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 ListViews 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 ListViews 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 lavel
- 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 doInBackgound() 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.