

TA-007: Android Programming

0321804333

Android was created by the Open Handset Alliance which is led by Google. The early feedback on developing applications for the Android platform was mixed. Issues cited include bugs, lack of documentation, inadequate QA infrastructure, and no public issue-tracking system. (Google announced an issue tracker on January 18, 2008.) In December 2007, MergeLab mobile startup founder Adam MacBeth stated, "Functionality is not there, is poorly documented or just doesn't work... It's clearly not ready for prime time." Despite this, Android-targeted applications began to appear the week after the platform was announced. The first publicly available application was the Snake game.

Android software development is the process by which new applications are created for the Android operating system. Applications are usually developed in Java programming language using the Android Software Development Kit (SDK), but other development environments are also available.

As of July 2013, more than one million applications have been developed for Android, with over 25 billion downloads. A June 2011 research indicated that over 67% of mobile developers used the platform, at the time of publication. In Q2 2012, around 105 million units of Android smartphones were shipped which acquires a total share of 68% in overall smartphones sale till Q2 2012.

Content

Introduction:

Mobile application development is the process by which application software is developed for low-power handheld devices, such as personal digital assistants, enterprise digital assistants or mobile phones. These applications can be pre-installed on phones during manufacturing, downloaded by customers from various mobile software distribution platforms, or delivered as web applications using server-side or client-side processing (e.g. JavaScript) to provide an "application-like" experience within a Web browser. Application software developers also have to consider a lengthy array of screen sizes, hardware specifications and configurations because of intense competition in mobile software and changes within each of the platforms.

Duration:

40 hrs.

Course Content:

1. Your First Android Application

- App Basics
- Creating an Android Project
- Navigating in Eclipse
- Laying Out the User Interface
- The view hierarchy

- Widget attributes
- Creating string resources
- Previewing the layout
- From Layout XML to View Objects
- Resources and resource IDs
- Wiring Up Widgets
- Organizing imports
- Getting references to widgets
- Setting listeners
- Running on the Emulator
- For the More Curious: Android Build Process
- Android build tools

2. Android and Model-View-Controller

- Creating a New Class
- Generating getters and setters
- Model-View-Controller and Android
- Benefits of MVC
- Updating the View Layer
- Updating the Controller Layer
- Running on a Device
- Connecting your device
- Configuring your device for development
- Adding an Icon
- Adding resources to a project
- Referencing resources in XML
- Challenges
- Challenge: Add a Listener to the TextView
- Challenge: Add a Previous Button
- Challenge: From Button to ImageButton

3. The Activity Lifecycle

- Logging the Activity Lifecycle
- Making log messages
- Using LogCat
- Rotation and the Activity Lifecycle
- Device configurations and alternative resources
- Saving Data Across Rotation
- Overriding onSaveInstanceState(Bundle)
- The Activity Lifecycle, Revisited
- For the More Curious: Testing onSaveInstanceState(Bundle)
- For the More Curious: Logging Levels and Methods

4. Debugging Android Apps

- The DDMS Perspective
- Exceptions and Stack Traces
- Diagnosing misbehaviors
- Logging stack traces
- Setting breakpoints
- Using exception breakpoints
- File Explorer
- Android-Specific Debugging
- Using Android Lint
- Issues with the R class

5. Your Second Activity

- Setting Up a Second Activity
- Creating a new layout
- Creating a new activity subclass
- Declaring activities in the manifest
- Adding a Cheat button to QuizActivity
- Starting an Activity
- Communicating with intents
- Passing Data Between Activities
- Using intent extras
- Getting a result back from a child activity
- How Android Sees Your Activities
- Challenge

6. Android SDK Versions and Compatibility

- Android SDK Versions
- Compatibility and Android Programming
- Honeycomb was big
- Minimum SDK version
- Target SDK version
- Build SDK version
- Adding code from later APIs safely
- Using the Android Developer Documentation
- Challenge: Reporting the Build Version

7. UI Fragments and the Fragment Manager

- The Need for UI Flexibility
- Introducing Fragments
- Starting CriminalIntent
- Creating a new project
- Fragments and the support library
- Creating the Crime class
- Hosting a UI Fragment

- The fragment lifecycle
- Two approaches to hosting
- Defining a container view
- Creating a UI Fragment
- Defining CrimeFragment's layout
- Creating the CrimeFragment class
- Adding a UI Fragment to the FragmentManager
- Fragment transactions
- The FragmentManager and the fragment lifecycle
- The Reason All Our Activities Will Use Fragments
- For the More Curious: Developing for Honeycomb, ICS, Jelly Bean, and Beyond

8. Creating User Interfaces with Layouts and Widgets

- Upgrading Crime
- Updating the Layout
- Wiring Widgets
- More on XML Layout Attributes
- Styles, themes, and theme attributes
- Screen pixel densities and dp and sp
- Android's design guidelines
- Layout parameters
- Margins vs. padding
- Using the Graphical Layout Tool
- Adding a new widget
- Editing attributes in properties
- Reorganizing widgets in the outline view
- Updating child layout parameters
- How android:layout_weight works
- Summary of graphical layout tool
- Widget IDs and multiple layouts
- Challenge: Formatting the Date

9. Displaying Lists with ListFragment

- Updating CriminalIntent's Model Layer
- Singletons and centralized data storage
- Creating a ListFragment
- An Abstract Activity for Hosting a Fragment
- A generic fragment-hosting layout
- An abstract Activity class
- ListFragment, ListView, and ArrayAdapter
- Creating an ArrayAdapter<T>
- Responding to list item clicks
- Customizing List Items
- Creating the list item layout

- Creating an adapter subclass

10. Using Fragment Arguments

- Starting an Activity from a Fragment
- Putting an extra
- Retrieving an extra
- Updating CrimeFragment's view with Crime data
- The downside to direct retrieval
- Fragment Arguments
- Attaching arguments to a fragment
- Retrieving arguments
- Reloading the List
- Getting Results with Fragments

11. Using ViewPager

- Creating CrimePagerActivity
- Laying out views in code
- ViewPager and PagerAdapter
- Integrating CrimePagerActivity
- FragmentStatePagerAdapter vs. FragmentPagerAdapter
- For the More Curious: How ViewPager Really Works

12. Dialogs

- Creating a DialogFragment
- Showing a DialogFragment
- Setting a dialog's contents
- Passing Data Between Two Fragments
- Passing data to DatePickerFragment
- Returning data to CrimeFragment
- Challenge: More Dialogs

13. Audio Playback Using MediaPlayer

- Adding Resources
- Defining the layout for HelloMoonFragment
- Manually resetting the app theme
- Creating HelloMoonFragment
- Using a Layout Fragment
- Audio Playback
- Wiring buttons to play and stop
- Challenge: Pausing Audio Playback
- For the More Curious: Playing Video
- Challenge: Playing Video in HelloMoon

14. Retained Fragments

- Retaining a Fragment
- Rotation and Retained Fragments
- Retained Fragments: As Nice as All That?
- Rotation Handling and onSaveInstanceState(Bundle)
- For the More Curious: Rotation Before Fragments

15. Localization

- Localizing Resources
- Default resources
- Configuration Qualifiers
- Prioritizing alternative resources
- Multiple qualifiers
- Finding the best-matching resources
- More Resource Rules and Regulations
- Resource naming
- Resource directory structure
- Testing Alternative Resources

16. The Action Bar

- Options Menus
- Defining an options menu in XML
- Creating the options menu
- Responding to options menu selections
- Enabling Ancestral Navigation
- Enabling the app icon
- Responding to the Up button
- An Alternative Menu Item
- Creating an alternative menu file
- Toggling the menu item title
- “Just one more thing...”
- Challenge: An Empty View for the List

17. Saving and Loading Local Files

- Saving and Loading Data in CriminallyIntent
- Saving crime data to a JSON file
- Loading crimes from the filesystem
- Challenge: Use External Storage
- For the More Curious: The Android Filesystem and Java I/O
- Accessing files and directories

18. Context Menus and Contextual Action Mode

- Defining a Context Menu Resource
- Implementing a Floating Context Menu
- Creating the context menu

- Registering for the context menu
- Responding to an action
- Implementing Contextual Action Mode
- Enabling multiple selection
- Action mode callbacks in a list view
- Changing activated item backgrounds
- Implementing contextual action mode in other views
- Compatibility: Fallback or Duplicate?
- Challenge: Deleting from CrimeFragment
- For the More Curious: ActionBarSherlock
- Challenge: Using ActionBarSherlock
- Basic ABS integration in CriminalIntent
- More advanced integration
- Even more advanced integration

19. Camera I: Viewfinder

- Creating the Fragment Layout
- Creating CrimeCameraFragment
- Creating CrimeCameraActivity
- Adding activity and camera permissions to the manifest
- Using the Camera API
- Opening and releasing the camera
- SurfaceView, SurfaceHolder, and Surface
- Determining preview size
- Starting CrimeCameraActivity from CrimeFragment
- For the More Curious: Running Activities from the Command Line

20. Camera II: Taking Pictures and Handling Images

- Taking a Picture
- Implementing Camera callbacks
- Setting the picture size
- Passing Data Back to CrimeFragment
- Starting CrimeCameraActivity for a result
- Setting a result in CrimeCameraFragment
- Retrieving filename in CrimeFragment
- Updating the Model Layer
- Adding a Photo class
- Giving Crime a photo property
- Setting the photo property
- Updating CrimeFragment's View
- Adding an ImageView
- Image handling
- Showing Larger Image in a DialogFragment
- Challenge: Crime Image Orientation

- Challenge: Deleting Photos
- For the More Curious: Deprecation in Android

21. Implicit Intents

- Adding Buttons
- Adding a Suspect to the Model Layer
- Using a Format String
- Using Implicit Intents
- Parts of an implicit intent
- Sending a crime report
- Asking Android for a contact
- Checking for responding activities
- Challenge: Another Implicit Intent

22. Two-Pane Master-Detail Interfaces

- Adding Layout Flexibility
- Modifying SingleFragmentActivity
- Creating a layout with two fragment containers
- Using an alias resource
- Activity: Fragment Boss
- Fragment callback interfaces
- For the More Curious: More on Determining Device Size

23. More About Intents and Tasks

- Setting Up NerdLauncher
- Resolving an Implicit Intent
- Creating Explicit Intents at Runtime
- Tasks and the Back Stack
- Using NerdLauncher as a Home Screen
- Challenge: Icons, Reordering Tasks
- For the More Curious: Processes vs. Tasks

24. Styles And Includes

- Setting Up the RemoteControl Project
- Setting up RemoteControlActivity
- Setting up RemoteControlFragment
- Cleaning Up with Styles
- Finishing the Layout
- For the More Curious: include and merge
- Challenge: Style Inheritance

25. XML Drawables And 9-Patches

- XML Drawables
- State List Drawables

- Layer List and Inset Drawables
- Using 9-Patch Images

26. HTTP & Background Tasks

- Creating PhotoGallery
- Networking Basics
- Asking permission to network
- Using AsyncTask to Run on a Background Thread
- You and Your Main Thread
- Beyond the main thread
- Fetching XML From Flickr
- Using XmlPullParser
- From AsyncTask Back to the Main Thread
- For the More Curious: More on AsyncTask
- Cleaning Up AsyncTasks
- Challenge: Paging

27. Loopers, Handlers, and HandlerThread

- Preparing GridView for Displaying Images
- Downloading Lots of Small Things
- Communicating with the Main Thread
- Assembling a Background Thread
- Messages and Message Handlers
- Message anatomy
- Handler anatomy
- Using handlers
- Passing handlers
- For the More Curious: AsyncTask vs. Threads
- Challenge: Preloading and Caching

28. Search

- Searching Flickr
- The Search Dialog
- Creating a search interface
- Searchable activities
- Hardware search button
- How Search works
- Launch modes and new intents
- Simple persistence with shared preferences
- Using SearchView on Post-Android 3.0
- Challenges

29. Background Services

- Creating an IntentService

- What Services Are For
- Safe background networking
- Looking for New Results
- Delayed Execution with AlarmManager
- PendingIntent
- Managing alarms with PendingIntent
- Controlling Your Alarm
- Updating options menu items
- Notifications
- For the More Curious: Service Details
- What service does (and does not) do
- A service's lifecycle
- Non-sticky services
- Sticky services
- Bound services

30. Broadcast Intents

- Waking Up on Boot
- Broadcast receivers in the manifest
- How to use receivers
- Filtering Foreground Notifications
- Sending broadcast intents
- Dynamic broadcast receivers
- Using private permissions
- Receiving results with ordered broadcasts
- Receivers and Long-running Tasks

31. Browsing The Web & WebView

- One Last Bit of Flickr Data
- The Easy Way: Implicit Intents
- The Harder Way: WebView
- Using WebChromeClient to spruce things up
- Proper rotation with WebView
- For the More Curious: Injecting JavaScript Objects

32. Custom Views and Touch Events

- Setting Up the DragAndDraw Project
- Setting up DragAndDrawActivity
- Setting up DragAndDrawFragment
- Creating a Custom View
- Creating BoxDrawingView
- Handling Touch Events
- Tracking across motion events
- Rendering Inside onDraw(...)

- Challenge: Rotations

33. Tracking the Device's Location

- Getting Started with RunTracker
- Setting up RunActivity
- Setting up RunFragment
- Locations and the LocationManager
- Receiving Broadcast Location Updates
- Updating the UI with Location Data
- Faster Answers: the Last Known Location
- Testing Locations on Real and Virtual Devices

34. Local Databases with SQLite

- Storing Runs and Locations in a Database
- Querying a List of Runs From the Database
- Displaying a List of Runs Using CursorAdapter
- Creating New Runs
- Working with Existing Runs
- Challenge: Identifying the Current Run

35. Loading Asynchronous Data With Loaders

- Loaders and the LoaderManager
- Using Loaders in RunTracker
- Loading the List of Runs
- Loading a Single Run
- Loading the Last Location for a Run

36. Using Maps

- Adding the Maps API to RunTracker
- Use a real device to test maps
- Install and use the Google Play services SDK
- Obtain a Google Maps API key
- Update RunTracker's manifest
- Showing the User's Location on a Map
- Displaying a Run's Path
- Adding Markers for Run Start and Finish
- Challenge: Live Updates

37. Afterword

- The Final Challenge
- Shameless Plugs