

Android Programming

This Android Training Course will help you build your first working application quick-quick. You'll learn hands-on how to structure your app, design interfaces, create a database, make your app work on various smartphones and tablets, and if you want to pass the international exam AND-401, in Android, the syllabus is covered.

Prerequisites

Before attempting Android Programming on its own, you should have completed at least [Beginner Java](#), if not [Advanced Java](#) or be very familiar with OO and advanced topics in Java or a similar OO language .

Alignment

AND-401 Android Certification. The exam is excluded, but we cover and support you in full if you want to sit for the international exam.

Course Contents

DAY 1

Welcome to Android

- The Android platform dissected
- Your development environment
- Install Java
- Build a basic app
- Activities and layouts from 50,000 feet
- Building a basic app (continued)
- Building a basic app (continued)
- You've just created your first Android app
- Android Studio creates a complete folder structure for

- you
- Useful files in your project
 - Edit code with the Android Studio editors
 - Run the app in the Android emulator
 - Creating an Android Virtual Device
 - Run the app in the emulator
 - You can watch progress in the console
 - What just happened?
 - Refining the app
 - What's in the layout?
 - `activity_main.xml` has two elements
 - The layout file contains a reference to a string, not the string itself
 - Let's look in the `strings.xml` file

Interactive Apps

- You're going to build a Beer Adviser app
- Create the project
- We've created a default activity and layout
- Adding components with the design editor
- `activity_find_beer.xml` has a new button
- Changes to the XML...
- ...are reflected in the design editor
- Use string resources rather than hardcoding the text
- Change the layout to use the string resources
- Add values to the spinner
- Get the spinner to reference a string-array
- We need to make the button do something
- Make the button call a method
- What activity code looks like
- Add an `onClickFindBeer()` method to the activity
- `onClickFindBeer()` needs to do something
- Once you have a View, you can access its methods
- Update the activity code
- The first version of the activity
- Building the custom Java class

- Enhance the activity to call the custom Java class so that we can get REAL advice
- Activity code version 2
- What happens when you run the code

Multiple Activities and Intents

- Apps can contain more than one activity
- Here's the app structure
- Create the project
- Create the second activity and layout
- Welcome to the Android manifest file
- Use an intent to start the second activity
- What happens when you run the app
- Pass text to a second activity
- Update the text view properties
- putExtra() puts extra information in an intent
- Update the CreateMessageActivity code
- Get ReceiveMessageActivity to use the information in the intent
- What happens when the user clicks the Send Message button
- How Android apps work
- What happens when the code runs
- How Android uses the intent filter
- You need to run your app on a REAL device
- Change the code to create a chooser

DAY 2

The Activity LifeCycle

- How do activities really work?
- The Stopwatch app
- The stopwatch layout code
- Add code for the buttons
- The runTimer() method
- Handlers allow you to schedule code

- The full runTimer() code
- The full StopwatchActivity code
- Rotating the screen changes the device configuration
- From birth to death: the states of an activity
- The activity lifecycle: from create to destroy
- How do we deal with configuration changes?
- What happens when you run the app
- There's more to an activity's life than create and destroy
- The activity lifecycle: the visible lifetime
- The updated StopwathActivity code
- What happens when you run the app
- But what if an app is only partially visible?
- The activity lifecycle: the foreground lifetime
- Stop the stopwatch if the activity's paused
- The complete activity code
- Your handy guide to the lifecycle methods

The User Interface

- Three key layouts: relative, linear, and grid
- Positioning views relative to the parent layout
- Positioning views relative to other views
- Attributes for positioning views relative to other views
- RelativeLayout: a summary
- LinearLayout displays views in a single row or column
- Let's change up a basic linear layout
- Adding weight to one view
- Adding weight to multiple views
- Using the android:gravity attribute: a list of values
- More values you can use with the android:layout
- gravity attribute
- The full linear layout code
- LinearLayout: a summary
- GridLayout displays views in a grid
- Adding views to the grid layout
- Let's create a new grid layout

- Row 0: add views to specific rows and columns
- Row 1: make a view span multiple columns
- Row 2: make a view span multiple columns
- The full code for the grid layout
- GridLayout: a summary
- Layouts and GUI components have a lot in common
- Playing with views

List Views and Adapters

- Every app starts with ideas
- Categorize your ideas: top-level, category, and detail/edit activities
- Navigating through the activities
- Use ListView to navigate to data
- We're going to build the Starbuzz app
- The drink detail activity
- The Starbuzz app structure
- The top-level layout contains an image and a list
- The full top-level layout code
- Get ListView to respond to clicks with a Listener
- The full TopLevelActivity code
- How to create a list activity
- Connect list views to arrays with an array adapter
- Add the array adapter to DrinkCategoryActivity
- What happens when you run the code
- How we handled clicks in TopLevelActivity
- The full DrinkCategoryActivity code
- A detail activity displays data for a single record
- Update the views with the data
- The DrinkActivity code

DAY 3

Fragments

- The Workout app structure
- The Workout class

- How to add a fragment to your project
- What fragment code looks like
- Activity states revisited
- The fragment lifecycle
- Your fragment inherits the lifecycle methods
- How to create a list fragment
- The updated `WorkoutListFragment` code
- Wiring up the list to the detail
- Using fragment transaction
- The updated `MainActivity` code
- The `WorkoutDetailFragment` code
- The phone and tablet app structures
- The different folder options
- The `MainActivity` phone layout
- The full `DetailActivity` code
- The revised `MainActivity` code

Nested Fragments

- Creating nested fragments
- The `StopwatchFragment` code
- The `StopwatchFragment` layout
- `getFragmentManager()` creates transactions at the activity level
- Nested fragments need nested transactions
- The full `WorkoutDetailFragment` code
- Why does the app crash if you press a button?
- the `StopwatchFragment` layout code
- Make the fragment implement `OnClickListener`
- Attach the `OnClickListener` to the buttons
- The `StopwatchFragment` code
- The `WorkoutDetailFragment` code

DAY 4

Action Bars

- Great apps have a clear structure

- Different types of navigation
- Let's start with the action bar
- The Android support libraries
- Your project may include support libraries
- We'll get the app to use up to date themes
- Apply a theme in `AndroidManifest.xml`
- Define styles in style resource files
- Set the default theme in `styles.xml`
- What happens when you run the app
- Adding action items to the action bar
- The menu resource file
- The menu `showAsAction` attribute
- Add a new action item
- Create `OrderActivity`
- Start `OrderActivity` with the Create Order action item
- The full `MainActivity.java` code
- Sharing content on the action bar
- Specify the content with an intent
- The full `MainActivity.java` code
- Enabling Up navigation
- Setting an activity's parent
- Adding the Up button

Navigation Drawers

- The Pizza app revisited
- Navigation drawers deconstructed
- The Pizza app structure
- Create `TopFragment`
- Create `PizzaFragment`
- Create `PastaFragment`
- Create `StoresFragment`
- Add the `DrawerLayout`
- The full code for `activity_main.xml`
- Initialize the drawer's list
- Changing the action bar title
- Closing the navigation drawer

- The updated MainActivity.java code
- Using an ActionBarDrawerToggle
- Modifying action bar items at runtime
- The updated MainActivity.java code
- Enable the drawer to open and close
- Syncing the ActionBarDrawerToggle state
- The updated MainActivity.java code
- Dealing with configuration changes
- Reacting to changes on the back stack
- Adding tags to fragments

SQLite Databases

- Back to Starbuzz
- Android uses SQLite databases to persist data
- Android comes with SQLite classes
- The current Starbuzz app structure
- The SQLite helper manages your database
- The SQLite helper
- Create the SQLite helper
- Inside a SQLite database
- You create tables using Structured Query Language (SQL)
- Insert data using the insert() method
- Update records with the update() method
- Multiple conditions
- The StarbuzzDatabaseHelper code
- What the SQLite helper code does
- What if you need to change the database?
- SQLite databases have a version number
- Upgrading the database: an overview
- How the SQLite helper makes decisions
- Upgrade your database with onUpgrade()
- Downgrade your database with onDowngrade()
- Let's upgrade the database
- Upgrading an existing database
- Renaming tables
- The full SQLite helper code

- The SQLite helper code (continued)
- What happens when the code runs

DAY 5

Cursors and Asynch Tasks

- The current DrinkActivity code
- Specifying table and columns
- Applying multiple conditions to your query
- Using SQL functions in queries
- Navigating cursors
- The DrinkActivity code
- Add favorites to DrinkActivity
- The DrinkActivity code
- The new top-level activity code
- The revised TopLevelActivity.java code
- The onPreExecute() method
- The doInBackground() method
- The onProgressUpdate() method
- The onPostExecute() method
- The AsyncTask class
- The DrinkActivity.java code

Services

- The started service app
- The IntentService from 50,000 feet
- How to log messages
- The full DelayedMessageService code
- The full DelayedMessageService.java code
- How you use the notification service
- Getting your notification to start an activity
- Send the notification using the notification service
- The full code for DelayedMessageService.java
- The steps needed to create the OdometerService
- Define the Binder
- The Service class has four key methods

- Add the LocationListener to the service
- Registering the LocationListener
- The full OdometerService.java code
- Update AndroidManifest.xml
- Update MainActivity's layout
- Create a ServiceConnection
- Bind to the service when the activity starts
- Display the distance traveled
- The full MainActivity.java code

Material Design

- Welcome to Material Design
- The Pizza app structure
- Create the CardView
- The full card_captioned_image.xml code
- Create the basic adapter
- Define the adapter's ViewHolder
- Create the ViewHolders
- Each card view displays an image and a caption
- Add the data to the card views
- The full code for CaptionedImagesAdapter.java
- Create the recycler view
- Add the RecyclerView to the layout
- The PizzaMaterialFragment.java code
- A RecyclerView uses a layout manager to arrange its views
- Specifying the layout manager
- The full PizzaMaterialFragment.java code
- Get MainActivity to use the new PizzaMaterialFragment
- What happens when the code runs
- Create PizzaDetailActivity
- What PizzaDetailActivity.java needs to do
- The code for PizzaDetailActivity.java
- Getting a RecyclerView to respond to clicks
- Add the interface to the adapter
- Implement the listener in PizzaMaterialFragment.java

- Bring the content forward
- The full code for fragment_top.xml
- The full code for TopFragment.java

Duration and pricing

Price [Group A](#)

Certificate

1. Upon completion of this course we will issue you with attendance certificate to certify your attendance.
2. You may sit for our competency assessment test and on passing you will obtain our competency certificate.
3. Our competency assessment can be booked and taken by someone who has not attended the course at a cost of R950.

Bookings

You can download the course registration form on our home page or by clicking [here](#)

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions

Please [email us](#)

Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page.

HTML5 / CSS3 Responsive Web Design

This HTML5 CSS3 Training course will teach you in detail how to design responsive websites that will run on any browser and device. A must – do for any developer!

Prerequisites

You should be familiar with [HTML and CSS](#) before attempting this course Responsive Web Design with HTML and CSS3

Intended Audience

For beginning to intermediate level computer users with a good comprehension of HTML and CSS already.

Course Material

Included in the course price.

Course Contents

Day 1

THE ESSENTIALS OF RESPONSIVE WEB DESIGN

- Beginning our quest
- Defining responsive web design
- Setting browser support levels
- Our first responsive example
- The shortcomings of our example

MEDIA QUERIES – SUPPORTING DIFFERING VIEWPORTS

- Why media queries are needed for a responsive web design
- Media query syntax
- Combining media queries
- Using media queries to alter a design
- Considerations for organizing and authoring media

queries

- Combine media queries or write them where it suits?
- The viewport meta tag
- Media Queries Level 4

Day 2

FLUID LAYOUTS AND RESPONSIVE IMAGES

- Converting a fixed pixel design to a fluid proportional layout
- Introducing Flexbox
- Getting Flexy
- Responsive images

HTML5 FOR RESPONSIVE WEB DESIGNS

- HTML5 markup – understood by all modern browser
- Starting an HTML5 page the right way
- Easy-going HTML5
- New semantic elements in HTML5
- HTML5 text-level semantics
- Obsolete HTML features
- Putting HTML5 elements to use
- WCAG and WAI-ARIA for more accessible web applications
- Embedding media in HTML5
- Responsive HTML5 video and iFrames
- A note about ‘offline first’

Day 3

CSS3 – SELECTORS, TYPOGRAPHY, COLOR MODES, AND NEW FEATURES

- Anatomy of a CSS rule
- Quick and useful CSS tricks
- Word wrapping
- Facilitating feature forks in CSS
- New CSS3 selectors and how to use them
- CSS3 structural pseudo-classes

- CSS custom properties and variables
- CSS calc
- CSS Level 4 selectors
- Web typography
- New CSS3 color formats and alpha transparency

STUNNING AESTHETICS WITH CSS3

- Text shadows with CSS3
- Box shadows
- Background gradients
- Repeating gradients
- Background gradient patterns
- Multiple background images
- High-resolution background images
- CSS filters
- A warning on CSS performance

Day 4

STUNNING AESTHETICS WITH CSS3

- Text shadows with CSS3
- Box shadows
- Background gradients
- Repeating gradients
- Background gradient patterns
- Multiple background images
- High-resolution background images
- CSS filters
- A warning on CSS performance

USING SVGS FOR RESOLUTION INDEPENDENCE

- A brief history of SVG
- The graphic that is a document
- Creating SVGs with popular image editing packages and services
- Inserting SVGs into your web pages
- Inserting an SVG inline
- What you can do with each SVG insertion method (inline,

- object, background-image, and img)
- Extra SVG capabilities and oddities
- Animating SVG with JavaScript
- Optimising SVGs
- Using SVGs as filters
- A note on media queries inside SVGs

Day 5

: CONQUER FORMS WITH HTML5 AND CSS3

- HTML5 forms
- Understanding the component parts of HTML5 forms
- HTML5 input types
- How to polyfill non-supporting browsers
- Styling HTML5 forms with CSS3

: APPROACHING A RESPONSIVE WEB DESIGN

- Get designs in the browser as soon as possible
- View and use the design on real devices
- Embracing progressive enhancement
- Defining a browser support matrix
- Tiering the user experience
- Linking CSS breakpoints to JavaScript
- Avoid CSS frameworks in production
- Coding pragmatic solutions
- Use the simplest code possible
- Hiding, showing, and loading content across viewports
- Validators and linting tools
- Performance
- The next big things

Duration and pricing

Price Group A

Certificate

Upon completion of this course we will issue you with attendance certificate to certify your attendance and / or

completion of the prescribed minimum examples.
You may sit for our competency assessment test and on passing you will obtain our competency certificate.
Our competency assessment can be booked and taken by someone who has not attended the course at a cost of R950.

Bookings

You can download the course registration form on our home page or by clicking [here](#)

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions

Please [email us](#)

Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page.

Ionic (for Android & iOS)

This Ionic training course will teach you how to create hybrid mobile applications by combining the capabilities of Ionic, Cordova, and AngularJS. Reduce the time to market your application using Ionic, that helps in rapid application development.

Prerequisites

[HTML5 / CSS3](#), [Angular](#) recommended. There is time for a small re-cap of Angular at the beginning of the course, but not

enough time to master it.

Course Contents:

Ionic – Powered by Angular

- Understanding the separation of concerns
- AngularJS components
- AngularJS directives
- AngularJS services
- AngularJS resources

Welcome to Ionic

- Mobile Hybrid Architecture
- What is Apache Cordova?
- What is Ionic?
- Software setup
- Install Node.js
- Install Git
- Install Bower
- Install Gulp
- Install Sublime Text
- Install Cordova and Ionic CLI
- The platform guide
- Hello Ionic
- The browser developer tools setup
- Google Chrome
- Mozilla Firefox
- The Ionic project structure
- The config.xml file
- The www folder
- Scaffolding the tabs template
- Scaffolding the side menu template
- generator-ionic
- Installing generator-ionic

Ionic CSS Components and Navigation

- Ionic CSS components
- The Ionic grid system
- The page structure
- Buttons
- Lists
- Cards
- Ionicons
- Form elements
- Integrating Ionic CSS components with AngularJS
- The Ionic router
- A simple two-page app

Ionic and SCSS

- What is Sass?
- Setting up SCSS in our Ionic project
- The manual setup
- The Ionic CLI task
- Working with Ionic SCSS
- Basic swatch
- Understanding the Ionic SCSS setup
- Using variables and mixins
- The SCSS workflow
- Building a swatch

Ionic Directives and Services

- Ionic directives and services
- The Ionic Platform service
- registerBackButtonAction
- The on method
- Headers and footers
- Content
- ion-content
- ion-scroll
- ion-refresher
- ion-infinite-scroll
- \$ionicScrollDelegate

- Navigation
- ion-view
- Ionic view events
- ion-nav-bar
- ion-nav-buttons
- \$ionicNavBarDelegate
- \$ionicHistory
- Tabs and side menu
- Ionic loading
- The Action Sheet service
- Popover and Popup services
- \$ionicPopup
- The ion-list and ion-item directives
- Gesture directives and services
- Utilities

Building a Bookstore App

- An introduction to the Bookstore application
- The Bookstore architecture
- The server architecture
- The server-side API documentation
- The client architecture
- Code on GitHub
- A Bookstore demo
- The development flow
- Setting up the server
- Building the application
- Step 1 – Scaffolding the side menu template
- Step 2 – Refactoring the template
- Refactoring the menu
- Refactoring the module name
- Adding a run method and modifying routes
- Refactoring templates
- Step 3 – Building authentication, localStorage, and the REST API factory
- The Ionic loading factory

- The localStorage factory
- The Authentication factory
- The REST API factory

Creating controllers for each route and integrating with the factory

- The application controller
- The browse controller
- The book controller
- The cart controller
- The purchase controller
- Step 5 – Creating templates and integrating with the controller data
- The Login template
- The Browse template
- The Book template
- The Cart template
- The Purchase template

Cordova and ngCordova

- Setting up a platform-specific SDK
- The Android setup
- The iOS setup
- Testing the setup
- Testing for Android
- Testing for iOS
- Getting started with Cordova plugins
- The Ionic plugin API
- Add a plugin
- Remove a plugin
- List added plugins
- Search plugins
- The Cordova whitelist plugin
- ngCordova
- Setting up ngCordova
- Legend

- \$cordovaToast
- \$cordovaDialogs
- \$cordovaFlashlight
- \$cordovaLocalNotification
- \$cordovaGeolocation

Building a Messaging App

- The Ionic Chat app
- Firebase
- Setting up a Firebase account
- AngularFire
- The application architecture
- Authentication
- The application flow
- Previewing the app
- Data structure
- Cordova plugins
- Code on GitHub
- Developing the application
- Scaffolding and setting up the app
- Installing the required Cordova plugins
- Getting the Google API key
- Setting up routes and route authentication
- Setting up services/factories
- Setting up a map directive
- Setting up controllers
- Setting up templates
- Setting up SCSS
- Testing the application

Releasing the Ionic App

- Preparing the app for distribution
- Setting up icons and splash screens
- Updating the config.xml file
- The PhoneGap service
- Generating installers using the Cordova CLI

- Android installer
 - iOS installer
 - The Ionic package
 - Uploading the project to Ionic cloud
 - Generating the required keys
-

Duration and pricing

In [Pricing Group A](#)

Certificate

Read [here about our certificates](#)

Bookings

You can download the course registration form on our home page or by clicking [here](#)

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions

Please [email us](#)

Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page.

Mobile Web Development

Professional Mobile Web Development using mainly HTML5, CSS3 and JavaScript. Learn how to build lean and fast performing web mobile apps like native apps

Prerequisites

Before attempting Mobile Web Development Programming on its own, you should have previous web design and development experience and knowledge of HTML, CSS and basic Javascript.

Contents:

Day 1:

Responsive Web Design

- Get on the mobile bandwagon
- The recipe for Responsive Web Design
- An example of a responsively designed site
- CSS media queries
- Analyze the current CSS
- Steps to creating the mobile-specific CS

Responsible Responsiveness

- Mobile-first Responsive Web Design
- Can I get a proxy to set up my proxy?
- What to do when things aren't blazing fast
- Find the drags on page speed
- It looks mobile friendly, but it isn't
- What is progressive enhancement?
- Mobile-first media queries
- Add the map back using JavaScript
- Move iframe attributes to CSS equivalents
- Breakpoints to the rescue

Day 2:

A Separate Mobile Website

- Creature Comforts has agents in the field
- Getting to know user agents
- Straight talk: Most major sites have a separate mobile website
- When what you really want to do is (re-)direct
- Make a mobile mockup
- Not all phones are smartphones...not by a sight
- Let's keep it basic: Meet XHTML-MP
- Access keys in action
- Mobile-savvy CSS

What devices should we support?

- How do you know where to draw the line?
- Step away from the keyboard for a second
- Things you don't support vs. those you can't support
- Ask questions about your project
- Ingredients for your magic mobile potion
- Draw from your cupboard of tools and data

Day 3:

Device Databases and Classes

- A panic button for freaked-out students
- The button is for mobile phones only
- WURFL and its capabilities
- WURFL: Clever API code
- Steps for building our explore page
- Use WURFL to help differentiate content
- Make the page a bit smarter with WURFL
- The panic button: For phones only
- Expanding a lucrative part of AcedIt!'s business
- Get acquainted with the matching function
- Make something actually happen with device classes
- We need a bigger safety net

Build a Mobile Web App Using a Framework

- HTML5 is a specific thing...
- How “traditional” websites typically behave
- A Games Unlimited mobile HTML5 web app
- The master plan for phase 1 of the Game
- Why use mobile web app frameworks?
- Our choice for the Game: jQuery Mobile
- Build a basic page with jQuery Mobile
- Link to multiple pages with jQuery Mobile
- Make the Game feel more applike: to-dos
- Time to make that tartan-building form
- Build an HTML5 form
- Give jQuery Mobile hints about the fields

Day 4:

Mobile Web Apps in the Real World

- Mobile apps in the real world
- Make a better form
- A widget to manage the list of colors and sizes
- The two sides of generate.php
- Offline is important
- A basic recipe to create a cache manifest
- Dev tools to the rescue
- How to ask W3C-compliant browsers where they are
- Let’s integrate geolocation

Build Hybrid Mobile Apps

- How do hybrid apps work?
- Bridge the web-native gap with PhoneGap
- Get acquainted with PhoneGap Build
- Keep track of discovered tartans
- Anatomy of the Tartan Hunt project
- What makes localStorage so special?
- Use a function to show which tartans are found
- Rope in PhoneGap to take pictures

- Now we're ready for the mediaCapture API

Day 5:

How to Be Future Friendly

- Time to dispel our collective illusions of control
- A future-friendly manifesto
- There are no silver bullets
- App today, web page tomorrow
- Remove PhoneGap references

Project

- Develop a mobile app from start to finish

Duration and pricing

Pricing [Group A](#)

Certificate

Please read about [our certificates.](#)

Bookings

You can download the course registration form on our home page or by clicking [here](#)

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions

Please [email us](#)

Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page