Android Java
Live and In Action

Norman McEntire
Founder, Servin Corp
UCSD Extension Instructor
norman.mcentire@servin.com

Opening Remarks

- Welcome!
- Thank you!
- My promise to you is this:
  - Show you **Android Java** skills faster than any method on planet earth
    - Great INTRO if you are new to Android Java
    - Great REVIEW if you already know Android Java

Agenda

- Part 1 - 7:15pm - 7:50pm
  - 100% Android Java Source Code (No XML)
    - Architecture, Tools, Activities, Intents
- Part 2 – 8pm - 8:45pm
  - XML + Android Java Source Code
    - XML, Activities, Broadcast Receivers, Services, and Content Providers
- Part 3 – 8:45pm - 9pm
  - Questions/Answers

Summer 2013 UCSD Extension Android I Programming Course

- This presentation is a SAMPLE of the **Android I Programming** Course I give at UCSD Extension
- Summer 2013 Starts June 29!
  - 100% On-Line! REPEAT: 100% ON-LINE!
- Register Today!
  - http://extension.ucsd.edu
  - Android I always fills up! Register Today!

UCSD Extension Android I Programming - Course Schedule

- Total of Nine Lessons
- One Lesson Per Week
- Format of Each Lesson
  - Slides for Concepts
  - Coding Demos
  - Quiz
  - Programming Assignment

Course Lessons

- Lesson 1. Introduction
- Lesson 2. Widgets
- Lesson 3. Activities and Intents
- Lesson 4. Storage
- Lesson 5. Internet
- Lesson 6. Multimedia
- Lesson 7. Broadcast Receivers
- Lesson 8. Services
- Lesson 9. Content Providers

About MySelf

- Norman McEntire
  - norman.mcentire@servin.com
  - B.S/M.S. Computer Engineering
    - University of South Carolina
  - 30+ Years Computer Engineering Experience
    - Hardware and Software (Android, iOS, Linux)
  - UCSD Extension Instructor
    - Android I, Intro to Objective-C, iOS I, iOS II
  - Founder/Owner of Servin Corp
    - “Since 1995, Software Technology Training for Software Technology Professionals(tm)”

Question:

Why Cover **Android Java** at SDJUG?

Why Cover **Android Java** at SDJUG?

Answer #1:
Android Java is ONE OF the most widely used versions of Java used by Java Software Developers

Answer #2:
Android Java is THE most widely run JVM:
1+ Million More PER DAY!

So for the members of SDJUG, knowing **Android Java** is a valuable addition to your Java Skills Set

Part 1

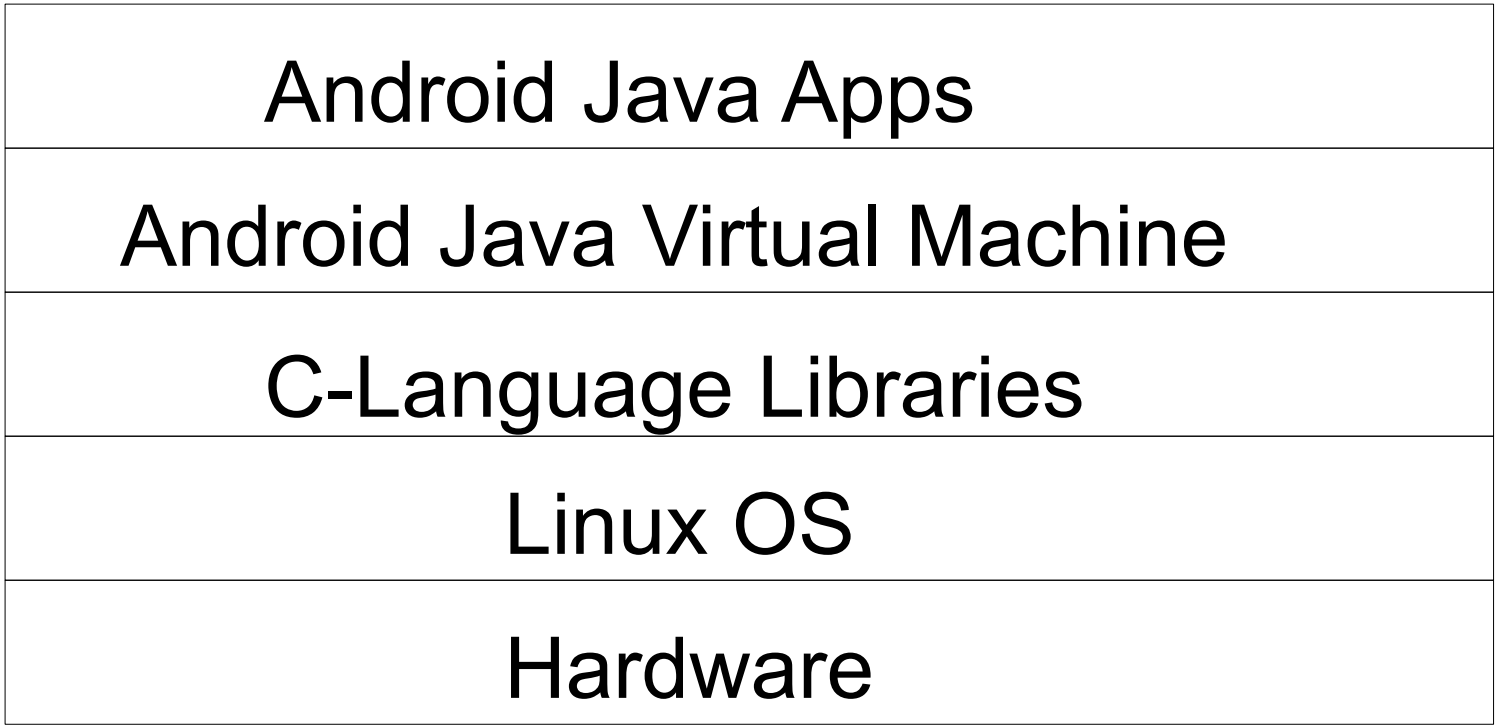
100% Android Java Source Code (No XML)

Architecture, Tools, and Activities

Key Android Skill

Android Architecture

Android Architecture



Key Android Skill:

Android Development Tools

ADT = Android Developer Tools

You can download **ADT Bundle** from here:
http://developer.android.com

Step 1. Startup ADT (Eclipse)

- Click on “Green icon” on Desktop
- Quick Tour of Eclipse
  - Workspace
  - Perspective
    - Java Perspective
    - DDMS Perspective
    - Debug Perspective
  - Views

Step 2. Create New Android Application Project

- File > New > Android Application Project
  - Application Name: SDJUG
  - Project Name: SDJUG
  - Package Name: org.sdjug.android.hello
  - Minimum Required SDK: API 8
    - DEMO: An aside on Android Versioning
  - Target SDK: 17
  - Next > Next > Next > Finish

Key Android Skill

android.app.Activity

MainActivity.java

- package org.sdjug.android.hello;
  - Package name will also be your process name!
- import android.os.Bundle
- import android.app.Activity
- public class MainActivity extends Activity
  - protected void onCreate(Bundle savedInstance)
    - super.onCreate(savedInstance)
    - //Not yet! setContentView(R.layout.activity_main)

Key Android Skill

android.widget.Toast

android.widget.Toast

- Displays a floating view over the application
- Never receives focus
- Disappears after a brief moment
- Example
  - onCreate()
    - Toast.makeText(this, “onCreate”, Toast.LENGTH_LONG).show();

Key Android Skill

Activity Lifecycle

Activity Lifecycle

- Demo 1
  - onCreate(), onRestart(), onDestroy()
- Demo 2
  - onStart(), onStop() - view/hide
  - onResume(), onPause() - interact/no-interact
- What to test for each demo
  - 1. Pressing HOME key
  - 2. Pressing BACK key
  - 3. Rotating Device

Key Android Skill

Touch Handling

onTouchEvent()

- public boolean onTouchEvent(MotionEvent event)
  - switch (event.getAction())
    - case MotionEvent.ACTION_DOWN:
      - // Handle Motion Down
      - return true
    - case MotionEvent.ACTION_UP:
      - // Handle Motion Up
      - return true
  - return super.onTouchEvent(event)

Key Android Skill

android.view.View

android.view.View

- All user interface elements in Android Java are built using **View** and **ViewGroup** objects
- A **View** draws something on the screen
- Demo
    - View view
    - onCreate()
        - view = new View(this)
        - view.setBackgroundColor(Color.RED)
        - setContentView(view)

android.graphics.Color

- Android color is a 32-bit value
  - Alpha, Red, Green, Blue
- Demo 1
  - int color = Color.YELLOW
  - view.setBackgroundColor(color)
- Demo 2
  - int color = Color.argb(255,0,255,0); //Green
  - view.setBackgroundColor(color)

android.widget.LinearLayout

- A ViewGroup is an object that holds other View and ViewGroup Objects
- Demo
    - LinearLayout linearLayout
    - linearLayout = new LinearLayout(this)
    - linearLayout.setOrientation(LinearLayout.VERTICAL)
    - linearLayout.setBackgroundColor(Color.GREEN)
    - setContentView(linearLayout)

android.widget.TextView

- Use a TextView widget to display text on the screen
- Demo
  - TextView textView
  - int width =
    LinearLayout.LayoutParams.MATCH_PARENT
  - int height =
    LinearLayout.LayoutParams.WRAP_CONTENT
  - textView = new TextView(this)
  - textView.setBackgroundColor(Color.YELLOW)
  - linearLayout.addView(textView,width,height)

android.widget.Button - 1

- Use a Button to allow user to send event to app
- Demo
    - Button button
    - int width =
      LinearLayout.LayoutParams.MATCH_PARENT
    - int height =
      LinearLayout.LayoutParams.WRAP_CONTENT
    - button = new Button(this)
    - button.setBackgroundColor(Color.CYAN)
    - button.setText(“Touch Me!”)
    - linearLayout.addView(button,width,height)

android.widget.Button - 2

- Using Anonymous Inner class to respond to button touch
- Demo
  - button.setOnClickListener(new View.OnClickListener()
    - public void onClick(View view)
      - Toast.makeText(MainActivity.this, “Touched The Button!“, Toast.LENGTH_LONG).show()

android.content.Intent

- An intent provides a facility for performing late runtime binding
  - The code can be in different applications
  - The most significant use of an Intent is launching activities
- Demo
  - Intent intent = new Intent()
  - intent.setAction(Intent.ACTION_DIAL)
  - startActivity(intent)

Part 1 Summary

- 100% Android Java Source Code
  - Architecture
  - Android Developer Tools
  - Activity (including Activity Lifecycle)
  - Toast
  - onTouchEvent(MotionEvent event)
  - View
  - TextView
  - Button
  - Intent

Android Java
Live and In Action

Part 2 – XML + Android Java

Norman McEntire
Founder, Servin Corp
UCSD Extension Instructor
norman.mcentire@servin.com

Key Android Skill

AndroidManifest.xml

AndroidManifest.xml

- The AndroidManifest.xml file defines all of your applications components
  - <?xml version=”1.0” encoding=”utf-8”?>
  - <manifest …
    - <uses-sdk …
    - <application …
      - <activity ....
      - <receiver …
      - <service …
      - <provider ...

To Create New Activity <activity … />

- The Activity is the foundation of the user interface
  - 1. Select AndroidManifest.xml
  - 2. Select Application Tab
  - 3. Scroll to bottom
  - 4. Application Nodes → Add Activity
    - Click on Name Hyperlink
    - Name: AboutActivity
    - Superclass: android.app.Activity

AboutActivity.java

- onCreate()
  - setTitle(“About This App”)
  - Button button = new Button(this)
  - button.setText(“Touch Button\nTo End Activity”)
  - button.setOnClickListener(new View.OnClickListener()
    - public void onClick(View view)
      - finish();
  - setContentView(button);

MainActivity.java

- onDoAboutActivity()
  - Intent intent = new Intent(this, AboutActivity.class)
  - startActivity(intent)

To Create New Broadcast Receiver <receiver … />

- A Broadcast Receiver receives broadcasts
  - 1. Select AndroidManifest.xml
  - 2. Select Application Tab
  - 3. Scroll to bottom
  - 4. Application Nodes → Add Receiver
  - Click on Name Hyperlink
  - Name: MyReceiver
  - Superclass: android.content.BroadcastReceiver

To Create New Service <service … />

- A Service runs in the background without a user interface
  - 1. Select AndroidManifest.xml
  - 2. Select Application Tab
  - 3. Scroll to bottom
  - 4. Application Nodes → Add Service
  - Click on Name Hyperlink
  - Name: MyService
  - Superclass: android.app.Service

To Create New Provider <provider … />

- A Content Provider provides content to other applications
  - 1. Select AndroidManifest.xml
  - 2. Select Application Tab
  - 3. Scroll to bottom
  - 4. Application Nodes → Add Provider
    - Click on Name Hyperlink
    - Name: MyProvider
    - Superclass: android.content.ContentProvider
    - Authorities: org.sdjug.android.demo

Key Android Skill

XML Layout

res/layout

- res/layout/main.xml
  - Use this as the default layout
- res/layout-port/main.xml
  - Use this as the layout for portrait mode
- res/layout-land/main.xml
  - Use this as the layout for landscape mode

res/layout/main.xml

- <?xml version=”1.0” encoding=”utf-8”?>
- <LinearLayout
  - android:layout_width=”match_parent”
  - android:layout_height=”match_parent”
  - android:orientation=”vertical”>
  - <View
    - android:id=”@+id/view1”
    - android:layout_width=”wrap_content”
    - android:layout_height=”wrap_content”

Key Android Skill

Mapping XML ids To Java Objects

findViewById()

- Use findViewById() to map from XML layout to Android Java class
  - View view
  - view = findViewById(R.id.view1)
  - ...

Key Android Skill

XML Values

res/values/strings.xml

- <?xml version=”1.0” encoding=”utf-8”?>
- <resources>
  - <string name=”app_name”>SDJUG</string>
  - <string name=”hello_world”>Hello World!</string>
- </resources>

res/values/colors.xml

- <?xml version=”1.0” encoding=”utf-8”?>
- <resources>
  - <color name=”green”>#ff00ff00</color>
- </resources>

getResources().getColor()

- Use getResources().getColor() to map from XML value to integer color
  - View view
  - view = findViewById(R.id.view1)
  - int color = getResources().getColor(R.color.green)
  - view.setBackgroundColor(color)

Part 2 Summary

- XML + Android Java Source Code
  - AndroidManifest.xml
  - res/layout
    - res/layout-port/main.xml
    - res/layout-land/main.xml
    - findViewById()
  - res/values
    - res/values/strings.xml
    - res/values/colors.xml

Summary

- This was a sample of the topics covered in the UCSD Extension Course titled Android I Programming
  - Starts June 29! 100% ON-LINE! Register Today!
- In Part 1, we did an app with 100% Android Java Source Code
- In Part 2, we did an app with XML + Android Java Source Code

Part 3

Questions / Answers

