CSIS 3175 – INTRODUCTION TO MOBILE DEVELOPMENT

Padmapriya Arasanipalai Kandhadai (Priya)

1-2

Course Overview

Prerequisites:	CSIS 1275
Instructor:	Padmapriya Arasanipalai Kandhadai (Priya)
Email:	kandhadaip@douglascollege.ca
Office Location:	NW-North Bldg. Room N5107
Class Time(s):	Wed: 9:30AM-12:20PM
Office Location:	NW-North Bldg. Room N4360B
Office Hours:	W: 12:30-1:00PM; Th: 12:30-1:30PM

1-3

Course Description

- This course introduces students to mobile device development technology concepts used in business.
- Students will learn how to build applications and services to solve business problems.
- Topics covered include different mobile platforms, tools for development, some aspects of user interface, and user experience design.

1-4

Course materials

- **Recommended Text:**
  - Android Boot Camp for Developers Using Java®: A Guide to Creating Your First Android Apps, 3rd Edition Corinne Hoisington Central Virginia Community College ISBN-10: 1305857992 | ISBN-13: 9781305857995
- **Online course materials**
  - All instructor materials, resources and communications such as announcements, lecture slides and other such documents will be shared through the Douglas College Blackboard Community.
  - All assessments will be created through Blackboard and will have to be submitted through Blackboard. [No exceptions will be made]

1-5

Course content

- Installing the Android Studio SDK
- Introduction to Android and the market
- Android UI
- User input
- Implementing icons and themes
- Variables
- Control structures
- Lists and arrays
- Implementing audio, graphics and animation
- Fragments and layouts
- Text file input and output
- SQLite on mobile devices

1-6

Evaluations and Assessments

Assessment	Percentage
Project	30%
Midterm Examination	30%
Final Examination	40%
Total Grade	100%

Final course grade: To receive a non-zero grade in the course, **a student must attempt and hand in for marking at least 70% of the assessments; and must attend the FINAL Exam**.
Final exam schedule: Check the final exam schedule for potential scheduling conflicts ** don't make travel arrangements until final exam schedule is released**

Course overview - policies

- ACADEMIC INTEGRITY
  - Cannot stress enough: ZERO tolerance for cheating, plagiarism, dishonesty
  - Academic Fraud: any impersonation
  - CHEATING: DO NOT copy or attempt to copy any material from any student, or any other source during exams, quizzes or assignments
  - PLAGIARISM: submitting other's work as own work; self-plagiarism
- WHAT HAPPENS
  - All parties will be penalized (so be aware that if you share your materials, you will be penalized)
  - Will be taken straight to the dean
  - Will get a zero in assessment, and will be formally investigated
  - **I use several monitoring software applications and tools to monitor cheating**

1-8

Course overview - policies

- Timeliness:
  - Students are expected to be here at the start of the class
- Exams:
  - In class as per schedule
  - No exceptions – will get a 0 if you miss it
  - No make up exams and quizzes
- Projects:
  - Must be done in groups of 2 or 3
  - If you decide to do it alone, need prior permission (will be evaluated in the same way as groups)

1-9

Course overview - policies

- Illness and other unavoidable circumstances:
  - Medical and supporting documentation
- Preparation, Attendance and Participation:
- Phones and wireless devices:
- Emailing your instructor:
  - Check the syllabus, blackboard announcements before emailing me

1-10

Course overview - policies

- **Douglas College Calendar, Policies and Procedures**
  - Students are responsible for being familiar with the information contained in the Douglas College Calendar and policies and procedures relating to appeals, petitions and formal complaints, sexual and personal harassment, standards of conduct, violence and academic honesty.
  - College policies are available at
- **Academic Integrity**
  - Cannot stress enough: ZERO tolerance for cheating, plagiarism, dishonesty

Course outline/schedule overview

- Registered students: can see tentative course outline on blackboard
- Waitlisted students: email me to get a copy of the course outline
  - kandhadaip@douglascollege.ca
  - Subject line should say waitlisted for 3175_001 – course outline request

Questions

Android Boot Camp for Developers Using Java, 3E

Chapter 1: Voilà! Meet the Android

Objectives

- Understand the market for Android applications
- Identify the role of the Android device in the mobile market
- Describe the features of the Android phone
- Identify which languages are used in Android development
- Describe the role of Google Play in the mobile marketplace

Objectives

- Create an Android project using Android Studio
- Explain the role of the Android project view
- Specify the use of layout and widget controls in the user interface
- Execute an Android application on an emulator
- Open a saved Android project in Android Studio

Meet the Android

- **Apps**
  - Mobile applications created for smartphones
- **Open-Source operating system**
  - No company or individual defines the features or direction of the development
- **Open Handset Alliance**
  - 80 firms that develop standards for mobile devices

Meet the Android (continued)

- **Android Phone Device**
  - Most popular are Moto X, Galaxy, Droid, Xperia, OnePlus, Nexus and HTC One
  - OS also powers tablets, readers, MP4 players and Internet TVs
- **Emulator**
  - Duplicates how the app looks and feels on a particular device

Figure 1-2 Android models

Meet the Android

Feature	Description
3-D graphics	The interface can support 3-D graphics for a 3-D interactive game experience or 3-D image rendering.
Facial recognition	Android provides this high-level feature for automatically identifying or verifying a person's face from a digital image or a video frame.
Front- and rear-facing camera	Android phones can use either a front- or rear-facing camera, allowing developers to create applications involving video calling.
Multiple language support	Android supports multiple human languages.
On-screen keyboard	The on-screen keyboard offers suggestions for spelling corrections as well as options for completing words you start typing. The on-screen keyboard also supports voice input.
Power management	Android identifies programs running in the background using memory and processor resources. You can close those apps to free up the phone's processor memory, extending the battery power. For optimized gaming, Android supports the use of a gyroscope, gravity and barometric sensors, linear acceleration, and rotation vector, which provide game developers highly sensitive and responsive controls.
Voice-based recognition	Android recognizes voice actions for calling, texting, and navigating with the phone.
Wi-Fi Internet tethering	Android supports tethering, which allows a phone to be used as a wireless or wired hot spot that other devices can use to connect to the Internet.

Table 1-1 Android platform features

Meet the Android (continued)

- **Writing Android Apps**
  - **Java** is an **Object-oriented programming language** patterned after the C++ language
- **Android Studio**
  - An integrated development environment (IDE) for building and integrating application development tools and open-source projects.
  - Android Studio IDE is exclusively dedicated to the purpose of creating Android applications
  - Includes **the Android Software Development Kit (SDK)**
  - **XML** is used to assist in the layout of the Android emulator

Meet the Android

- **Android Emulator**
  - Design, develop, prototype, and test Android apps without using a physical device
  - Mimics almost every feature of a real Android handset, except placing phone calls

Version Name	Release Date
1.0 First version	September 2008
1.5 Cupcake	April 2009
1.6 Donut	September 2009
2.0 Éclair	October 2009
2.2 Froyo (Frozen Yogurt)	May 2010
2.3 Gingerbread	December 2010
3.0 Honeycomb	February 2011
4.0 Ice Cream Sandwich	May 2011
4.1 Jelly Bean	July 2012
4.4 Kit Kat	October 2013
5.0 Lollipop	November 2014

Table 1-2 Android version history

Meet the Android

Figure 1-4 Android Studio live layout mode

Getting Oriented with Market Deployment

- **Getting Oriented with Market Deployment**
  - Platform consists of the Android OS, application development tools, and a marketplace Apps are compiled into package files with an .apk extension
  - **Google Play** (http://play.google.com) sells and deploys all apps
  - Programs must meet minimum standards
  - Apps are free or paid (70/30 split between developer and wireless carrier)
  - Also sold through Amazon (amazon.com/appstore) and iTunes (both charge a $99 registration fee)

Opening Android Studio to Create a New Project

- Download and install the Android Studio from https://developer.android.com/sdk/index.html
- You'll need an Application name, Company domain, package name, Project location, Form factor, Minimum SDK, Activity and activity name

- **Creating the Hello World Project**
  - Open the Android Studio program
  - Create a new project, name it, supply a company domain and location

Opening Android Studio to Create a New Project

Figure 1-6 Android Studio dialog box

- This is Version 2.2.3
- Lab computers have 2.3.3
- Android Studio 2.3.3 and 2.2.3 versions have slight differences in interface but basic functionality is broadly similar

Opening Android Studio to Create a New Project (cont'd)

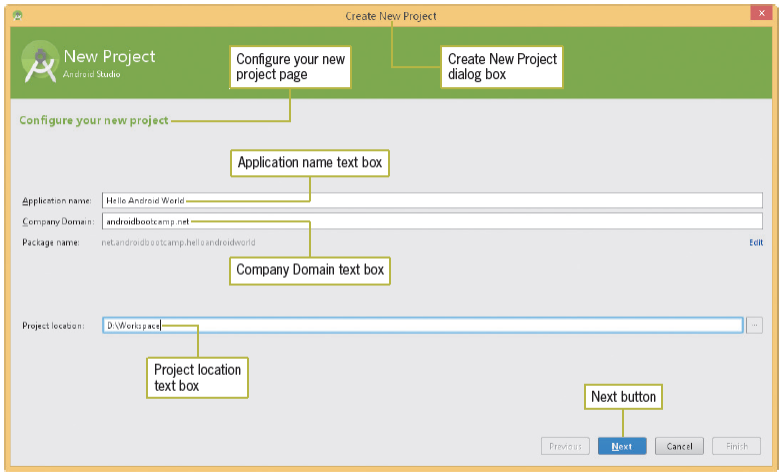


Figure 1-7 Configure your new project

Opening Android Studio to Create a New Project (cont'd)

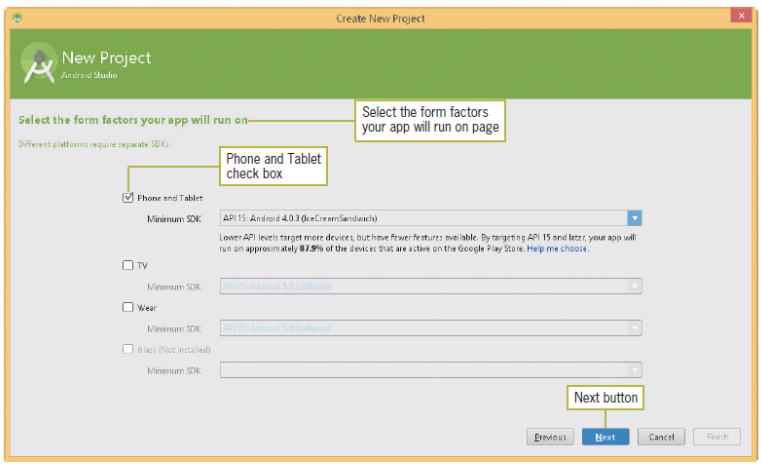


Figure 1-8 Form factors

Opening Android Studio to Create a New Project (cont'd)

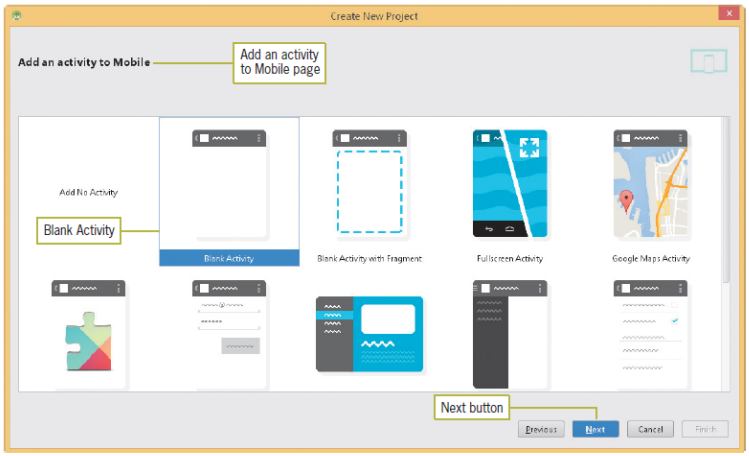


Figure 1-9 Add a blank Activity

Opening Android Studio to Create a New Project (cont'd)

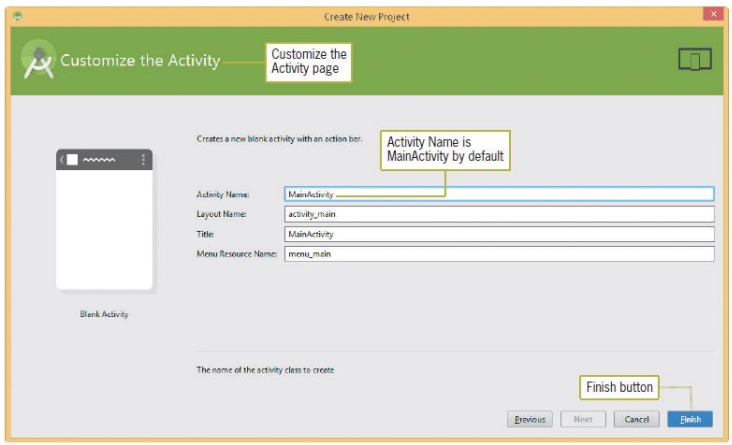


Figure 1-10 Creating MainActivity

Opening Android Studio to Create a New Project (cont'd)

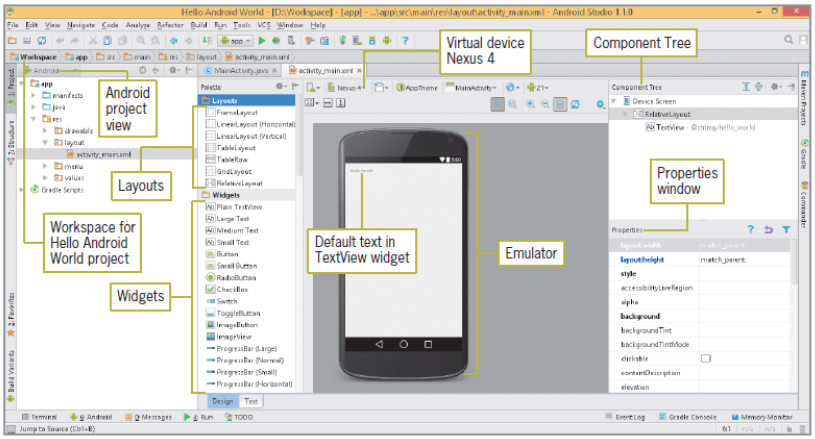


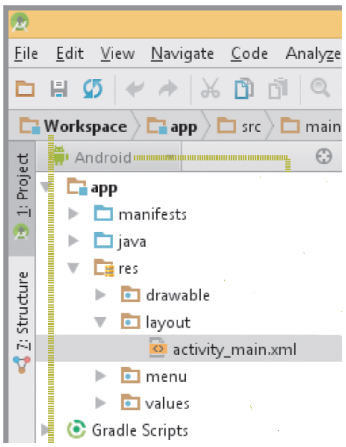
Figure 1-11 Android Studio user interface

Building the User Interface

- Must be intuitive
- Interface must not distract from functionality
- Java code or XML layout files are needed
  - XML method is preferred
  - Can design interface without writing large amounts of code

Taking a Tour of the Android Project View

- **Java folder** – contains Java source code
- **Res folder** – contains images, music, and video
- **Manifests folder** contains the **Android Manifest.xml** – which contains information about the application that Android needs to run



Designing the User Interface Layout within the Virtual Device

- **Layout** – a container that holds as many widgets as needed
- **Widget**– a single element on the screen (Button, Text Box, etc.)
- The **Properties pane** contains the properties of the currently active app project or object
- Android Studio displays an emulator configuration for design and layout purposes called **the Android Virtual Device (AVD)**

Designing the User Interface Layout within the Virtual Device

- Step 1: Tap or click 'the virtual device to render the layout with' button (the emulator button) directly to the right of the Palette on the activity_main.xml tab, and then tap or click Nexus 5 (5.0", 1080 x 1920, xxhdpi).

The Nexus 5 emulator is displayed on the activity_main.xml tab. Note that Android placed a default TextView control in the emulator (Figure 1-12).

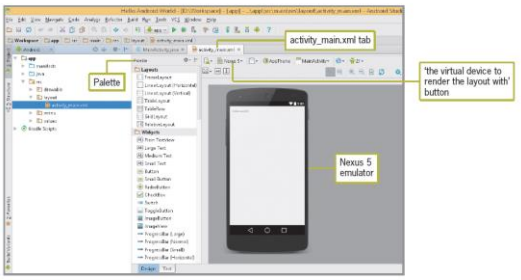


Figure 1-12 Nexus 5 Android Virtual Device

Designing the User Interface Layout within the Virtual Device

- Step 2: In the emulator, select the default TextView control, which reads Hello World!

The default TextView control is selected and displayed in a blue selection box (Figure 1-13).

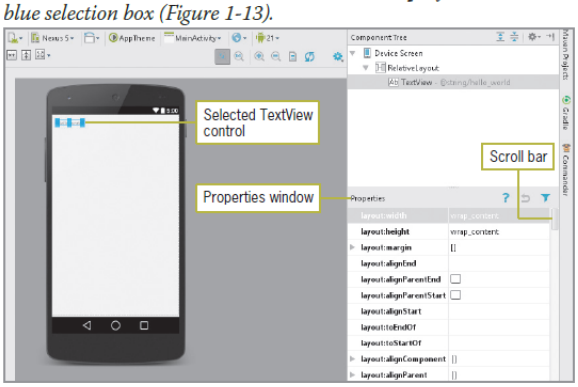


Figure 1-13 Selected TextView control

Designing the User Interface Layout within the Virtual Device

- Step 3: Scroll down the Properties pane to display the text property

The text property of the default TextView control is displayed (Figure 1-14). The Component Tree indicates that the TextView control is selected. The ***Component Tree*** *shows you the structure of your emulator layout.*

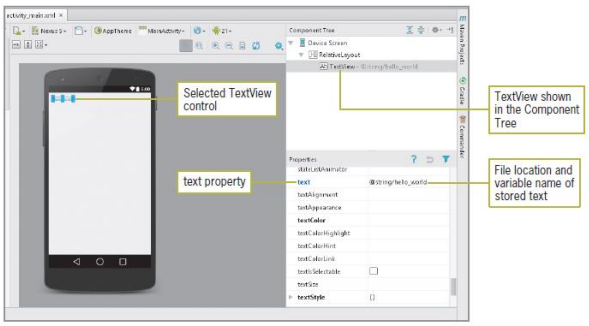


Figure 1-14 text property of the TextView control

Modifying the Text in the TextView Control

- Step 1: In the Android project view, tap or click the expand arrow for the values subfolder in the res folder to expand the folder's contents

Three default string text values are displayed in strings.xml tab in the Project window (Figure 1-15).

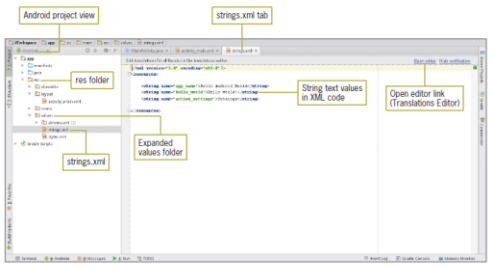


Figure 1-15 strings.xml default values

- Double-tap or double-click the strings.xml file to display the strings.xml tab in the Project window

Modifying the Text in the TextView Control

- Step 2: Tap or click the Open editor link on the right side of the Project window to display the Translations Editor tab
- Tap or click the Default Value (Hello world!) to the right of the hello_world Key column in the Translations Editor
- At the bottom of the Translations Editor tab, type **Hello World – My First Android App** in the Default Value text box

The hello_world default value is changed to Hello World – My First Android App in the Translations Editor (Figure 1-16).

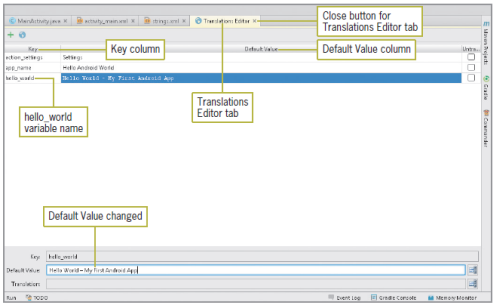


Figure 1-16 Translations Editor

Modifying the Text in the TextView Control

- Step 3: Close the Translations Editor window by tapping or clicking the Close button on the Translations Editor tab

The hello_world text is modified in strings.xml (Figure 1-17).

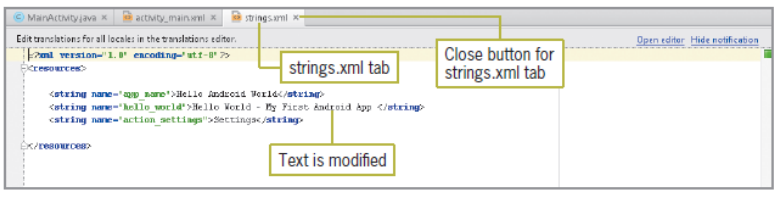


Figure 1-17 Modified strings.xml

Modifying the Text in the TextView Control

- Step 4: Close the strings.xml tab by tapping or clicking its Close button

The TextView control displays the modified text in the emulator (Figure 1-18).

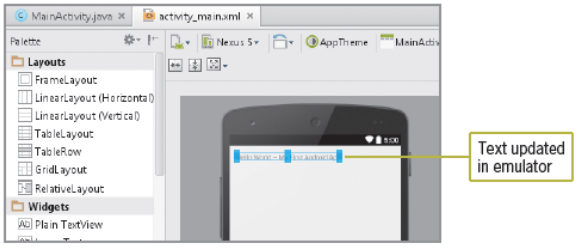


Figure 1-18 Emulator with TextView control

Testing the Application in the Emulator

- Step 1: Tap or click the Run 'app' button on the toolbar

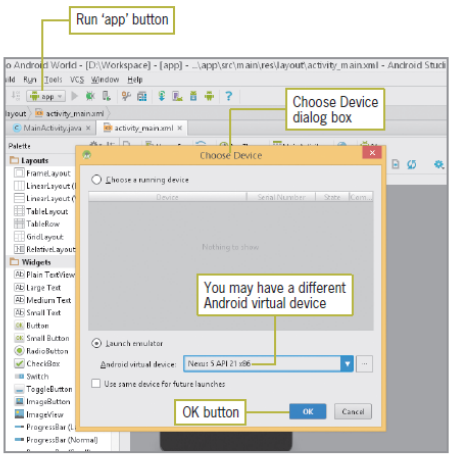


Figure 1-19 Choose Device dialog box

Testing the Application in the Emulator

- Step 2: If necessary, select Nexus 5 API 21 x86 (or a recent version emulator) in the Android virtual device list
  - Tap or click the OK button in the Choose Device dialog box

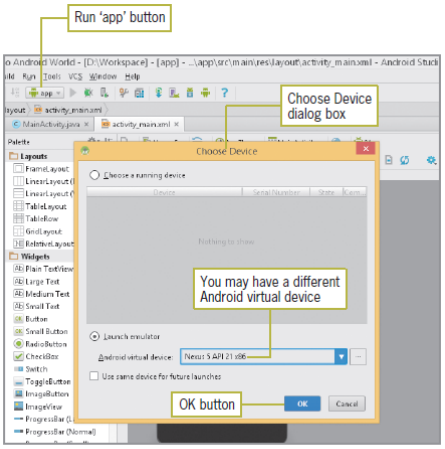


Figure 1-19 Choose Device dialog box

Testing the Application in the Emulator

- Step 3: Tap or click the lock icon and slide it upward toward the time in the vertical center of the screen to unlock the virtual device

  (If you are using a Mac, drag the lock icon until it changes to an unlock icon)

Figure 1-20 Android emulator

Figure 1-21 Emulator unlocked

Testing the Application in the Emulator

- Step 4: Tap or click the Run 'app' button on the toolbar again to launch the code into the opened emulator
- Step 5: Tap or click the OK button in the Choose Device dialog box

The Choose Device dialog opens again displaying the running emulator option (Figure 1-22).

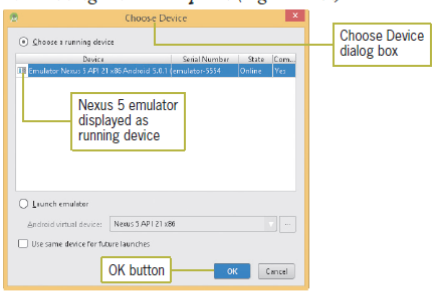


Figure 1-22 Choose Device dialog box displays the running emulator

- Step 6 – Close the emulator and the project

Opening a Saved App in Android Studio

- Step 1: Open Android Studio
  - Tap or click the Hello Android World project in the Recent Projects column (If the project is not listed in the Recent Projects, tap or click Open an existing Android Project in the Quick Start column and navigate to the path of the saved project and tap or click the OK button

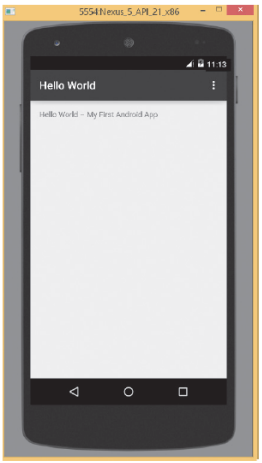


Figure 1-23 Hello Android World app displayed in emulator

Summary

- Android OS is released under a full open-source license for free
- Android OS powers all types of mobile devices
- To write apps, you can use Android Studio, a dedicated development environment for building Android applications, using Java
- The Android emulator lets you design, develop prototype, and test Android applications
- The Android platform consists of the Android OS, the Android application development platform, and Google Play - the Android marketplace

Summary

- Android supports both Java code and XML layout files, although XML is preferred
- The Android project view on the left side of the Android Studio program window contains the folders for an Android project
- To design a user interface for an Android app, you can create a layout containing widgets (objects)
- The text property can be updated using the Translations Editor
- You can test applications in the Android emulator to make sure they run correctly