawing and touch handling, asynchronous network access and timer management, animation, and proper data flow design along with usage of several the standard Apple-provided user interface components.

We start with the Swift programming language and portions of its system library. Language features which will be most focused on will include: the Swift type of system (tuple/enum/struct/class/func/protocol), closures, optionals, and generics. This basis will then lead to discussion of various functional programming techniques in Swift centered around the map/reduce and other higher order functions. As part of our survey of Swift, we will cover in detail Apple's Combine functional reactive programming layer, in particular its usage to perform network access and timed event handling.

We then proceed to study the architecture of an application which uses Apple's new SwiftUI toolchain. The emphasis will be on creating a unidirectional dataflow architecture from many small elements which can be composed into an overall architecture for the application. Specific attention will be paid to proper practices for coding each of the various portions of an app and techniques for communicating between them using a (State, Action) -> (State, Effects) paradigm. Additional Swift language features such as protocol extensions and their proper use will be added to the student's skill set during this section.

We continue by surveying the major features of the XCode IDE focusing on building interfaces with the XCode Preview tool. Specific techniques to be explored will include layout, view hierarchy's, and Apple's SwiftUI DSL. We continue by exploring various drawing functions in SwiftUI Views, event handling, Bezier Path handling in SwiftUI. Finally, we explore various techniques in the implementation of touch handling. We also discuss integration of custom views within XCode Preview.

As an aside during this section, time permitting, we will discuss Apple's run time analysis tool, Instruments. This will be used to explore the details of the Apple memory model and performance debugging.

Program Student Learning Outcomes (PSLOs)

- Understand the fundamental principles of programming, including those of algorithm analysis, software design, operating systems, and database.
- Design and write computer programs that are correct, simple, clear, efficient, well organized, and well documented. Know and be able to apply important data structures and algorithms
- Identify the hardware and software aspects of computer systems that support application software development.
- Demonstrate knowledge of technology applicable to the field, and a proficiency in appropriate software.
- Technology Knowledge/Proficiency -Demonstrate knowledge of technology applicable to the field, and a proficiency in appropriate software.

Course Student Learning Outcomes (CSLOs)

The student will be able to:

- Install and configure mobile device development tools.
- Work with a programming language that supports the underlying architecture and the mobile framework.
- Identify the role of various components of mobile system architecture.
- Develop entry level native apps implementing various types of layouts.
- Follow good practice of application development life cycle.
- Produce concept documentation and testing strategies.
- Deploy mobile apps to emulators and physical devices.
- Execute deployment procedures for specific mobile devices platform.

Learning Objectives

Learning Objectives for each CSLO are under development.

Student Success

Category	Weight
Quizzes	10%
Programming Assignments	35%
Exams (2 exams)	30%
Final Exam	25%
Final Project	200%
Total	100%

Grade	Total Points
A	100-90
B	89-80
C	79-70
D	69-60
F	< 60
Fx	Failed Due to lack of participation (can affect financial-aid status)

Expect to spend at least twice as many hours per week outside of class as you do in class studying the course content. Additional time will be required for the assignments. The assignments provided will help you practice the concepts discussed in class lectures and hone your hands-on skills. Successful completion of this course requires a combination of the following:

- Reading the textbook
- Attending class in person and/or online
- Completing assignments
- Participating in class activities

There is no short cut for success in this course; it requires reading and studying the material and more importantly completing the assignments.

ASSIGNMENTS AND READINGS

Frequent small assignments progress from basic programming to realistic app development with a focus on responsive device graphics and algorithms. We will build a working app covering all topics as a final project. The final project will reuse code from each of the assignments, so please be aware: the final project is cumulative over all the material discussed in the class, failure to absorb any of the basic concepts mentioned above will be very detrimental to the student's outcome. Readings will come from a combination of: The Apple Developer Library; Chapters from e-Books; and online blog posts and tutorials.

GRADING

Each of the 4 regular assignments is valued at 100 points, the final project at 200 points. It should be noted that the eventual grade of the student will depend most heavily on building the

application for the final project. If the student can demonstrate mastery of all the techniques in the final project, they may have their grade be based solely on the final project. The instructor will grade on the higher of the overall average across all assignments, including the final project or of the final project grade alone. When grading programming assignments, a strong emphasis is placed on good coding practices, code design and correct implementation of platform-specific architectural principles. These criteria are to some extent subjective but expectations for them will be clearly specified in class.

LATE POLICY

As the course builds on previous material and students will receive a working version of code immediately following the due date for each assignment, there must be a zero-tolerance late policy for all assignments.

TOOLS

XCode and InterfaceBuilder and git. Class assignments (which include readings) will be hosted on Canvas and lecture materials will be hosted on the instructors GitHub account which will be publicly available. As part of Assignment 0 to be completed before the first class, each student will need to set up an individual repository, with a username and password, into which they will submit all assigned work. The naming conventions will be spelled out so that all submissions have the same directory structure. The first class will review how to obtain all materials to verify that students have successfully completed Assignment 0. For more information on git and GitHub see:

- <https://services.github.com/on-demand/>
- <https://guides.github.com/activities/hello-world/>
- <https://guides.github.com>

ONLINE DISCUSSION

We will use the discussion forums as provided in the Canvas platform to allow students to communicate with the instructors, TAs, and each other. All questions about the course material should be directed to this forum:

- for maximum visibility to staff and students
- to assure that all students are doing their own work
- to make sure that everyone has equal access to all resources.

While you are welcome to read additional material on the internet, asking questions using outside sources for assistance such as StackOverflow will be considered an honors violation and treated accordingly.

LECTURES AND WORKLOAD

Lectures will be heavily example driven. The final project requires cumulative knowledge drawn from each lecture, so attendance, while not part of your grade, is EXTREMELY important. For a well-prepared student, the workload is ~2 hours of reading and ~2 hours of programming/per week, with the final project being roughly 15 hours of effort.

Instructor and Student Responsibilities

As your instructor, it is my responsibility to:

- Provide the grading scale and detailed grading formula explaining how student grades are to be derived
- Facilitate an effective learning environment through learner-centered instructional techniques
- Provide a description of assignments
- Inform students of policies such as attendance, withdrawal, tardiness, and making up assignments
- Provide the course outline and class calendar
- Arrange to meet with individual students as needed

As a student, it is your responsibility to:

- Attend class in person and/or online
- Be on-time!!
- Participate actively by reviewing course material, interacting with classmates, and responding promptly in your communication with me
- Read and comprehend the textbook
- Complete the required assignments and exams
- Ask for help when there is a question or problem
- Keep copies of all paperwork, including this syllabus, handouts, and all assignments
- Be aware of and comply with academic honesty policies.

Course Calendar – Fall 202X

Week starts	Session	Topics	Reading Assignments
Feb 16/22	1/2	Course Overview, Orientation Install and configure XCode Module 1- Intro to Swift and iOS environment Swift: Basic programming constructs iOS: XCode, Build, Run and Debug iOS App	DSF 1.1-1.4 DSF 1.5-1.8
Mar 1	3	Module 2 Swift: Functions, Structures, Classes, Collections, and Loops iOS: Intro to UIKit, Controls, Auto Layout and Stack Views	DSF 2.1-2.6 DSF 2.7-2.10
Mar 8	4	Module 3 Swift: Optional, Guard, Enumerations iOS: Segues and Navigation Controllers, Tab View Controller, View Controllers Life Cycle Review for Exam 1 Assignment 1 Due	DSF 3.1-3.5 DSF 3.6-3.9
Mar 22	5	Module 3 Swift: Optional, Guard, Enumerations iOS: Segues and Navigation Controllers, Tab View Controller, View Controllers Life Cycle Exam 1 – covers Modules 1-2	DSF 3.1-3.5 DSF 3.6-3.9
		Mar 15 - 21 Spring Break (Office Closed)	
Mar 29	6	Module 4 Swift: Protocol, Delegation iOS: App Anatomy and Life Cycle, MVC (Model View Controller), Scroll Views, Table Views Assignment 2 Due	DSD 1.1 DSD 1.2-1.5
Apr 5	7	Module 4 Swift: Protocol, Delegation iOS: App Life Cycle, MVC (Model View Controller), Scroll Views, Table Views	DSD 1.1 DSD 1.2-1.5
Apr 12	8	Module 5 Swift: Error Handling iOS: Intermediate Table Views, System View Controllers, Saving Data, Complex Input Screens Review for Exam 2 Assignment 3 Due	DSD 1.6-1.9
Apr 19	9	Exam 2 – covers Modules 3-5 Module 5 iOS: Intermediate Table Views, System View Controllers, Saving Data, Complex Input Screens	DSD 1.6-1.9
		Apr 15 - Last Day for Administrative/Student Withdrawals	
Apr 26	10	Module 6 iOS: Introduction to Swift UI	
May 3	11	Module 6 iOS: Introduction to Swift UI Module 7 iOS: Deploying your app Review for Final Assignment 4 Due	
May 11-12	12	Final Exam Return MacBook Air	All Modules
		DSF: Develop in Swift Fundamentals iBook DSD: Develop in Swift Data Collections iBook	

Course Calendar – Spring 202X

Dates	Session	Topics	Reading Assignments
Sep 21	1	Course Overview, Orientation, Assign MacBook Air Install and configure XCode Module 1- Review Swift and iOS Framework	
Sep 28	2	Module 2 – Collection Views, Split View Controllers, Auto Layout Collection Views - Protocols, Flow Layout, Drag and Drop Split View Controller – Master-Detail and Popovers Auto Layout – Constrains, Size Inspector, Trait Variations, Size Class Project Assignment	Collection View Split View Controller Auto Layout
Oct 5	3	Module 2 – Collection Views, Split View Controllers, Auto Layout Collection Views - Protocols, Flow Layout, Drag and Drop Split View Controller – Master-Detail and Popovers Auto Layout – Constraints, Size Inspector, Trait Variations, Size Class	
Oct 12	4	Module 3 – Multitouch and Multithreading Touches, Taps and Gestures Grand Central Dispatch – global queues and main queue Review for Exam 1	Multi-Touch Tap Gestures Dispatch
Oct 19	5	Module 3 – Multithreading Example Exam 1 – covers Modules 1-3	
Oct 26	6	Module 4 – Data Persistence and Document Core Data - NSManagedObjectContext Cloud Storage using Cloud Kit Document – UIDocument, UIDocumentBrowserViewController	Core Data Model CloudKit UIDocument
Nov 2	7	Module 4 – Data Persistence and Document Core Data - NSManagedObjectContext Cloud Storage using Cloud Kit Document – UIDocument, UIDocumentBrowserViewController	
Nov 9 - Last Day for Administrative/Student Withdrawals			
Nov 9	8	Module 5 – Drawings and Animations Core Graphics and Core Animation - Frame, Bounds, Transform	Core Graphics Core Animation
Nov 16	9	Module 6 – Multimedia Video, Audio, Speech Recognition and Media Controller Review for Exam 2	Core Media
Nov 23	10	Exam 2 – covers Modules 4-6 Module 7 – Working with Maps MKMapItem and Geocoding MapKit and MKMapView Core Location Framework	MapKit
Nov 30	11	Module 7 – Working with Maps MKMapItem and Geocoding MapKit and MKMapView Core Location Framework App Distribution, Project Presentations , Review for Final	Core Location
Dec 8-9	12	Final Exam	All Modules

Syllabus Modifications

The instructor reserves the right to modify the syllabus at any time during the semester and will promptly notify students in writing, typically by e-mail, of any such changes.

Instructor's Practices and Procedures

Missed Assignments

Other than the final exam, late work is accepted, with a penalty, unless proper documentation is provided.

Academic Integrity

Scholastic Dishonesty will result in a referral to the Dean of Student Services. See the link below for details. Students may ask questions to other students, to me, or to anyone else. This is how we learn, and I encourage it. HOWEVER, all work must be started and completed in its entirety on your own. If it is found that students are sharing the same files, and then making minor changes to submit the work as their own, students will receive a 0 on the assignment and be referred to the Dean of Student Services.

Attendance Procedures

1. On-campus students may be withdrawn if the student is absent for more than 12.5% of class. See the Student Handbook for more information.
2. Online students may be withdrawn before the Official Day of Record if they have not logged in or shown sufficient participation. See the Online Student Handbook for more information.
3. As an Online section or On-Campus section of this course, you must make satisfactory progress in this course. Students may be withdrawn if the student misses turning in assignments (Labs, quizzes, or discussions) that total to more than 12.5% of the course work prior to the Final Exam.
4. Contact your instructor if you are having a problem.
5. If you decide to quit participating in the course before the Last Day for Administrative/Student Withdrawals, you may withdraw yourself, or ask the instructor to withdraw you, or in some cases the instructor may withdraw you without your approval if you have not attended and participated. Talk to you instructor for details.
6. After the withdrawal date deadline, the instructor is not able to withdraw you.
7. You may receive an FX as your grade due to lack of participation if you have not withdrawn yourself. If you receive an FX, you may or may not have to return the financial aid.

Student Conduct

Two simple rules:

- (1) Students should be respectful to everyone in the classroom.
- (2) Students should come to class on-time, prepared, and ready to learn.

Instructor's Course-Specific Information (As Needed)

WHERE DO I BEGIN?

1. **ON CANVAS, READ THE MAIN PAGE OF OUR CLASS WEBSITE** and then click on **MODULES**. Start with the first module, read each link in the module, and proceed to the second module.
2. **Download iBooks and Activities** as soon as possible.
3. **PRINT OUT THE COURSE CALENDAR** found in this syllabus. Mark off each assignment as completed. This will help you stay on schedule.
4. **EMAIL ME VIA THE ONLINE INBOX TOOL.** If you cannot access the Eagle Online (Canvas) Inbox tool, you may send me a message via your normal email provider to jessica.ku@hccs.edu, making sure to add your NAME and COURSE and CRN *in the Subject line. Emails received without this information will not be answered.*
5. **POP Ups** If a pop-up is blocked, you will need to set your pop-up blocker to allow pages from our Eagle Online site. More information is in the first module of the course. Do this before beginning the course as many of the early links in the course are pop-ups!
6. The hardest part of this course is completing all the work on-time. Do not procrastinate. Work ahead! Have fun!

Electronic Devices

Two simple rules:

- (1) Cellphones should not be used during class, unless to test your App.
- (2) Do not use the computers for non-course activities during lecture.

