iPhone App Programming Course

Work harder on yourself than you do on your job. – Jim Rohn

If you don't design your own life plan, chances are you'll fall into someone else's plan. And guess what they have planned for you? Not much.
– Jim Rohn

Revision History
Shawn Arney – Document Creation – 10/26/2011 – Version 0.1
Shawn Arney – Completion 11/17/2011 – Version 1.0

What is this course about?

This is the three-day course I teach to businesses who want to teach their programmers how to do iPhone programming.

Cost of the Live Course

The live course I teach is customized to the particular business' programmer needs. This video course is more generic, but still teaches the fundamentals of that class.

I charge businesses $3,000 per programmer for the customized, live-course.

If your company would like live training, please contact us at:

Shawn@LearnApps.Org

Or visit our website for more details: http://www.LearnApps.Org

Requirements

This course requires an Apple Macintosh computer that has OSX Lion installed.

You'll need to install the Xcode software. We have a free tutorial on how to do this.

Don't have a mac? Don't want to buy one yet?

You can **rent** a mac here (~$20/month):

http://www.macincloud.com/

Who is this course for?

You should be familiar with programming and Object-Oriented Programming.

If you have not programmed before, please consider learning 'introductory' programming via Khan Academy's computer science topics:
http://www.khanacademy.org/#computer-science

And Harvard's introductory course:
http://academicearth.org/courses/introduction-to-computer-science-i

List of Free Computer Science Books:
http://stackoverflow.com/questions/194812/list-of-freely-available-programming-books

Results and Goals

I have taught this material to programmers and they have quickly been productive for their companies. I understand that the price of this course seems high. These concepts and the material presented here have been developed over the past 3 years.

By taking advantage of these video courses that start from the basics, using a see-and-do model, you should quickly learn by doing. If anything, you will be learning ahead of the 'standard learning' curve for new iPhone programmers.

BEAT THE LEARNING CURVE

Even though I am a professional software developer, who has been developing commercial software since high school, it took 6 months of reading iPhone programmer books and creating iPhone apps to 'get it'.

FAST TRACK YOURSELF TO A PUBLISHED APP

Programmers I have taught this material to, have beaten this learning curve by months. You will save time, which is money... you could be making on apps! I am sure you will 'get it' quickly, through this video course as well.

I envy you and the investment you have made in yourself and your future career in iPhone app development. This course will bootstrap you and enable you to be productive quickly as an iPhone app developer. I'll lead you by the hand from the basics to a published app in the app store.

Cheers to your success,

Shawn Arney - professional iOS Developer and trainer

Questions And Help During the Course?

Send questions about any video or topic to: Shawn@LearnApps.org

Need Mentorship?

If you find you need hands-on-training with a mentor, please inquire about our monthly mentorship program: Shawn@LearnApps.org

About the Course
Learn how to create iOS apps from start to finish. We crack this nut using a see and do approach. We will walk you thru the app creation process, teaching you the basic concepts and essentials to iPhone programming.
Your instructor is a professional software developer who developed iPhone banking apps for over ~30 Credit Unions. Shawn taught professional software developers how to develop professional, 'commercial' apps as well.
Benefits of the course include the following:
You'll learn how to use Xcode and the iOS SDK to create professional apps like a pro
Gain hands on coding experience as I walk you thru each coding exercise
Gain confidence and skill as an iPhone app developer as you complete each app exercise
Save time by learning with the guidance of an instructor who will walk you thru iOS' normally steep learning curve
Avoid frustration by being led thru simple app exercises that teach the fundamentals and concepts behind app development
Learn at your own pace and revisit tutorial videos to 'get it'
You'll come away from this course with an understanding of how to create iPhone apps. You'll be able to create your first iPhone app for the app store. You'll be creating dozens of apps in this course as we guide you thru iPhone app development.

Course Topics
Xcode 4 Tips and Tricks: Xcode is the editor and tool (IDE) used to develop iPhone and iPad applications. You'll learn how to use it effectively and you'll learn shortcuts and tips to make your app development quicker and easier Tips like: how to manage schemes and workspaces, create code snippets for the code you use often, navigate within the editor, quickly switch between source implementation and definition files, searching for files, where to get it and how to install it
An Introduction to Objective-C and Foundation Classes: Objective-C is the programming language that iOS applications are written in. You'll learn the basics and this will help jumpstart your app development. You'll learn about: Classes including Object Oriented Programming topics such as inheritance, polymorphism and properties and how to use them Messaging and Selectors and how this is central to Objective-C Foundation Classes and when to use them including: NSString, NSData, NSDate, NSArray, NSDictionary
Cocoa Touch Design Patterns including MVC Cocoa Touch is the collection of libraries used for creating apps. The design patterns you'll learn include model-view-controller (MVC), and delegation using protocols. These fundamental topics will enable you to become a skilled and competent iOS developer.
Memory Management Fundamentals You'll learn how to manage your memory so that you don't crash your app or consume too much memory causing iOS to kill it You'll learn the concepts behind alloc, release, and autoreleasing your objects You'll learn about autorelease pools to lower memory consumption, an important feature for making your app perform well for your app's users You'll learn how to find memory issues pre-emptively before your app reaches the app store You'll learn debugging tips on how to fix an app that is crashing and an app that is consuming too much memory. Tools and tips including: NSZombies for troubleshooting crashes and Clang for static memory analysis
Human Interface Guidelines Throughout the course, you'll learn about Apple's requirements for your app and how to conform to their standards of quality.
Application Life Cycle

Application Life Cycle
Understand the iOS App's life-cycle and how it affects your app.
You'll learn how to respond to Application events so your app behaves properly and in conformance with Apple's requirements and guidelines.

Views and Controllers
Windows, Views, and Controllers. These are the building blocks of your app. You'll learn how views and controllers work together. How to hook up outlets and actions to view controllers and perform code operations.

You'll explore Apple's pre-built view controllers to save you time and work.

Using Table Views (Comprehensive)
Table Views allow you to display and organize data. You'll learn how to insert, delete, arrange/reorder, and search table data. You'll learn how to display images in tables and how to navigate between other tables.

Tracking Keyboard Events
Using Apple's virtual keyboard you'll learn how to interact with keyboard input. Using delegates you'll respond to keyboard events.

A fun app you'll create when learning how to interact with the keyboard is our 'hangman game'. A great opportunity to respond interactively to keyboard input.

Animating Views
Learn how to animate views. You'll make your app interactive for users. Creating an enjoyable and fun experience for your users

Using Gestures
When a user swipes or taps within an app, you'll learn how to respond to these gestures.

Introduction to Core Data
Core Data allows one to manage and store data. You'll learn how to manage, model, add, edit, and delete data.

App Memory Tuning and Debugging
Learn how to debug memory crashes and leaks. You'll be able to profile your app using Clang Static Analyzer.

Transitioning your iPhone app to the iPad
You'll learn about the unique controllers for the iPad. Including the split view controller and popover. You'll learn how to convert an existing iPhone app to the iPad. You'll learn how to create a universal app from scratch to take advantage of the iPhone and iPad in a single app.

ADVANCED & BONUS TOPICS:

Communicate with a webservice (google books api), work with maps, time-zones.

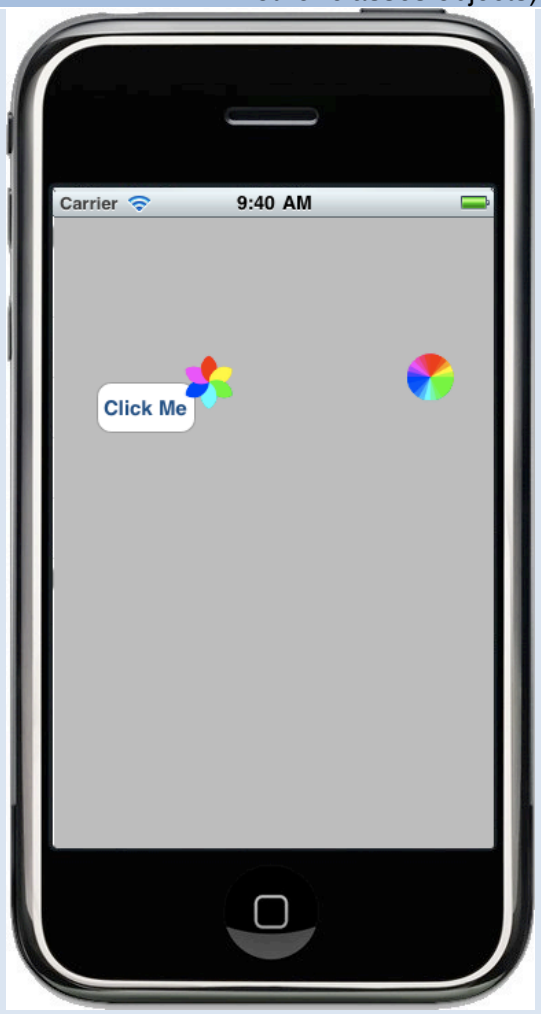
Registering for the free and the paid Apple Developer Programs.

How to run your app on your device.

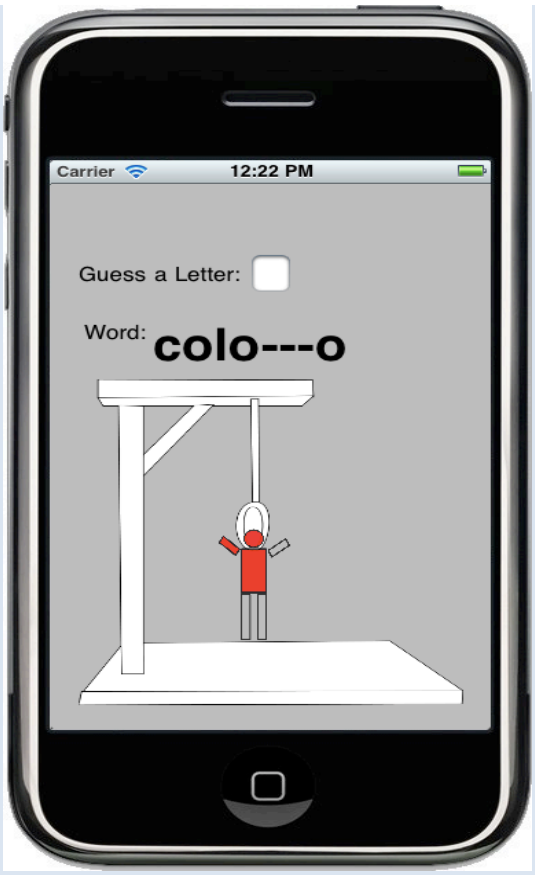
How to publish your app on the Apple Store.

Video Lesson Detail			
Video Name	**Description**	**Length**	**Sample Code: Folder Name**
Pre-Class (Before Class)			
intro	About this iPhone Programming Course	06:06	<none>
xcode_downloadinginstalling	Xcode: Downloading and Installing	02:14	<none>
firstapphelloworld	First Application: Hello World!	07:22	helloworld

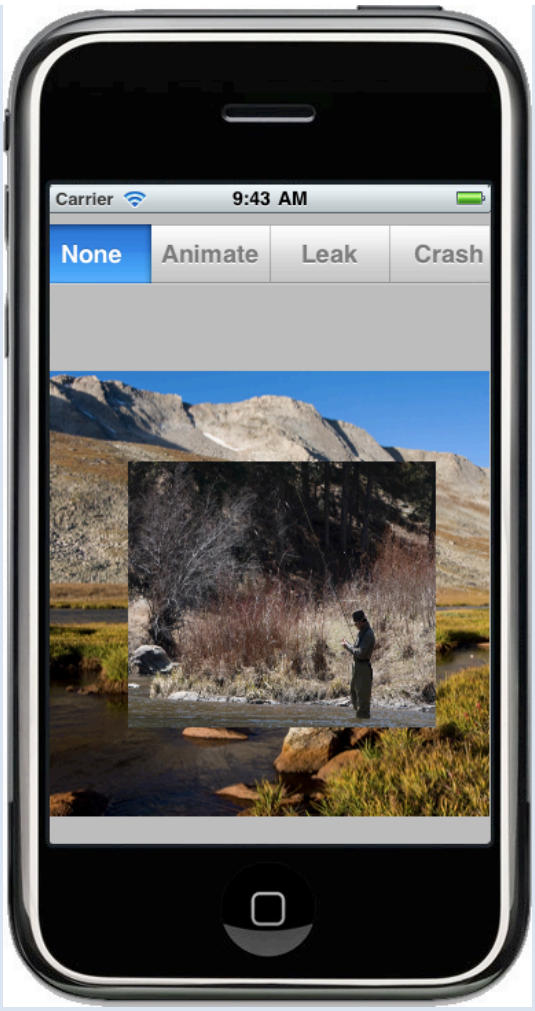
Xcode: Editor tools and tips (Day 1 : Morning Session 8:00 – 9:30 am)			
introducingxcode part1	An Introduction to Xcode: Editor Walkthru	06:02	mygame_final
xcodeaddfiles	Xcode: Adding Files	02:08	Mygame
xcodenavigatecodeeditor	Xcode: Navigating Code Editor	05:35	Mygame
xcodeeditortips	Xcode: Editor Tips	06:02	Mygame
Xcodemanageschemes	Xcode: Managing Schemes	06:53	Mygame
Xcodecodesnippets	Xcode: Code Snippets	05:55	Mygame
Introduction to Objective-C: Selectors and Memory (Day 1 : 10:00 – 11:30 am)			
introobjectivec_s	Introduction to Objective-C:	10:16	introobjectiveC_classes>obj

introobjectivec_selectors	Introduction to Objective-C: Selector	10:16	introobjectiveC_classes>objCIntro_template_finished
introobjectivec_protocol	Introduction to Objective-C: Protocols (Sending messages to other classes/objects)	11:44	introobjectiveC_classes>objCIntro_template_finished
			
introobjectivec_memorydemo	Introduction to Objective-C Memory Management	13:08	MemoryDemo > template
Introduction to Objective C: Classes (Day 1: 12:30 – 2:00 pm)			
introobjectivec_classes	Introduction to Objective-C: Classes	25:17	introobjectiveC_classes>objcintro_template
introobjectivec_classes_defs	Introduction to Objective-C: Classes:Definition	05:57	introobjectiveC_classes>objcintro_template
introobjectivec_classes_id	Introduction to Objective-C: Classes:id	02:45	introobjectiveC_classes>objcintro_template
introobjectivec_classes_variablesco	Introduction to Objective-C: Classes:variable scope	04:19	introobjectiveC_classes>objcintro_template

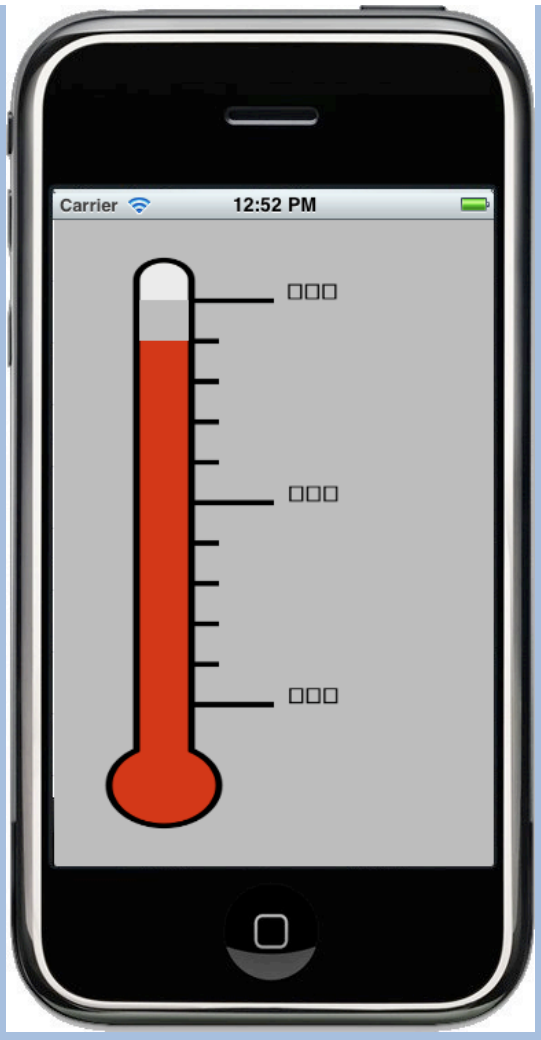
<Break for 15 Minutes>			
introobjectivec_fastenumeration	Introduction to Objective-C: Fast Enumeration (Looping thru collections)	06:57	introobjectiveC_classes>objc intro_template
introobjectivec_classes_properties	Introduction to Objective-C: Classes:Properties	10:05	introobjectiveC_classes>objc intro_template
introobjectivec_classes_polymorphism	Introduction to Objective-C: Classes:polymorphism	13:19	introobjectiveC_classes>objc intro_template
introobjectivec_categories	Introduction to Objective-C: Categories	11:45	introobjectiveC_classes>objc intro_template
introobjectivec_exceptions	Introduction to Objective-C: Exceptions	10:29	introobjectiveC_classes>objc intro_template
Introduction to Objective C: Foundation (Day 1: 2:30 – 4:00 pm)			
introobjectivec_foundation_nsarray	Introduction to Objective-C: Foundation: NSArray	12:39	introObjectiveC_Foundation> NSArray
introobjectivec_foundation_nsdata	Introduction to Objective-C: Foundation: NSData	10:07	introObjectiveC_Foundation> NSData
introobjectivec_foundation_nsdate	Introduction to Objective-C: Foundation: NSDate	10:17	introObjectiveC_Foundation> NSDate
<Break for 15 Minutes>			
introobjectivec_foundation_nsdictionary	Introduction to Objective-C: Foundation: NSDictionary (Sectioned table)	17:10	introObjectiveC_Foundation> NSDictionary
introobjectivec_foundation_nsstring	Introduction to Objective-C: Foundation: NSString (keyboard selectors) Demo: Hangman	27:48	introObjectiveC_Foundation> NSString



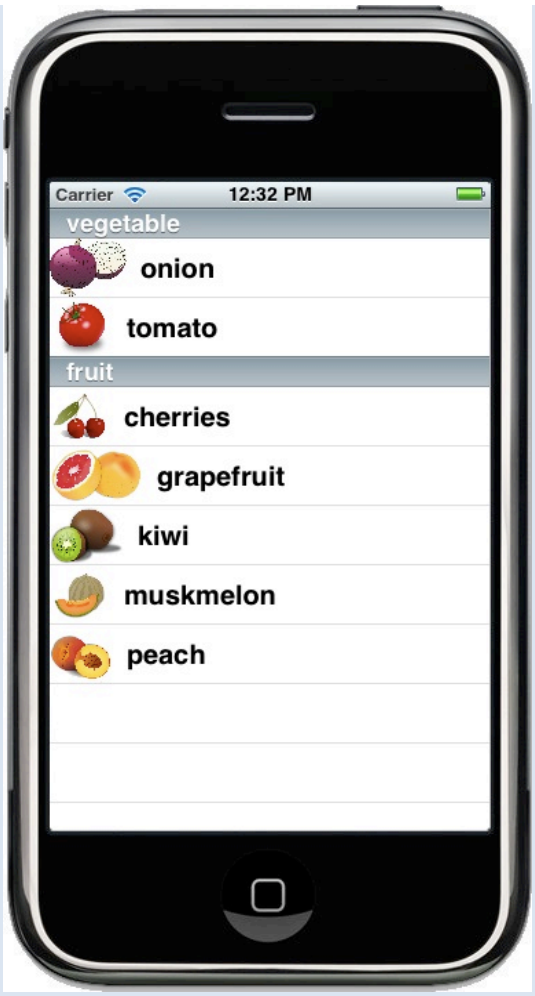
<End of Day 1, Questions & Answers Session> Send questions to: Shawn@LearnApps.org			
Introduction to Objective-C: Foundation (Day 2: 8:00 – 9:30 am)			
introobjectivec_foundation_autorelease	Introduction to Objective-C: Foundation: Autorelease	08:13	introObjectiveC_Foundation> AutoReleaseDemo
Memory Management (Day 2: 8:00 – 9:30am)			
introobjectivec_memorydemo	Introduction to Objective-C Memory Management (REPEATED FOR REVIEW)	13:08	MemoryDemo > template
introobjectivec_memorydemo_clang	Introduction to Objective-C Memory Management: Leaks (Clang)	05:27	MemoryDemo > final
introobjectivec_memorydemo_nszombies	Introduction to Objective-C Memory Management: Crashes (NSZombies)	07:02	MemoryDemo > final



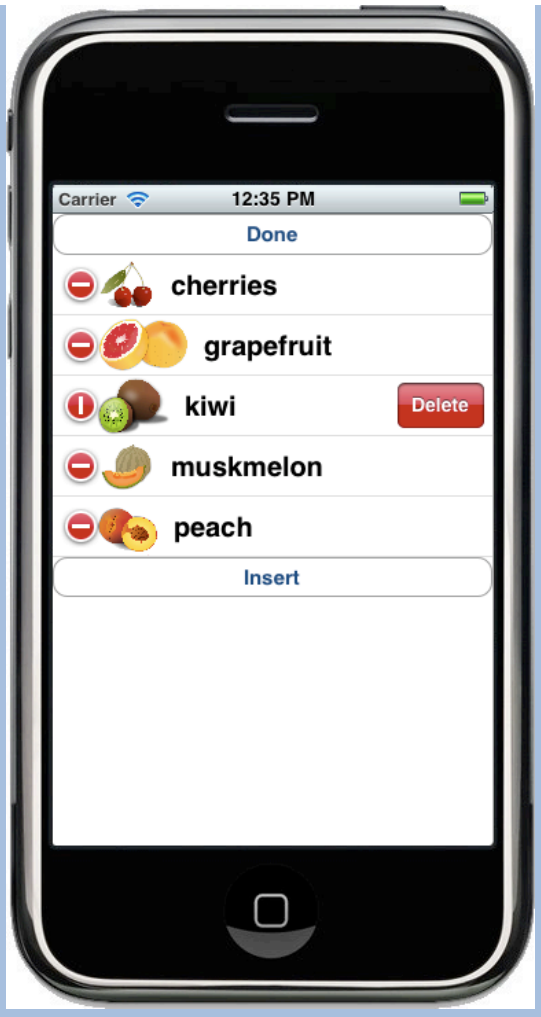
<Break for 15 Minutes>			
Application and View Life Cycle (Day 2: 8:00 – 9:30am)			
Application	Application Life Cycle	08:58	application
Viewhierarchycreation	Views: Hierarchy and Creation (Adding views to iPad's using ~ipad) Thermometer demo	16:52	Viewhierarchycreation > template



Tables (Day 2: 10:00 – 11:30 am)			
tables_minimumtable_mindemo	Tables: Minimum Requirements	09:47	Tables > mindemo > template
tables_navigatingtables	Tables: Navigating between tables	16:06	Tables > navigatingtables > template
tables_sectionedtable	Tables: Sectioned Table	10:32	Tables > sectionedtable > template
<Break for 15 Minutes>			
tables_sectionedtableadvanced	Tables: Advanced Sectioned Table (Images to cell and background image to table)	12:21	Tables > sectionedtableadvanced > template



tables_tablecustomcell	Tables: Custom Cells (Realigning cell image)	12:16	Tables > tablecustomcell > template
tables_tableInsertDelete_Delete	Tables: Remove rows	12:26	Tables > tableinsertdelete > delete > template
Tables (Day 2: 12:30 – 2:00 pm)			
tables_tableInsertDelete_insert	Tables: Adding rows to a table (Modal View Delegates/Protocols)	20:11	Tables > tableinsertdelete > delete > template



tables_tablereorder	Tables: Reordering rows	06:30	Tables > tablereorder > template
tables_tablesearch	Tables: Table Search	09:46	Tables > tablesearch > template
<Break for 15 Minutes>			
Navigation (Day 2: 12:30 – 2:00 pm)			
navigation_navigation	Navigation: Navigation Controller Demo: World Time Clock	30:19	Navigation > navigation controllers > template Notes: For Xcode 4.2, the 'navigation' project template is now called 'master/detail'
<Break for 15 Minutes>			
Navigation (Day 2: 2:30 – 4:00 pm)			
navigation_tabbar	Navigation: Tab bar Demo: reading a web page using Web View	12:24	Navigation> tabbar controllers> template

ipad_newipadapp	iPad: New iPad Application	07:07	Ipad > newipaddapp > template
<Break for 15 Minutes>			
ipad_universalapp	iPad: Universal App (converting iPhone app to iPad app)	10:46	Ipad > univeralapp > template
ipad_popoverapp	iPad: Popover App Demo: How to add a framework and how to use mapkit api (mapping)	18:50	Ipad > popoverapp > template

<End of Day 2, Questions & Answers Session>

Send questions to: Shawn@LearnApps.org

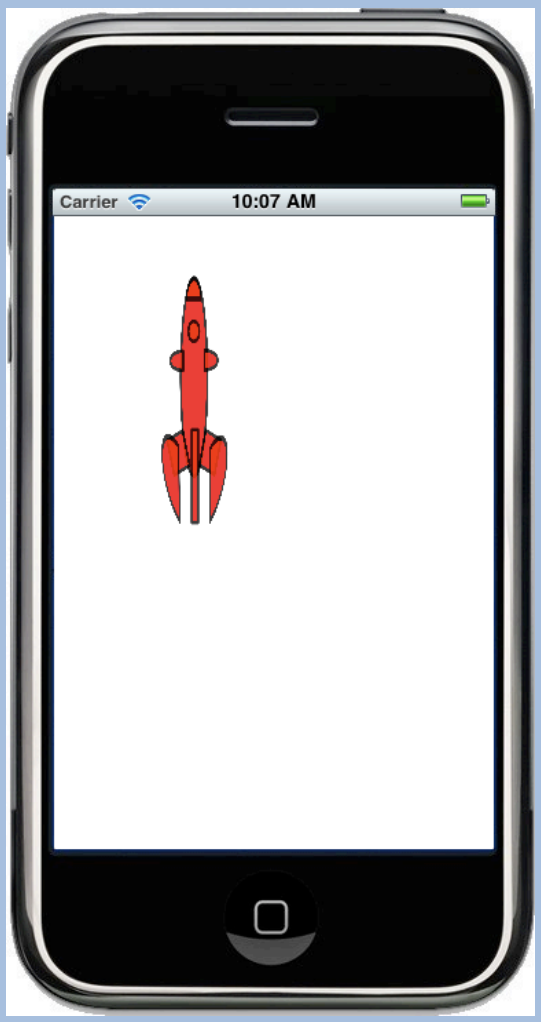
Gestures (Day 3: 8:00 – 9:30 am)

Gestures	Learn About Gestures (Animating views)	11:34	gesture

<Break for 15 Minutes>			
Core Data (Day 3: 8:00 – 9:30 am)			
Coredata	Introduction to Core Data (Google books web service)	30:44	coredata

Human Interface Guidelines and Device Setup (Day 3: 10:00 – 11:30 am)			
applicationIcons	Setting up Application Icons	11:00	applicationicons
devicerestoreos	Restoring Device to a particular O/S	02:50	<none>
<Break for 15 Minutes>			
Registering as an Apple Developer (Day 3: 10:00 – 11:30 am)			
Registereddeveloper	Registered Developer	06:19	<none>
paideveloper	Registering as Paid Developer (allowing you to sell apps on App Store)	07:18	<none>
Run App on Device (Day 3: 10:00 – 11:30 am)			
runappondevice	Running app on device (allowing you to test your app on your ipod or iphone)	19:06	RocketGame
Individual App Project: Start to Finish (Day 3: 12:30 – 2:00 pm)			
finalapp	Create your own app from Start to Finish	25:19	

appstoredistribution	App Store: Preparing and Submitting App	27:55	RocketGame_final\template



<End of Day 3, Questions & Answers Session>

Send questions to: Shawn@LearnApps.org

Resources
Screen Mockups:
http://www.balsamiq.com http://builds.balsamiq.com/b/mockups-web-demo/
Graphics Resources:
http://www.SmashingMagazine.com
http://www.openClipArt.org
http://blog.twg.ca/2009/09/free-iphone-toolbar-icons/
Graphics Editor & Tutorials (Inkscape):
http://inkscape.org/
http://inkscapetutorials.wordpress.com/
http://cocoawithlove.com/search/label/graphics
http://screencasters.heathenx.org/
iPhone Help Forums:
http://www.iphonedevsdk.com/
Apple Developer Website:
http://developer.apple.com
Apple App Store Manager:
http://itunesconnect.apple.com

Screen mockups

It may be useful to create screen drawings of your app idea. This will help firm up the workflow of your screens. It will also serve as a guide to the look and feel of your app.

Balsamiq.com allows free demos of their screen mockup software that works over the web. It has some iphone templates that work well. You may want to do a web search for other screen mockup software. Doing screen mockups can definitely help chart a path for your app layout. It will save considerable time and potentially wasted effort by allowing you to experiment with your potential app screens.

Graphics Resources

Smashingmagazine.com has some great icons and graphics, with many of these being available for free. For iPhone specific toolbar icons, there is http://blog.twg.ca/2009/09/free-iphone-toolbar-icons/ . Please be mindful of any licensing requirements for these icons. You are likely better off just going with a graphics designer for many of your app's icons. It will look more professional and present a custom polish to your app.

Graphics Dimensions

The iPhone screen is 320 x 480 pixels. The app icon that displays on the iTunes store is 512 x 512 pixels. The app icon that is displayed on the iPhone menu (springboard) is 57 x 57 pixels. Toolbar icons are 27x30 and need to be an Alpha channel only graphic. Use the PNG file format for your images.

Open Source Code

Google code (code.google.com) seems to be more popular for iPhone open source code than SourceForge.net. Open Source can be a great resource for seeing how larger projects are constructed. It can also be useful as starting sources for your own projects. Be sure to read and understand the source code's terms of use before incorporating any source into your project.

Help Forums

A good prefix keyword to use when searching for solutions to problems is: 'iphone sdk'. This will usually bring up programmer relevant topics. This can be useful to avoid sifting thru all the non-techie iPhone forums and blog posts. iPhoneDevSdk.com is a particularly helpful blog for programmers on the iPhone stack.

Dedication

This video course is dedicated to my wife Lisa and our three children. Jediah, Hannah, and Victoria made it all worth it. Because of them, I strive to learn, work, and play!