

A green Android robot is the central focus, wearing a pair of brown headphones. The robot's head is slightly tilted, and its eyes are visible. The background is a solid light blue color. The text 'Android User Interface' is overlaid on the robot's face in a large, bold, black font.

Android User Interface

Jestin James M
Assistant Professor, Dept of Computer Science
Little Flower College, Guruvayoor

The Android UI



- The Android UI is essentially a fourth-generation UI framework
- traditional C-based Microsoft Windows API the first generation
- the C++-based Microsoft Foundation Classes (MFC) the second generation.
- The Java-based Swing UI framework would be the third generation

The Android UI



- The Android UI, JavaFX, Microsoft Silverlight, and Mozilla XML User Interface Language (XUL) fall under this new type of fourth-generation UI framework
- Programming in the Android UI involves declaring the interface in XML files
- This is very much like HTML-based web pages. Much as in HTML, you find the individual controls through their IDs and manipulate them with Java code

The Android UI



- Screens or windows in Android referred to as *activities*,
- *which comprise multiple views that a user needs in order to accomplish a logical unit of action.*
- *Views are Android's basic UI building blocks*
- *combine them to form composite views called view groups.*

The Android UI



- Android 3.0 introduced a new UI concept called *fragments* to allow developers to chunk views and functionality for display on tablets
- One of the Android framework's key concepts is the life cycle management of activity windows.
- Protocols are put in place so that Android can manage state as users hide, restore, stop, and close activity windows

The Android Foundational Components



- An intent is an intra- and inter process mechanism to invoke components in Android.
- A component in Android is a piece of code that has a well defined life cycle.
- An activity representing a window in an Android application is a component

The Android Foundational Components

- Here is an example of using the Intent class to invoke or start a web browser:

```
• public static void invokeWebBrowser(Activity activity)
{
  Intent intent = new Intent(Intent.ACTION_VIEW);
  intent.setData(Uri.parse("http://www.google.com"));
  activity.startActivity(intent);
}
```

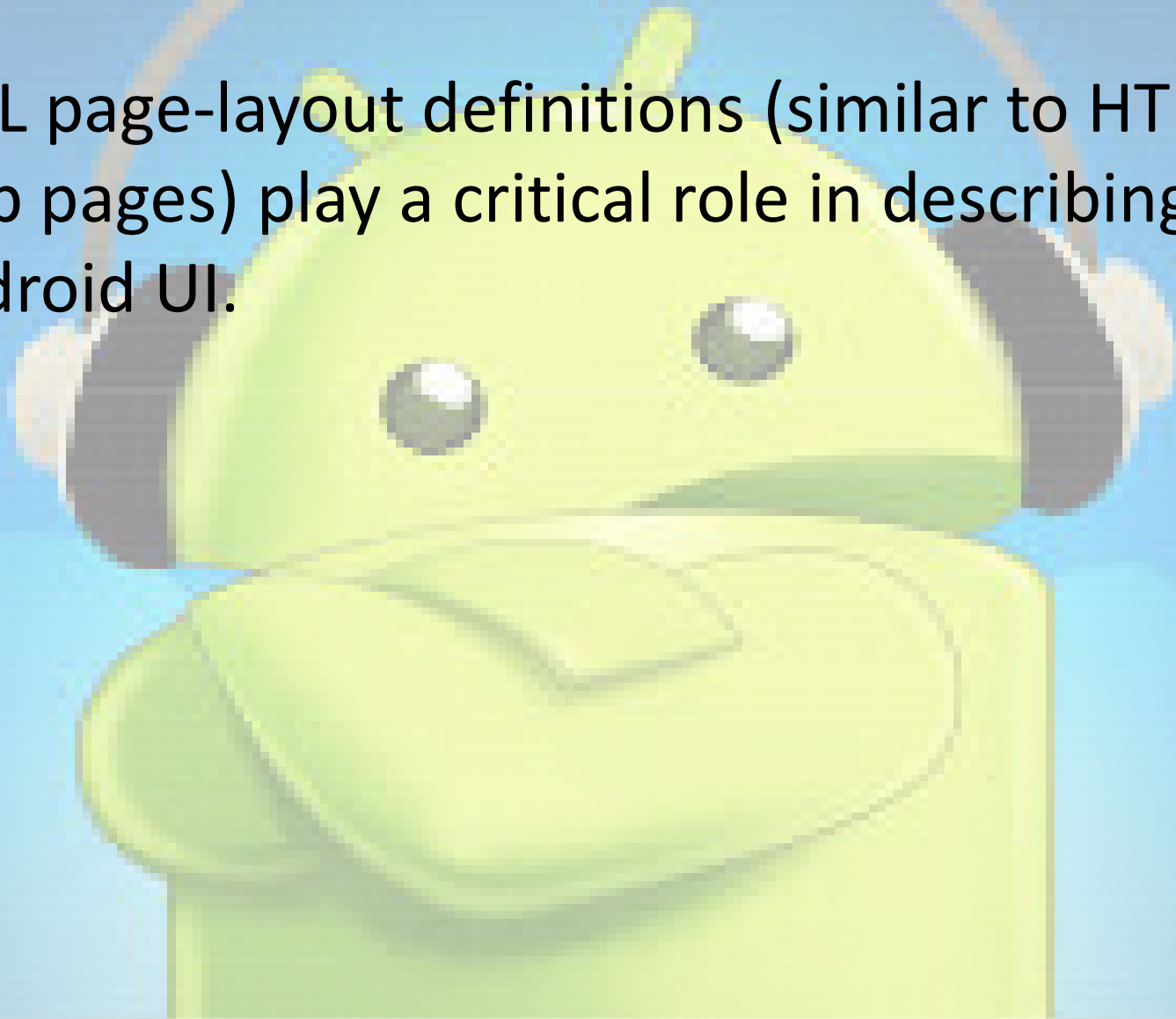
The Android Foundational Components



- In this example, through an intent, we are asking Android to start a suitable window to display the content of a web site.
- Depending on the list of browsers that are installed on the device, Android will choose a suitable one to display the site

Advanced UI Concepts

- XML page-layout definitions (similar to HTML web pages) play a critical role in describing the Android UI.



Advanced UI Concepts

- `<?xml version="1.0" encoding="utf-8"?>`
- `<!-- place it in /res/layout/sample_page1.xml -->`
- `<!-- will auto generate an id called:
R.layout.sample_page1 -->`
- `<LinearLayout ..some basic attributes..>`
- `<TextView android:id="@+id/textViewId"`
- `android:layout_width="fill_parent"`
- `android:layout_height="wrap_content"`
- `android:text="@string/hello"`
- `/>`
- `</LinearLayout>`

Advanced UI Concepts

A green Android robot character is positioned in the background of the slide. It is a stylized, rounded figure with a friendly appearance, wearing a white headband with a small antenna. The robot is set against a light blue gradient background.

- You will use an ID generated for this XML file to load this layout into an activity window
- Android supports dialogs, Menus
- Android offers extensive support for animation

Advanced UI Concepts



- 3 ways to do animation
- You can do frame-by-frame animation
- you can provide tweening animation by changing view transformation matrices (position, scale, rotation, and alpha)
- you can also do tweening animation by changing properties of objects.

Advanced UI Concepts



- Android has a number of new concepts that revolve around *information at your fingertips* using the home screen.
- Using live folders, you can publish a collection of items as a folder on the homepage.
- The second homepage-based idea is the *home screen widget*

Advanced UI Concepts



- *Integrated Android Search is the third homepage-based idea.*
- *Using integrated search, search for content both on the device and also across the Internet*
- Android also supports touch screen and gestures based on finger movements on the device.

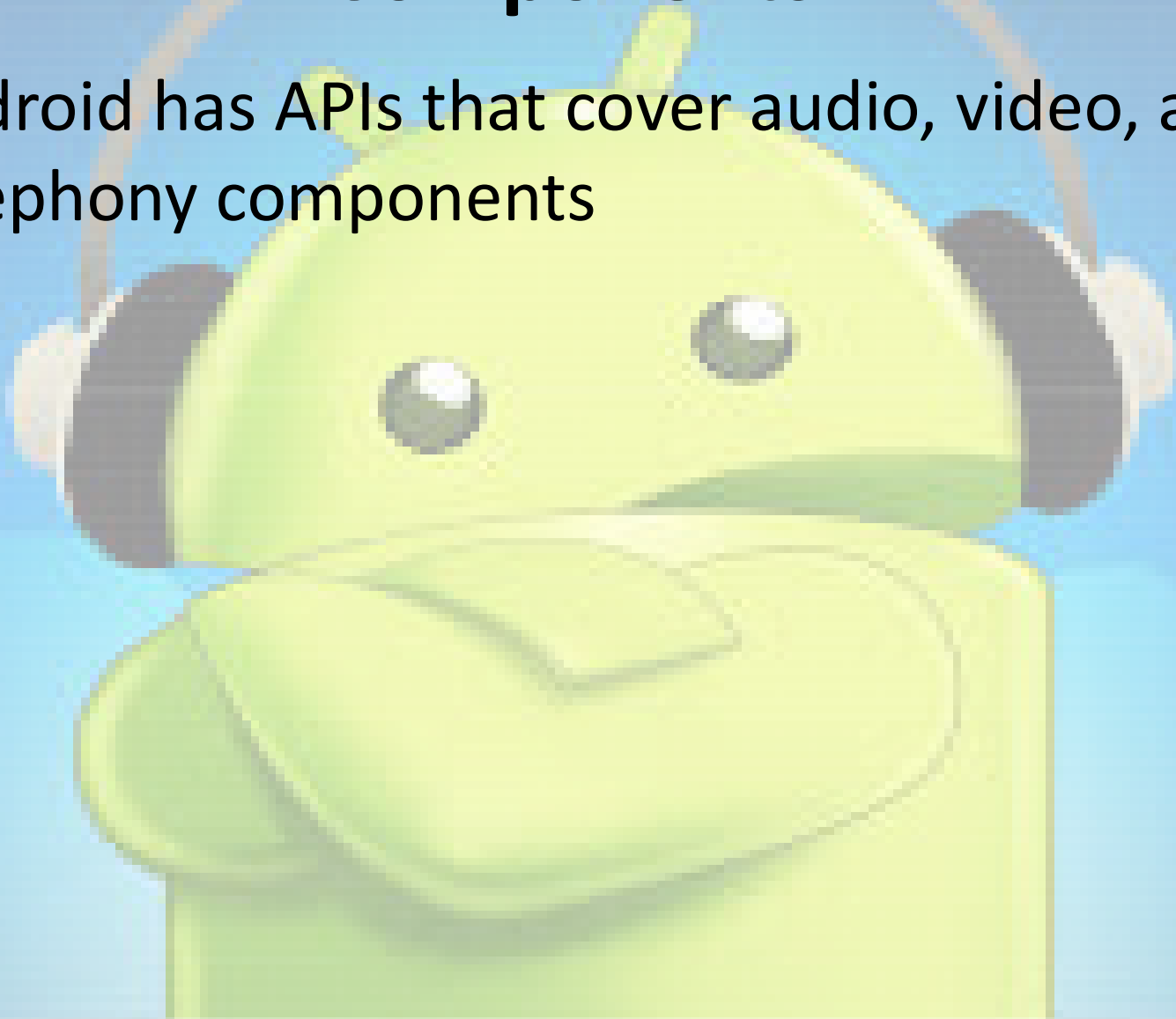
Android Service Components



- Security is a fundamental part of the Android platform.
- Location-based service is another of the more exciting components of the Android SDK.

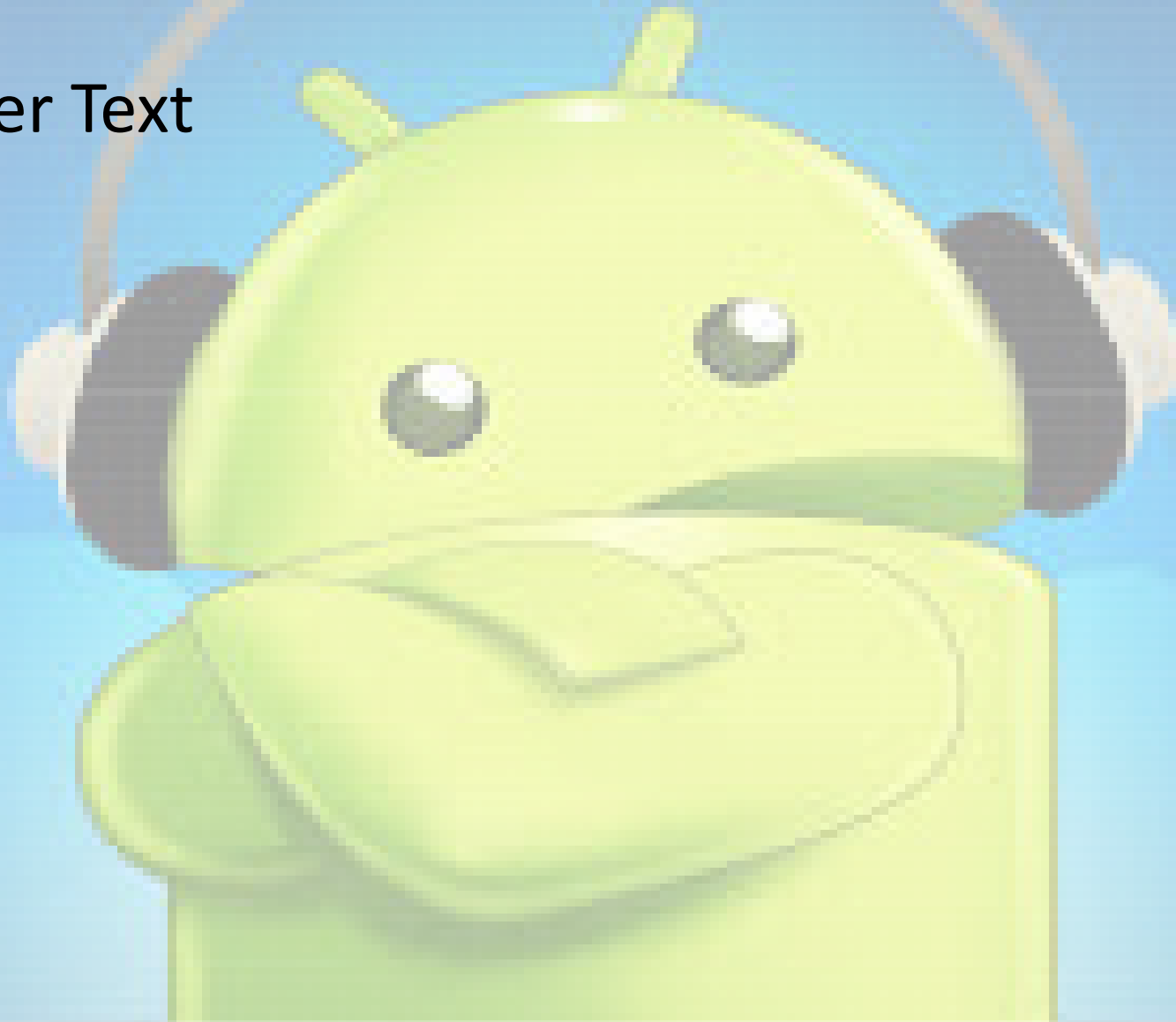
Android Media and Telephony Components

- Android has APIs that cover audio, video, and telephony components



Android Java Packages

- Refer Text



A green Android robot is the central focus, wearing large, brown and black headphones. The robot has a friendly, rounded appearance with two small antennae on its head. The background is a solid, light blue color. The text is overlaid on the left side of the image.

Chapter 2

SETTING UP YOUR DEVELOPMENT ENVIRONMENT

Setting Up Your Development Environment

- The Android SDK requires JDK 5 or JDK 6 (the examples use JDK 6) and Eclipse 3.5 or higher
- The Android SDK is compatible with Windows (Windows XP, Windows Vista, and Windows 7),
- Mac OS X (Intel only), and Linux (Intel only).
- In terms of hardware, you need an Intel machine, the more powerful the better

Setting Up Your Development Environment

- Android Development Tools (ADT).
- ADT is an Eclipse plug-in that supports building Android applications with the Eclipse IDE.
- The Android SDK is made up of two main parts:
 - The tools and the packages

Setting Up Your Development Environment

- When you first install the SDK, all you get are the base tools.
- The packages are the files specific to a particular version of Android (called a *platform*) or a *particular add-on to a platform*

Setting Up Your Environment

1. Downloading JDK 6

- The first thing you need is the Java SE Development Kit.
- The Android SDK requires JDK 5 or higher; we developed the examples using JDK 6.
- For Windows, download JDK 6 from the Oracle web site (www.oracle.com/technetwork/java/javase/downloads/index.html) and install it

1. Downloading JDK 6

- JAVA_HOME environment variable to point to the JDK install folder
- For Windows Vista and Windows 7, the steps to get to the Environment Variables screen are a little different.
- Choose Start ➤ Computer, right-click, choose Properties, click the link for Advanced System Settings, and click Environment Variables

1. Downloading JDK 6

- click New to add the variable or Edit to modify it if it already exists.
- The value of JAVA_HOME is something like C:\Program Files\Java\jdk1.6.0_27.

2. Downloading Eclipse 3.6

- download the Eclipse IDE for Java Developers
- You can download all versions of Eclipse from www.eclipse.org/downloads/.
- When you first start up Eclipse, it asks you for a location for the workspace.
- To make things easy, you can choose a simple location such as C:\android or a directory under your home directory.

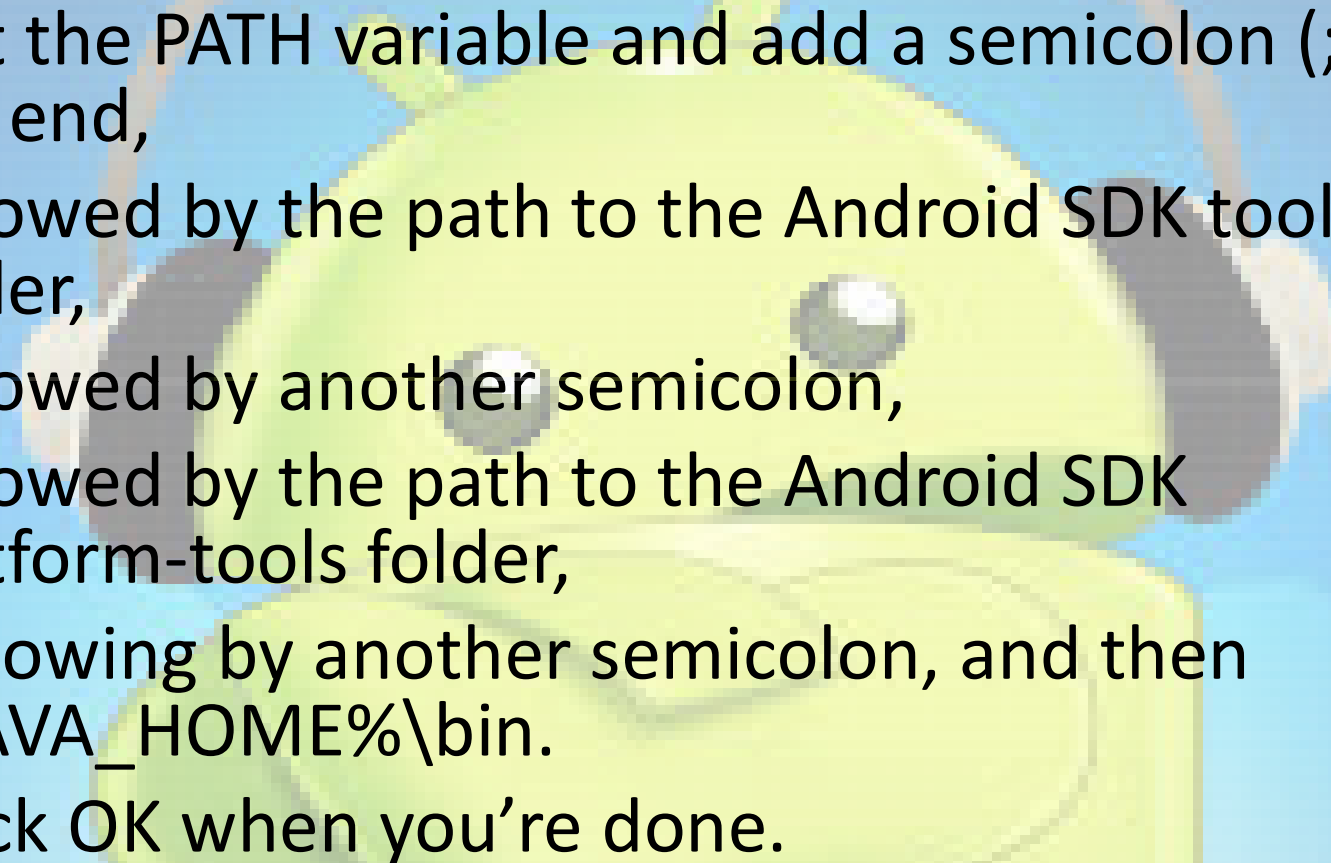
3. Downloading the Android SDK

- You can download the Android SDK from <http://developer.android.com/sdk>



4. Updating Your PATH Environment Variable

- The Android SDK comes with a tools directory that you want to have in your PATH.
- You also need in your PATH the platform-tools directory you just installed

- 
- Edit the PATH variable and add a semicolon (;) on the end,
 - followed by the path to the Android SDK tools folder,
 - followed by another semicolon,
 - followed by the path to the Android SDK platform-tools folder,
 - followed by another semicolon, and then %JAVA_HOME%\bin.
 - Click OK when you're done.

5. Installing Android Development Tools (ADT)

- To get started, launch the Eclipse IDE and follow these steps:
- **1. Select Help ➤ Install New Software.**
- **2. Select the Work With field, type in <https://dl-ssl.google.com/android/eclipse/> and press Enter.**

5. Installing Android Development Tools (ADT)

- **3. You should see an entry named Developer Tools with four child nodes:**
- Android DDMS, Android Development Tools, Android Hierarchy Viewer, and Android Traceview. Select the parent node Developer Tools, make sure the child nodes are also selected, and click the Next button.
- The versions you see may be newer than these, and that's okay

5. Installing Android Development Tools (ADT)

- 4. Eclipse asks you to verify the tools to install. Click Next
- 5. You're asked to review the licenses for ADT as well as for the tools required to install ADT. Review the licenses, click "I accept," and then click the Finish button.

A green Android robot is shown from the chest up, wearing a pair of brown headphones. The robot has two small antennae on its head and two circular eyes. The background is a solid light blue color. The text "LEARNING THE FUNDAMENTAL COMPONENTS" is overlaid on the lower left portion of the robot's body.

**LEARNING THE FUNDAMENTAL
COMPONENTS**

1. View

- *Views are user interface (UI) elements that form the basic building blocks of a user interface.*
- A view can be a button, a label, a text field, or many other UI elements
- Views are also used as containers for views, which means there's usually a hierarchy of views in the UI.

2. Activity

- *An activity is a UI concept that usually represents a single screen in your application*
- something that helps the user do one thing, which could be viewing data, creating data, or editing data

3. Fragment

- When a screen is large, it becomes difficult to manage all of its functionality in a single activity.
- *Fragments are like sub-activities, and an activity can display one or more fragments on the screen at the same time.*
- When a screen is small, an activity is more likely to contain just one fragment, and that fragment can be the same one used within larger screens.

4. Intent

An intent generically defines an “intention” to do some work

- Broadcast a message.
- Start a service.
- Launch an activity.
- Display a web page or a list of contacts.
- Dial a phone number or answer a phone call

4. Intent

- Intents can be explicit or implicit.
- to display a URL, the system decides what component will fulfill the intention.
- Intents loosely couple the action and action handler.

5. Content Provider

- Data sharing
- Android provide standard mechanism for applications to share data
- Through content providers, you can expose your data and have your applications use data from other applications.

6. Service

- *Services in Android resemble services you see in Windows or other platforms*
- Android defines two types of services: local services and remote services
- Local services are components that are only accessible by the application that is hosting the service

6. Service

- remote services are services that are meant to be accessed remotely by other applications running on the device.
- An example of a service is a component that is used by an e-mail application to poll for new messages.

7. AndroidManifest.xml

- defines the contents and behavior of your application.
- example, it lists your application's activities and services, along with the permissions and features the application needs to run.



8. Android Virtual Devices

- An Android Virtual Device (AVD) allows developers to test their applications without hooking up an actual Android device
- AVDs can be created in various configurations to emulate different types of real devices.

