



# Android Programming: Installation, Setup, and Getting Started

Originals of Slides and Source Code for Examples:  
<http://www.coreservlets.com/android-tutorial/>

**Customized Java EE Training:** <http://courses.coreservlets.com/>  
Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.  
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.



**For live Android training, please see courses at <http://courses.coreservlets.com/>.**

**Taught by the author of *Core Servlets and JSP*, *More Servlets and JSP*, and this Android tutorial. Available at public venues, or customized versions can be held on-site at your organization.**



- Courses developed and taught by Marty Hall
    - JSF 2, PrimeFaces, servlets/JSP, Ajax, jQuery, Android development, Java 6 or 7 programming, custom mix of topics
    - Ajax courses can concentrate on 1 library (jQuery, Prototype/Scriptaculous, Ext-JS, Dojo, etc.) or survey several
  - Courses developed and taught by coreservlets.com experts (edited by Marty)
    - Spring, Hibernate/JPA, EJB3, GWT, Hadoop, SOAP-based and RESTful Web Services
- Contact [hall@coreservlets.com](mailto:hall@coreservlets.com) for details

# Topics in This Section

- **Installing the Software and Documentation**
  - Java 6
  - Eclipse
  - Android SDK base
  - Eclipse ADT Plugin
  - Updated SDK components
  - AVD (Android Virtual Device)
- **Running Apps**
  - Import and test an existing app
    - Run on emulator
  - Create and test a new app
    - Run on emulator
  - Seeing standard output in the DDMS
  - Deploy app to USB-connected Android device

4

© 2012 Marty Hall



## Installing the Software and Documentation

**Customized Java EE Training: <http://courses.coreservlets.com/>**

Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.  
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

# JDK for Java SE 6

## • Overview

- Use Java 6 (aka JDK 1.6)
  - Java 5 supported only by older Android versions
  - Java 7 can technically be used, but new Java 7 features not supported
- For PC, Linux, Solaris, follow directions at <http://www.oracle.com/technetwork/java/javase/downloads/>
  - Get JDK, not just JRE
  - Get SE (Standard Edition), not EE or Micro Edition
  - Don't get version with the NetBeans IDE
- For MacOS, Java is preinstalled & updated automatically

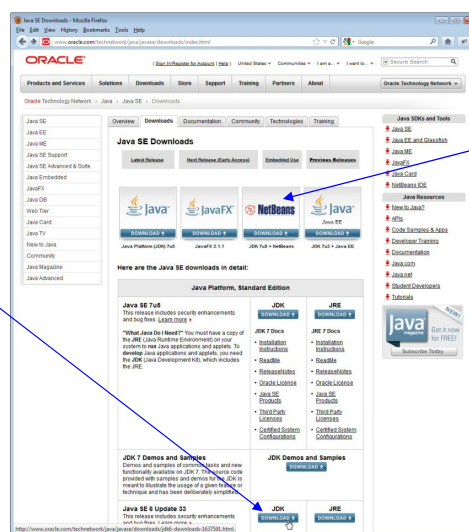
6

# Installing Java SE 6

## • Install Java 6 JDK

- <http://www.oracle.com/technetwork/java/javase/downloads/>
  - Scroll down for Java SE 6

Use this version. The "JDK – Java Development Kit" includes compiler for .java files, whereas the "JRE – Java Runtime Environment" is only for executing prebuilt .class files.



This tutorial uses Eclipse, so do not use this link.

As of summer 2012, there is no NetBeans plugin for Android development. So, Eclipse is strongly recommended even if you normally use NetBeans for Java development. However, IntelliJ IDEA has Android support: see [http://www.jetbrains.com/idea/features/google\\_android.html](http://www.jetbrains.com/idea/features/google_android.html)

After downloading, run installer and accept all defaults.

# Eclipse

## • Overview

- Eclipse is a free open source IDE (Integrated Development Environment). Support for Java, HTML, CSS, JavaScript, C++, PHP, and more.
- Google has free Eclipse plugin to integrate with the Android SDK.

## • Features

- General
  - Checks your syntax as you type
  - Automatically compiles every time you save file
  - Refactoring, debugging, templates for common tasks, etc.
- Android-specific
  - Deploy apps to Android emulator
  - Configure virtual environments
  - Drag-and-drop GUI builder

8

# Installing Eclipse

## • Go to eclipse.org, click on “Downloads”

– Results in <http://eclipse.org/downloads/>

– Can use either “for Java” or “for Java EE”.

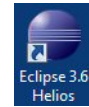
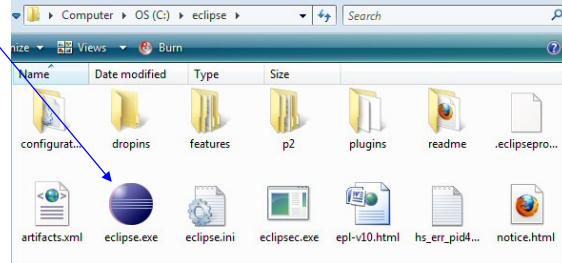
- I use EE version since I also use Eclipse for Web apps.

- Latest version (3.7 – Indigo) recommended.
  - Previous version (3.6 – Helios) still supported
  - Older versions (3.5 and earlier) not supported

9

# Running Eclipse

- **Unzip the downloaded file (no installer!)**
  - Call the folder you unzip into “installDir”
- **Double click eclipse.exe**
  - From *installDir/bin*
- **Click on “Workbench” icon**
  - Next time you bring up Eclipse, it will come up in workbench automatically
- **Shortcut**
  - Many developers put Eclipse link on their desktop
    - R-click eclipse.exe, Copy, then go to desktop, R-click, and Paste Shortcut (not just Paste!)



# The Android SDK

- **Overview**
  - Android-specific libraries
  - Dalvik (Android virtual machine) compiler
  - Android emulator (to run without physical device)
  - DDMS debugging environment
- **Documentation**
  - Installation
    - <http://developer.android.com/sdk/installing/index.html>
  - Developer’s Guide
    - <http://developer.android.com/guide/components/index.html>
  - JavaDoc (API Reference)
    - <http://developer.android.com/reference/classes.html>
  - Tutorials and articles
    - <http://developer.android.com/resources/index.html>

Bookmark these URLs!



# Installing the Android SDK

- **Download and run installer**
  - From <http://developer.android.com/sdk/>
    - I install in C:\android-sdk
  - Sets up basic SDK, but omits many components
- **Detailed instructions**
  - <http://developer.android.com/sdk/installing/index.html>
- **Postponed step**
  - After installing Eclipse plugin, we will run the Android SDK Manager to get important missing components
    - Easiest to do from Eclipse. See upcoming slide after Eclipse ADT installation.

12

# Eclipse ADT Plugin

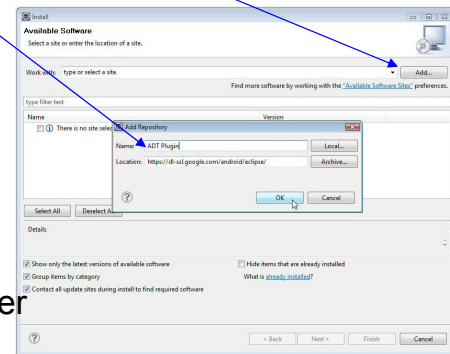
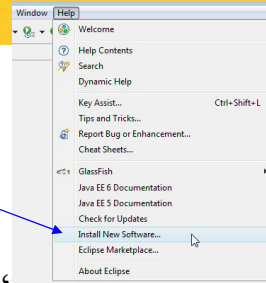
- **Overview**
  - ADT (Android Development Tools) provides many useful features accessible directly in Eclipse
    - Integration between Eclipse & Android command-line tools
    - Drag-and-drop GUI builder
    - Many development and debugging aids
- **Detailed installation instructions**
  - <http://developer.android.com/sdk/installing/installing-adt.html>
- **More details**
  - <http://developer.android.com/sdk/eclipse-adt.html>

13

# Installing Eclipse ADT

## • Steps

- Start Eclipse
  - Help → Install New Software ...
  - Click “Add” in upper-right
  - In Add Repository, for Name enter ‘ADT Plugin’ and for Location enter <https://dl-ssl.google.com/android/eclipse/>
  - Click OK, select checkbox next to Developer Tools, Next, see packages to be installed, accept license, Finish. Restart Eclipse if asked.
  - Run Android SDK Manager
    - Should be prompted; if not, Windows → Android SDK Manager
- See next slide for details



14

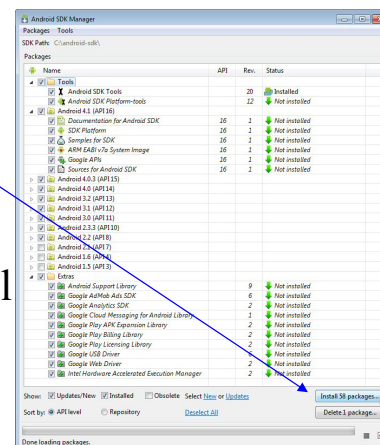
# Updating SDK Components

## • Run Android SDK manager

- Window → Android SDK Manager
  - Probably prompted automatically after installing ADT plugin and restarting Eclipse
- Select all entries, except that you can omit Android versions older than 2.2
  - Click “Install *n* packages”
  - Runs for a long time

## • Detailed instructions

- <http://developer.android.com/sdk/installing/adding-packages.html>



15

# Android Virtual Devices (AVDs)

## • Overview

- An AVD (Android Virtual Device) is an Android Emulator configuration that lets you model an actual device by defining hardware and software options

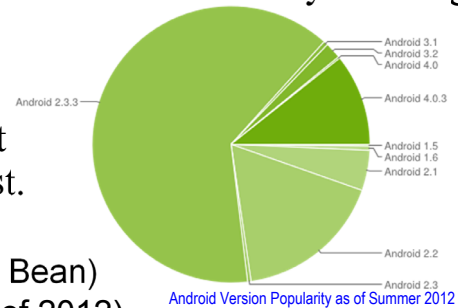
## • Idea

- Define several AVDs at different Android API levels to test against.

At least two:

- Newest version (e.g., 4.1 – Jelly Bean)
- Most common version (2.3.3 as of 2012)

- To see statistics for versions of currently used Android devices, see <http://developer.android.com/about/dashboards/index.html>



## • Detailed instructions

- <http://developer.android.com/guide/developing/devices/managing-avds.html>

16

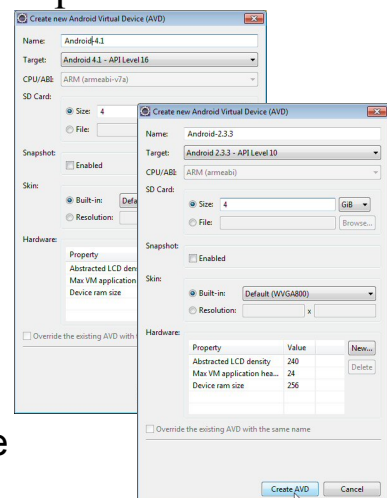
# Defining an AVD

## • Defining

- Window → AVD Manager
- Click on “New” at top. Choose name (arbitrary) and options. You can (should!) create multiple AVDs.

## • Options

- Target (i.e., target API version)
  - 4.1 to test tablet and new features
  - 2.3.3 to test most common phones
- SD Card size
  - Can be omitted. Or, choose middle of the road value, e.g., 4 GB
- Skin
  - Use default for the target you chose



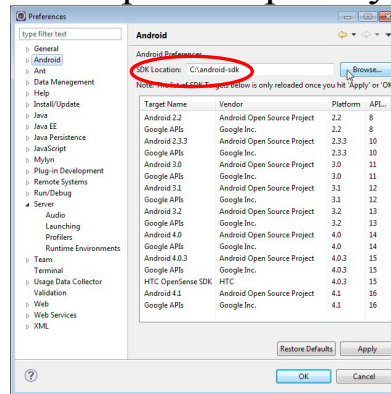
17



# Configuring Eclipse ADT

- **Set SDK Location**

- Window → Preferences → Android
- Click Browse and point at place you installed the SDK



- **Optional: disable sending stats to Google**

- Window → Preferences → Android → Usage Stats

18

© 2012 Marty Hall



## Running Apps on Emulator

Customized Java EE Training: <http://courses.coreservlets.com/>

Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.

Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

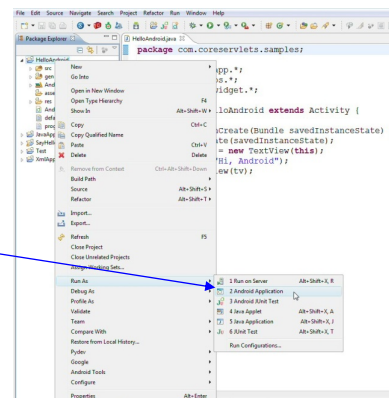
# Big Ideas

- **Running apps**
  - Soon, you want to learn how to write your own apps. First, however, we will practice running existing apps.
- **Ways to run**
  - Covered here
    - On the Android Emulator. Deploy directly from Eclipse.
      - During development, do your normal testing here
    - On an Android device. Deploy from your PC via USB.
  - Other options covered online
    - On an Android device. Deploy from a Web site.
    - On an Android device. Deploy via email.
    - On and Android device. Deploy from the Android Market
- **Writing apps**
  - Covered in later tutorial sections. That is the fun part!

20

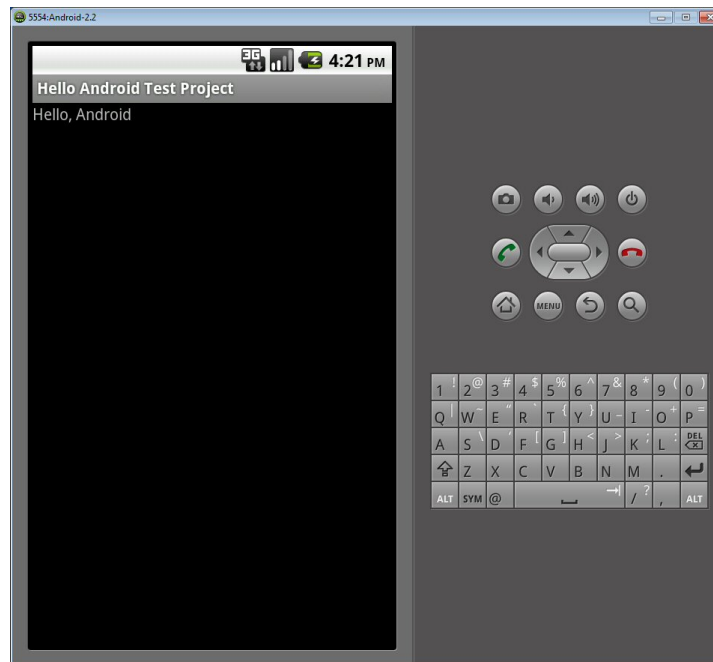
# Running the HelloAndroid App in Emulator

- **HelloAndroid**
  - Super-simple app to test deployment and execution steps.
    - For students in live classes, this project is already in your Eclipse workspace.
    - For online readers, download project from the Getting Started section of Android tutorial and import into Eclipse.
      - <http://www.coreservlets.com/android-tutorial/>
- **Steps to run it**
  - Import it if necessary
    - File → General → Existing Projects...
  - R-click on project on L side of Eclipse
  - Run As → Android Application
    - *Long wait while emulator initializes*
    - **Do not close emulator when done**
      - Next time, app will come up much faster



21

# HelloAndroid: Result



22

Code will be discussed in next tutorial section.

© 2012 [Marty Hall](#)



## Making Your Own Android App

**Customized Java EE Training: <http://courses.coreservlets.com/>**

Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.  
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

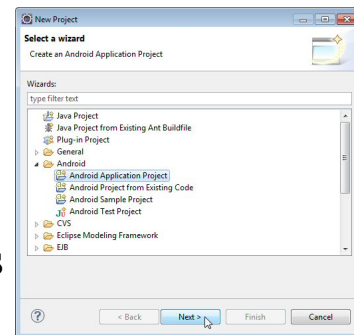
# Making Your Own Android App: Basics

- **Idea**

- When you create a new app, it has simple “Hello World” functionality built in.
  - So, you can create and test an app without knowing syntax (which is not discussed until next tutorial section)

- **Steps**

- File → New → Project → Android → Android Application Project
  - Next time you can do File → New → Android Application Project
- Fill in options as shown on next pages
- Run new project as shown previously
  - R-click → Run As → Android Application



24

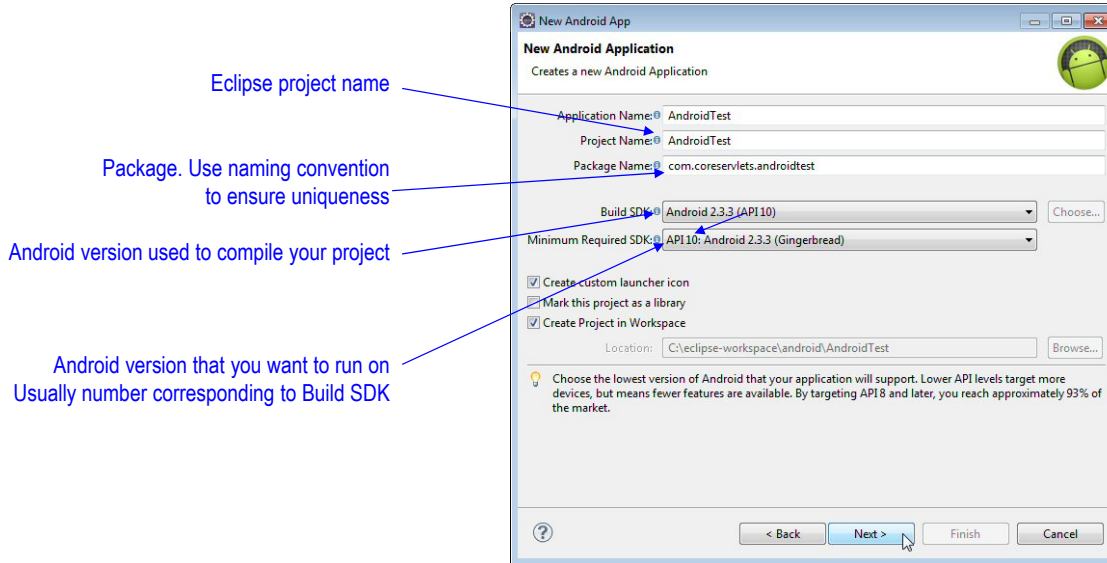
# New Android App: Setting Project Options (Pg. 1)

- **New Android Project Settings: Page 1**

- Application Name
  - Shown in Play Store and Settings → Manage Application List. Usually same as Project Name.
- Project Name
  - Eclipse project name. Follow naming convention you use for Eclipse. Not used elsewhere.
- Package name
  - Apps on a particular Android device must have unique packages, so use `com.yourCompany.project`
- Build SDK
  - The Android version used to build/compile your project. This can be any version (e.g., the most recent), but the safest option is to make it match the minimum SDK below.
- Minimum Required SDK
  - The Android version that you want to run on. For most phone apps, choose 2.3.3, since that is the most common version in use worldwide. For learning new features, use latest version (4.1 as of summer 2012).
  - The safest option is to use a number to match Build SDK. Summarized in Eclipse dialog, but for details, see <http://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>

25

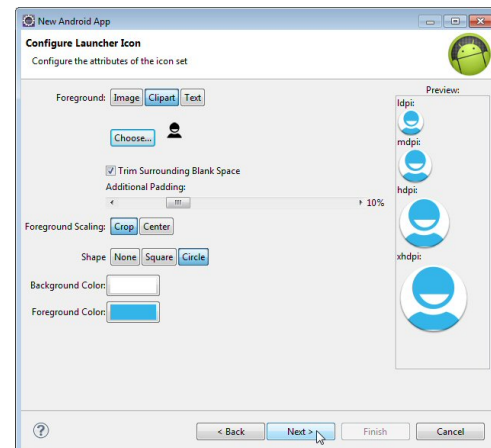
# New Android App: Setting Project Options (Pg. 1)



26

# New Android App: Configure Launcher Icon

- **Purpose**
  - To choose the picture displayed on the Android device, that, when clicked, launches the app.
- **Defaults**
  - Use defaults for development and testing. Just press “Next”.

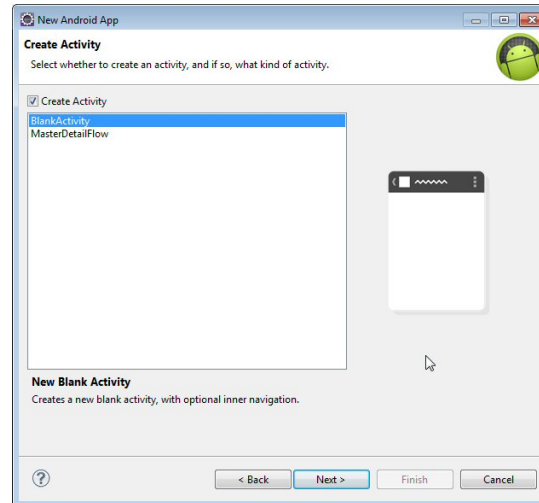


27



# New Android App: Create Activity

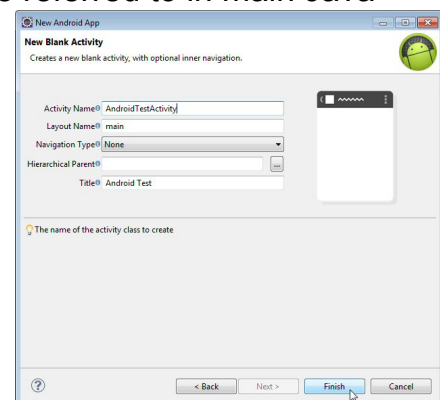
- **Choose “BlankActivity”**
  - The Master/Detail option is used rarely, and applies only to Android 3.0 or later.



28

# New Android App: New Blank Activity

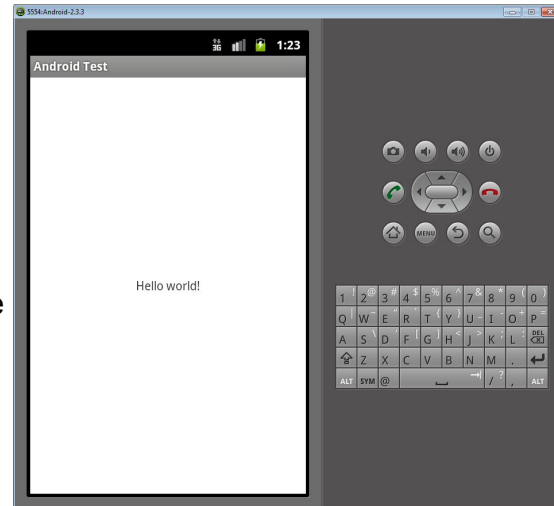
- **Options for Blank Activity**
  - Activity Name
    - Name of “main” Java class. This is the class you will edit first. Class name often corresponds to project name.
  - Layout Name
    - Base name of XML file in res/layout folder. Used to give layout to app. Often just called “main”. Will be referred to in main Java class with `R.layout.layout_name`.
  - Navigation Type
    - For now, leave this as “None”
  - Hierarchical Parent
    - Parent Activity (for when user presses Up). Empty for now.
  - Title
    - Text that will be shown on Android title bar.



29

# Running New App on Emulator

- **Builtin functionality**
  - Newly created projects automatically have simple “Hello World” behavior
- **Execution steps**
  - Same as with any project
    - R-click → Run As → Android Application
      - Reminder: do not close emulator after testing. Emulator takes a long time to start initially, but it is relatively fast to deploy a new or a changed project to the emulator.



30

© 2012 Marty Hall



## Seeing Standard Output in DDMS

Customized Java EE Training: <http://courses.coreservlets.com/>

Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.  
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

# DDMS Basics

- **Idea**

- DDMS (Dalvik Debug Monitor Service) is a tool that supports many things
  - Simulate incoming calls in emulator
  - Set GPS locations in emulator
  - See print statements and runtime errors
  - Set locations and take screenshots of actual Android device

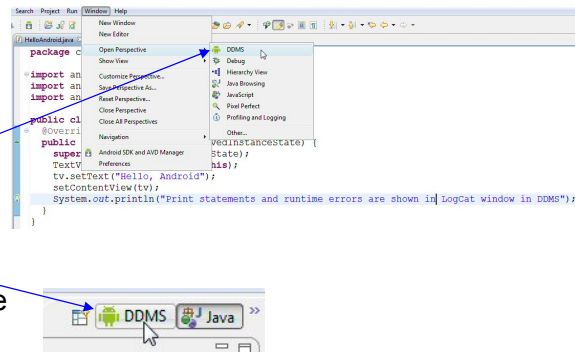
- **Simple usage now**

- Start DDMS

- Window → Open Perspective → Other → DDMS
- Once you do this once, you can click on “DDMS” at top right of Eclipse
- Click on “Java” to return to code

- See print statements

- Look in LogCat window at bottom
- Type part of output into Filter field to see specific output



32

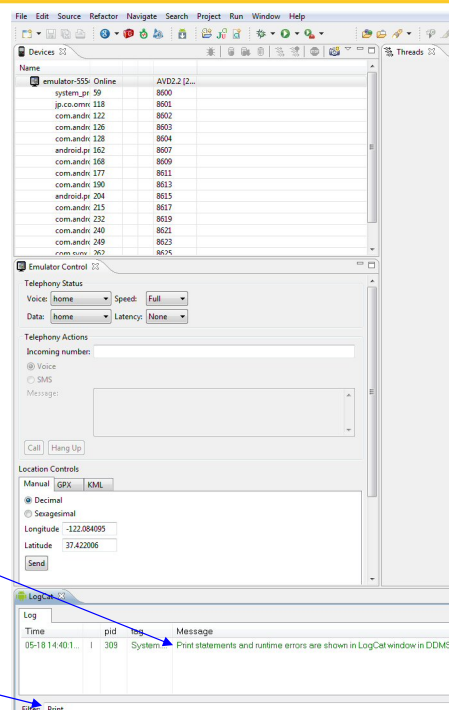
# DDMS Output (HelloAndroid)

- **Code**

- Put `System.out.println` in main `onCreate` method
  - See code in screenshot on previous page
  - `onCreate` and other syntax discussed in next section

Output of `System.out.println`

Entered so it is easier to find specific output among the many informational messages that emulator prints



33



# Running Apps on Physical Android Device

**Customized Java EE Training: <http://courses.coreservlets.com/>**

Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.  
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

## Running Apps on Android Device

- **Idea**
  - The vast majority of your testing will be on Android emulator. But sometimes you want to test on a physical phone or other Android device to test compatibility and to use camera, GPS, contact list, etc.
    - You first make a signed application package (YourApp.apk), then you have various options for sending it to the phone
- **Options**
  - Covered here
    - **Connect phone via USB, use Eclipse to deploy**
  - Can learn on your own
    - Export signed app from Eclipse, use adb to install on phone
      - See <http://developer.android.com/tools/help/adb.html#move>
    - Other approaches
      - [http://developer.android.com/tools/publishing/publishing\\_overview.html](http://developer.android.com/tools/publishing/publishing_overview.html)
        - Submit app to Android marketplace
        - Email apk file to email address of phone
        - Deploy apk file to a Web site, then connect there from phone. Must set MIME type to application/vnd.android.package-archive.

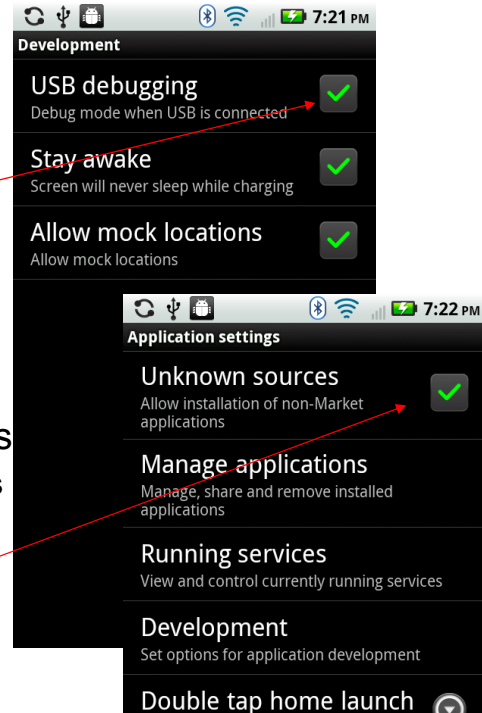
# Deploying via USB Connection

- **Prereq: install drivers for Android device**
  - Plug phone (or other Android device) into computer.
    - Recent OS's might find drivers automatically. If not, download from device manufacturer. See list at <http://developer.android.com/tools/extras/oem-usb.html>
      - After installation, plug in phone, then go to `android-dir/platform-tools/` and run "adb devices". Your device should now be listed.
- **Steps**
  - Android device
    - Enable USB debugging
    - Allow unknown sources
    - Verify USB drivers are on computer
      - <http://developer.android.com/tools/extras/oem-usb.html>
    - Plug into computer via USB
  - Eclipse
    - R-click app, Run As → Android Application
      - If emulator is open, will be given a choice of which device to deploy to. If emulator not open, will deploy to physical device automatically.

36

# Configuring Android Device

- **Enable USB debugging**
  - Settings → Applications → Development
    - Required: USB debugging
      - Allows PC to send commands via USB
    - Optional: Stay awake
      - Phone/device won't sleep when connected via USB
    - Optional: Allow mock locations
      - Let PC send fake GPS locations
- **Allow unknown sources**
  - Settings → Applications → Unknown sources

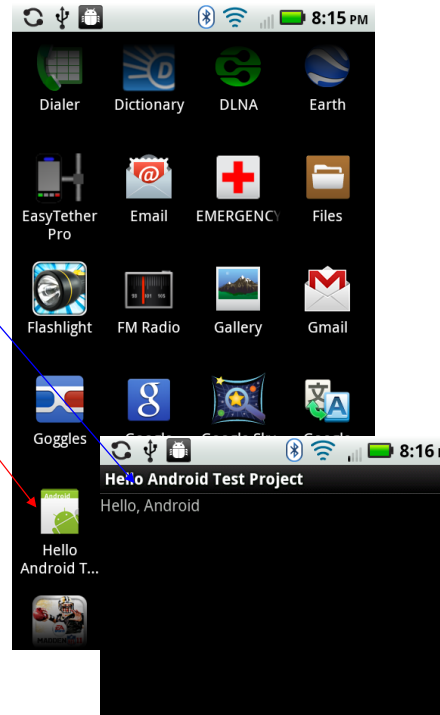


37



# Running App on Device

- **After running via Eclipse**
  - Will show automatically
- **To run later**
  - Go to installed apps
  - Tap new app
    - Remember that the human readable name (Application Name from new Android Project wizard) is shown, not the Java class name
  - If you want to install update
    - Uninstall old version first
      - Settings → Applications → YourApp → Uninstall
      - Or long-press your app and choose “Uninstall”



38

© 2012 Marty Hall



## Wrap-Up

Customized Java EE Training: <http://courses.coreservlets.com/>

Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.  
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

# Summary

- **Install software**
  - Java 6, Eclipse, Android SDK, Eclipse ADT plugin
- **Bookmark documentation**
  - Developer's Guide and more at [developer.android.com](http://developer.android.com)
- **Update/configure software**
  - Set SDK location in Eclipse
  - Get updated components via Android SDK Manager
  - Define at least one AVD to run apps on emulator
- **Run apps**
  - R-click project, Run As → Android Application
    - On emulator (usually)
    - On physical Android device (once in a while)
- **Make new app**
  - File → New → Project → Android → Android Application Project
- **See output of print statements**
  - In LogCat window of DDMS

40

© 2012 [Marty Hall](#)



## Questions?

[JSF 2, PrimeFaces, Java 7, Ajax, jQuery, Hadoop, RESTful Web Services, Android, Spring, Hibernate, Servlets, JSP, GWT, and other Java EE training.](#)

**Customized Java EE Training: <http://courses.coreservlets.com/>**

Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.

Developed and taught by well-known author and developer. At public venues or onsite at *your* location.