MOBILE APPLICATION DEVELOPMENT

PGDCA 203

BLOCK 1:
BASICS OF ANDROID APPLICATION

Dr. Babasaheb Ambedkar Open University
Ahmedabad

MOBILE APPLICATION DEVELOPMENT

KNOWLEDGE MANAGEMENT & RESEARCH ORGANIZATION

Knowledge Management and
Research Organization
Pune

Editorial Panel

Author
Er. Nishit Mathur

Language Editor
Prof. Jaipal Gaikwad

Graphic and Creative Panel
Ms. K. Jamdal
Ms. Lata Dawange
Ms. Pinaz Driver
Ms. Ashwini Wankhede
Mr. Kiran Shinde
Mr. Prashant Tikone
Mr. Akshay Mirajkar

Copyright © 2015 Knowledge Management and Research Organization.
All rights reserved. No part of this book may be reproduced, transmitted or utilized in any form or by a means, electronic or mechanical, including photocopying, recording or by any information storage or retrieval system without written permission from us.

Acknowledgment
Every attempt has been made to trace the copyright holders of material reproduced in this book. Should an infringement have occurred, we apologize for the same and will be pleased to make necessary correction/amendment in future edition of this book.
The content is developed by taking reference of online and print publications that are mentioned in Bibliography. The content developed represents the breadth of research excellence in this multidisciplinary academic field. Some of the information, illustrations and examples are taken "as is" and as available in the references mentioned in Bibliography for academic purpose and better understanding by learner.'

ROLE OF SELF INSTRUCTIONAL MATERIAL IN DISTANCE LEARNING

The need to plan effective instruction is imperative for a successful distance teaching repertoire. This is due to the fact that the instructional designer, the tutor, the author (s) and the student are often separated by distance and may never meet in person. This is an increasingly common scenario in distance education instruction. As much as possible, teaching by distance should stimulate the student's intellectual involvement and contain all the necessary learning instructional activities that are capable of guiding the student through the course objectives. Therefore, the course / self-instructional material are completely equipped with everything that the syllabus prescribes.

To ensure effective instruction, a number of instructional design ideas are used and these help students to acquire knowledge, intellectual skills, motor skills and necessary attitudinal changes. In this respect, students' assessment and course evaluation are incorporated in the text.

The nature of instructional activities used in distance education self-instructional materials depends on the domain of learning that they reinforce in the text, that is, the cognitive, psychomotor and affective. These are further interpreted in the acquisition of knowledge, intellectual skills and motor skills. Students may be encouraged to gain, apply and communicate (orally or in writing) the knowledge acquired. Intellectual-skills objectives may be met by designing instructions that make use of students' prior knowledge and experiences in the discourse as the foundation on which newly acquired knowledge is built.

The provision of exercises in the form of assignments, projects and tutorial feedback is necessary. Instructional activities that teach motor skills need to be graphically demonstrated and the correct practices provided during tutorials. Instructional activities for inculcating change in attitude and behavior should create interest and demonstrate need and benefits gained by adopting the required change. Information on the adoption and procedures for practice of new attitudes may then be introduced.

Teaching and learning at a distance eliminates interactive communication cues, such as pauses, intonation and gestures, associated with the face-to-face method of teaching. This is particularly so with the exclusive use of print media. Instructional activities built into the instructional repertoire provide this missing interaction between the student and the teacher. Therefore, the use of instructional activities to affect better distance teaching is not optional, but mandatory.

Our team of successful writers and authors has tried to reduce this.

Divide and to bring this Self Instructional Material as the best teaching and communication tool. Instructional activities are varied in order to assess the different facets of the domains of learning.

Distance education teaching repertoire involves extensive use of self-instructional materials, be they print or otherwise. These materials are designed to achieve certain pre-determined learning outcomes, namely goals and objectives that are contained in an instructional plan. Since the teaching process is affected over a distance, there is need to ensure that students actively participate in their learning by performing specific tasks that help them to understand the relevant concepts. Therefore, a set of exercises is built into the teaching repertoire in order to link what students and tutors do in the framework of the course outline. These could be in the form of students' assignments, a research project or a science practical exercise. Examples of instructional activities in distance education are too numerous to list. Instructional activities, when used in this context, help to motivate students, guide and measure students' performance (continuous assessment)

PREFACE

We have put in lots of hard work to make this book as user-friendly as possible, but we have not sacrificed quality. Experts were involved in preparing the materials. However, concepts are explained in easy language for you. We have included may tables and examples for easy understanding.

We sincerely hope this book will help you in every way you expect.

All the best for your studies from our team!

Mobile Application Development

Contents

BLOCK 3: ANDROID NETWORKING AND DEVELOPMENT

UNIT 1 **ADVANCED TOPICS - I**

Android Networking, Web and Telephony API, Search, Location and Mapping, Sensors, NFC, Speech, Gestures, and Accessibility,

UNIT 2 **ADVANCED TOPICS - II**

Communication, Identity, Sync, and Social Media, The Android Native Development Kit (NDK)

BLOCK 4: ANDROID APPLICATION PUBLISHING AND CONTENT PROVIDERS

UNIT 1 **MORE ON ANDROID**

Handling and Persisting Data, A Content Provider as a Facade for a RESTful Web Service, Using Content Providers

UNIT 2 **PUBLISHING ANDROID APPLICATION**

Deploying Android Application to the World, Selling your Android application

Dr. Babasaheb
Ambedkar
Open University

PGDCA-203

MOBILE APPLICATION DEVELOPMENT

BLOCK 1: BASICS OF ANDROID APPLICATION

BLOCK 1: BASICS OF ANDROID APPLICATION

Block Introduction

Android continues an open source as well as Linux-based operating system for mobile instruments comparable as smartphones along with tablet computers. Android had been constructed immediate the Open Handset collaboration, commanded by Google, along with external companies.

In this block, we will detail about the basic of Android application as open source Linux-based operating system with knowledge on its application. The block will focus on the study and concept of Android Studio IDE along with Google I/O 2013 and Android Studio. You will give an idea on application component on screen for interaction by users.

In this block, you will make to learn and understand about basic of Intent and services involved in Android applications. The concept related to designing of various Android applications with necessary tools is also explained to the students. The student will be demonstrated practically about various design pattern applied in designing of android applications.

Block Objective

After learning this block, you will be able to understand:

- About Android Platform
- Features of installation of Android Studio
- Idea about Java for Android
- Characteristics of Android Studio
- Framework for Well-Behaved Application
- Concept of Intent Filter

Block Structure

Unit 1: Introduction to Android, Tools and Basics

Unit 2: Android Application Design Essentials - I

UNIT 1: INTRODUCTION TO ANDROID, TOOLS AND BASICS

Unit Structure

1.0 Learning Objectives

After learning this unit, you will be able to understand:

- About Android Platform
- About Android Studio installation
- About Java for Android
- About Android Software Development

1.1 Introduction

Android continues an open source as well as Linux-based operating system for mobile instruments comparable as smartphones along with tablet computers. Android had been constructed immediate the Open Handset collaboration, commanded by Google, along with external companies. Android apps are transcribed in the Java computing jargon. The Android SDK appliances constitute your code—along with numerous data along with resource files—into an APK: an Android package, which continues an archive file with an .apk suffix. One APK file comprises all-inclusive the contents of an Android app also is the file that Android-powered appliances facilitate to install the app.

1.2 The Android Platform

Android endures an Operating System for mobile appliances constructed proximate Google, which continues developed upon Linux kernel. Android applies with Apple's iOS (for iPhone/iPad), RIM's Blackberry, Microsoft's Windows Phone, Symbian OS, along with abundant alien proprietary mobile OSes.

Android platform exists disinterred close-at-hand both numerals: a version identity (x.y) along with an API level integer (a moving character commences from 1, which exists applied in the Android Market/Google Play to identify new version).

The Android platform endures a platform for mobile appliances that facilitates a changed Linux kernel. The Android Platform had been commenced near the Open Handset coalition in November relevantly 2007. Furthermost approaches that bound on the Android platform are transcribed in the Java programming jargon.

Android continues an accessible development platform. Although, it exists not accessible in the perceive that all can avail while a version is underneath development. This is completely conclusive behind closed-doors at Google. Rather, the broadness of Android commences when its source code is authorized to the public following it exists completed. This means once it is released anyone interested can take the code and alter it as they see fit.

To develop an application for the platform, a developer mandates the Android SDK, which encompasses trowels along with APIs. To condense development course, Android developers denominationally encompass the SDK into graphical user IDEs (Integrated Development Environments). Beginners can additionally construct facilitate analogously the App Inventor, an application for developing Android apps that can be approached online.

Android contributes you everything you expect to develop best-in-class app experiences. It assigns you a individual application example that authorizes you contribute your apps widely to hundreds of millions of consumers across a broad range of devices—from phones to tablets as well as beyond. Android additionally assigns you tools for constructing apps that observe awesome additionally take advantage of the hardware abilities feasible on each appliance. It automatically adjusts your UI to look its better on each appliance, while assigning you as much control as you desire over and above your UI on contrasting device categorizes.

For instance, you can develop a single app binary that's optimized for dual phone along with tablet bearing factors. You admit your UI in lightweight sets of XML reserve, one set for parts of the UI that are dominant to all form factors as well as other sets for optimizing specific to phones or tablets. At runtime, Android exercises the correct resource sets based on its screen size, density, locale, and so on.

Check your progress 1

1. Android OS is based on__________.

   a. Windows

   b. Mac

   c. Linux

   d. None of these

2. Android is now a product of _________.

   a. Google

   b. Microsoft

   c. Apple

   d. None of these

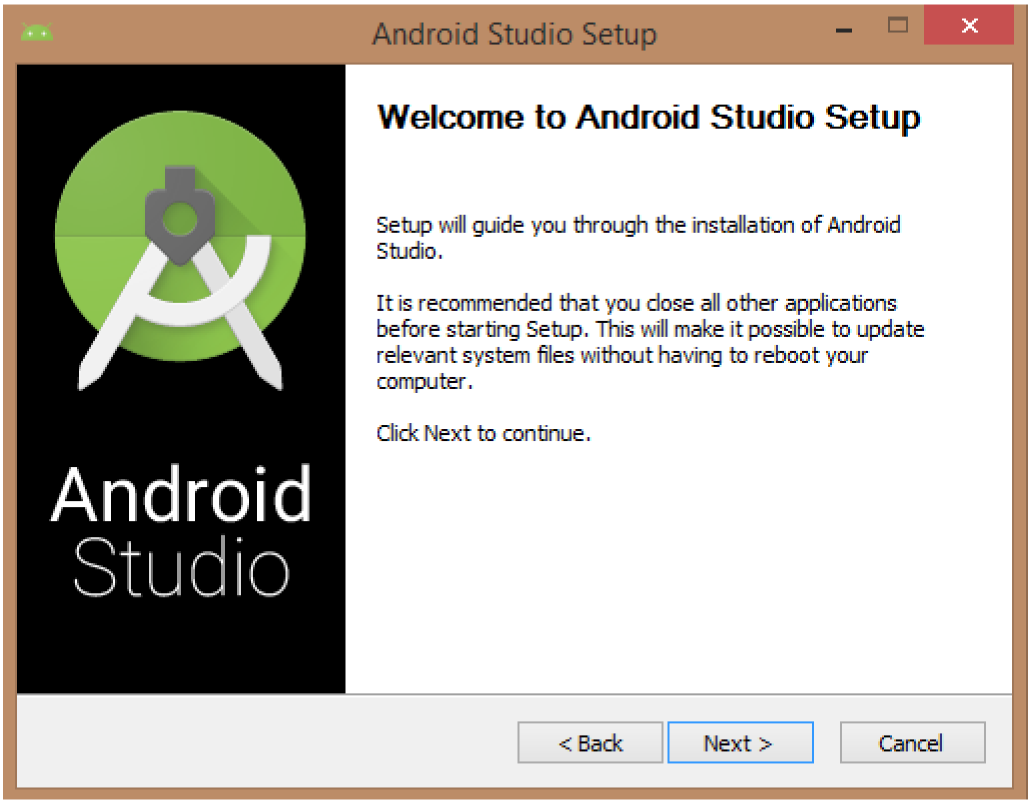
1.3 Installing Android Studio

Android Studio IDE subsists impelled at Google I/O 2013. Android Studio is based on Intellij IDEA which contrivances to exchange Eclipse + Android Developer Tools for Android creation. Also because Google itself continues pushing it, it may accumulate up with eclipse in due time.

Android Studio continues the official IDE for android application creation. It efforts based onIntelliJ IDEA, You can download the current transcription of android studio from Android Studio Download.

Installation

Start with Android Studio.exe. Before Android Studio, the computer should have Java JDK which if not can be installed taking references of Android environment setup



Welcome to Android Studio Setup

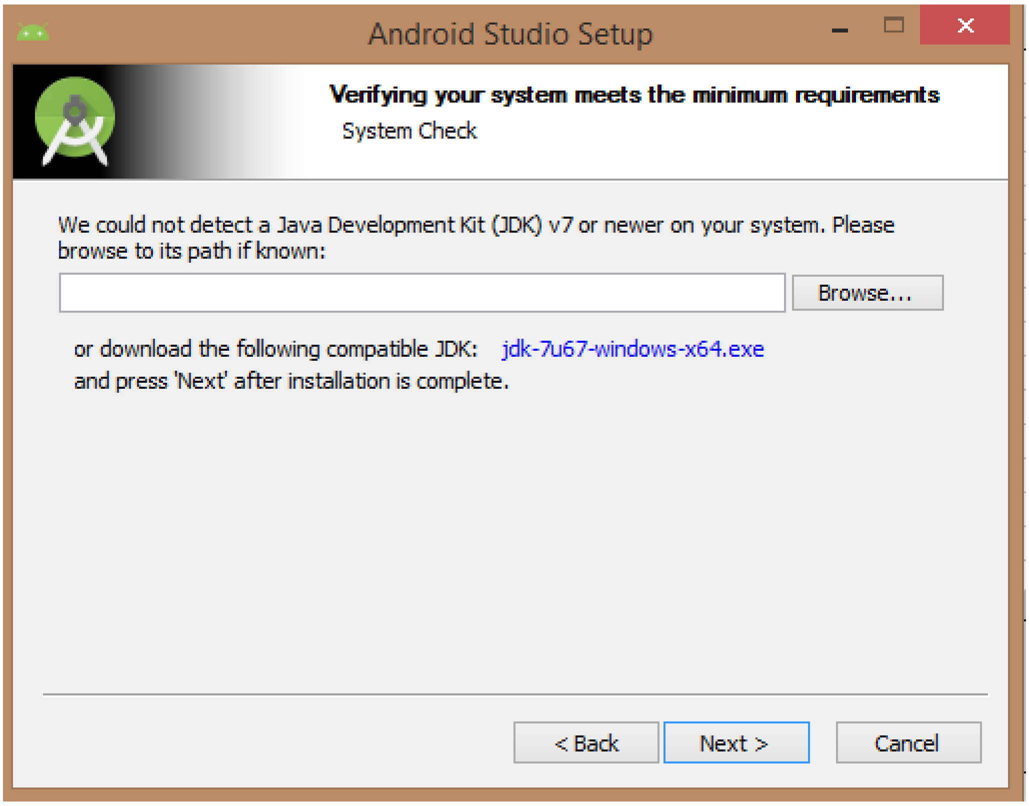
Setup will guide you through the installation of Android Studio.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Fig 1.1 Welcome screen

After launching Android Studio, show JDK5 path in android studio installer.



Android Studio Setup

Verifying your system meets the minimum requirements

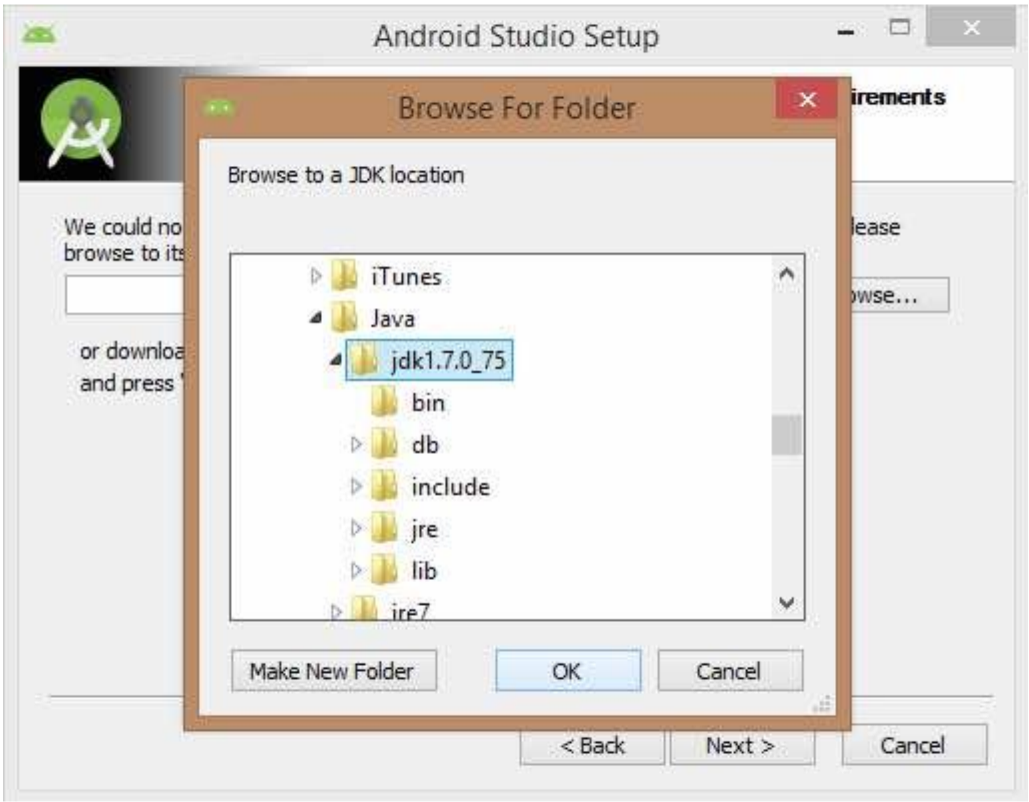
System Check

We could not detect a Java Development Kit (JDK) v7 or newer on your system. Please browse to its path if known:

or download the following compatible JDK: jdk-7u67-windows-x64.exe
and press 'Next' after installation is complete.

Fig 1.2 Android studio setup

Now initiate JDK to android SDK



Android Studio Setup

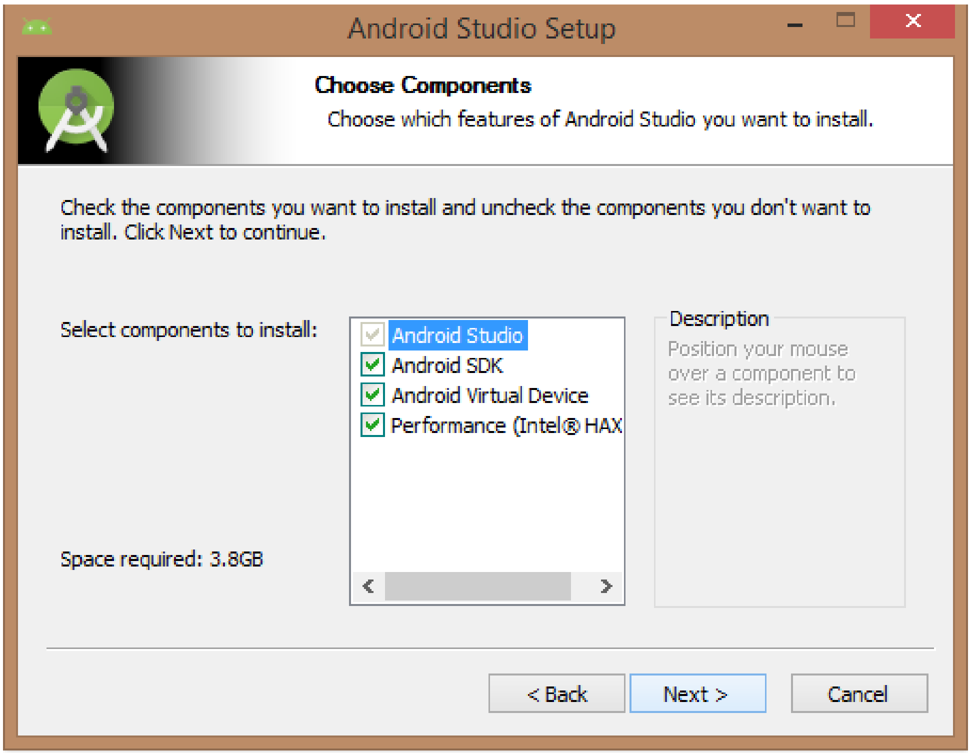
Browse For Folder

Browse to a JDK location

Fig 1.3 initiating JDK

Introduction to Android, Tools and Basics

After this, verify the components that frames applications a Android Studio, Android SDK, Android Virtual Machine and performance.



Android Studio Setup

Choose Components

Choose which features of Android Studio you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

☑ Android Studio
☑ Android SDK
☑ Android Virtual Device
☑ Performance (Intel® HAX

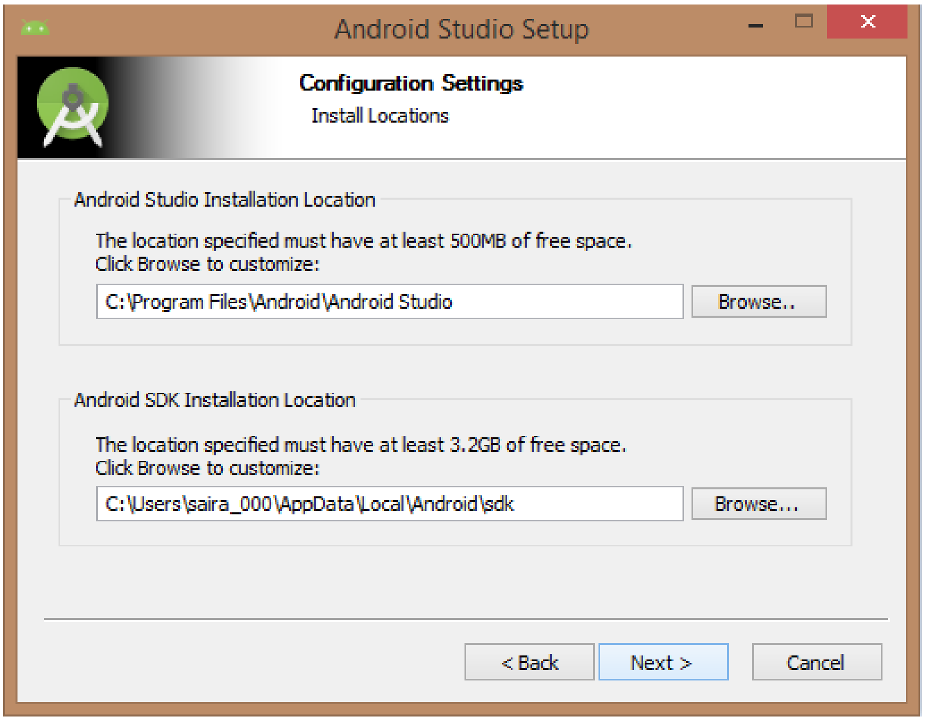
Description

Position your mouse over a component to see its description.

Space required: 3.8GB

Fig 1./4 Selecting components

Now you have to select the location of local machine path for Android studio and also for Android SDK as shown:



Android Studio Setup

Configuration Settings

Install Locations

Android Studio Installation Location

The location specified must have at least 500MB of free space.
Click Browse to customize:

C:\Program Files\Android\Android Studio

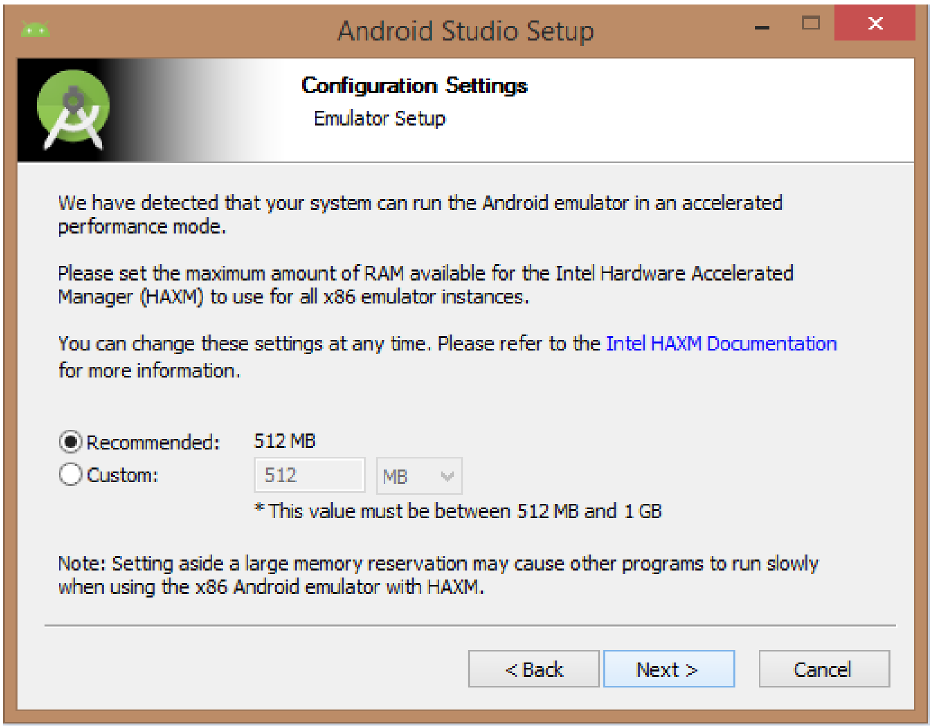
Android SDK Installation Location

The location specified must have at least 3.2GB of free space.
Click Browse to customize:

C:\Users\saira_000\AppData\Local\Android\sdk

Fig 1.5 Location

Now check for RAM. As the download needs minimum 512MB of RAM in the local machine.



Configuration Settings
Emulator Setup

We have detected that your system can run the Android emulator in an accelerated performance mode.

Please set the maximum amount of RAM available for the Intel Hardware Accelerated Manager (HAXM) to use for all x86 emulator instances.

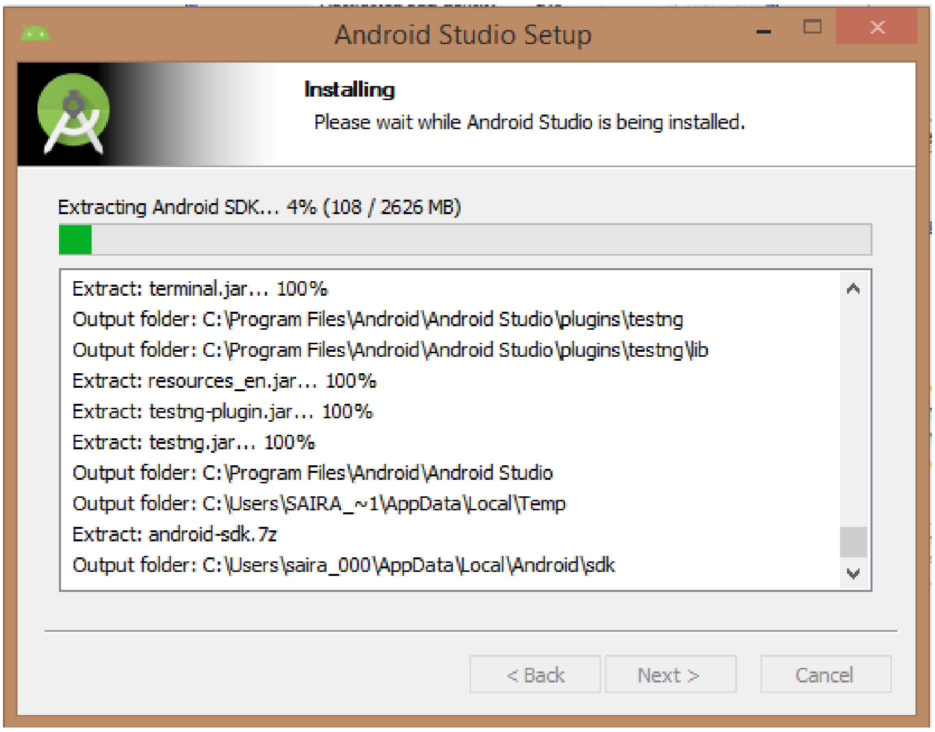
You can change these settings at any time. Please refer to the Intel HAXM Documentation for more information.

● Recommended: 512 MB
○ Custom: 512 MB
* This value must be between 512 MB and 1 GB

Note: Setting aside a large memory reservation may cause other programs to run slowly when using the x86 Android emulator with HAXM.

Fig 1.6 Configuration checking

Now finally extract SDK packages into local machine which requires about 2626MB of Hard disk space.



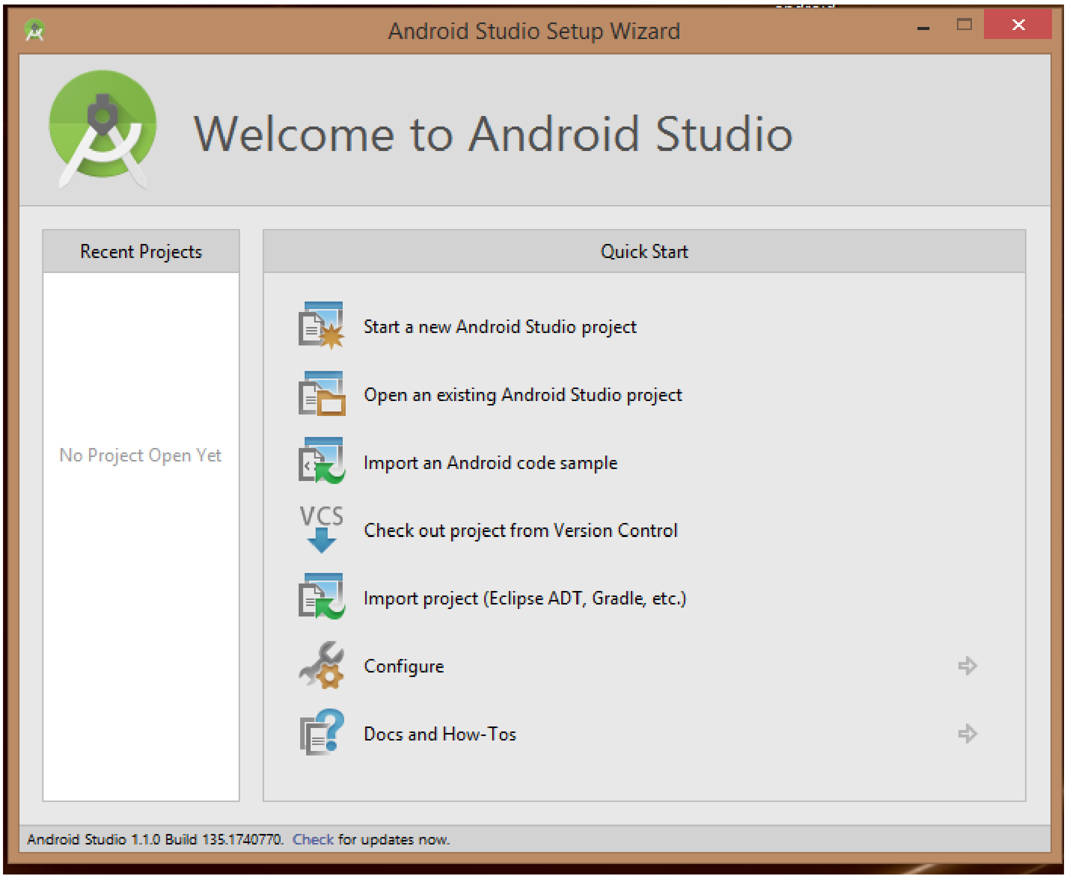
Installing
Please wait while Android Studio is being installed.

Extracting Android SDK... 4% (108 / 2626 MB)

```
Extract: terminal.jar... 100%
Output folder: C:\Program Files\Android\Android Studio\plugins\testng
Output folder: C:\Program Files\Android\Android Studio\plugins\testng\lib
Extract: resources_en.jar... 100%
Extract: testng-plugin.jar... 100%
Extract: testng.jar... 100%
Output folder: C:\Program Files\Android\Android Studio
Output folder: C:\Users\SAIRA_~1\AppData\Local\Temp
Extract: android-sdk.7z
Output folder: C:\Users\saira_000\AppData\Local\Android\sdk
```

Fig 1.7 Installation setup

Now after it, click on finish button so that the android studio project gets started with Welcome message as shown below



Android Studio Setup Wizard

Welcome to Android Studio

Recent Projects

No Project Open Yet

Quick Start

Start a new Android Studio project

Open an existing Android Studio project

Import an Android code sample

Check out project from Version Control

Import project (Eclipse ADT, Gradle, etc.)

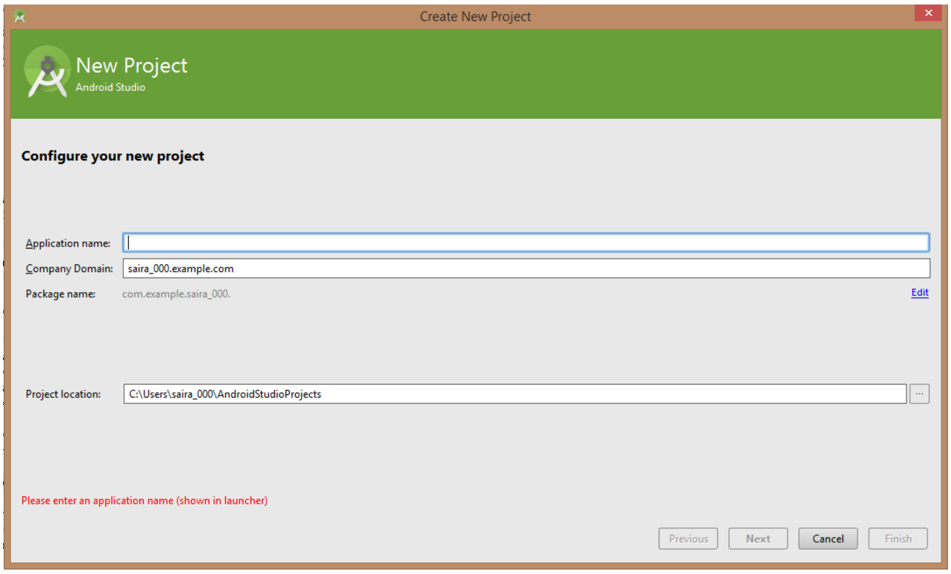
Configure

Docs and How-Tos

Android Studio 1.1.0 Build 135.1740770. Check for updates now.

Fig 1.8 Welcome to Android studio

Start your application development by calling start new android studio project. Here you need to specify Application name, package information and location of the project.



Create New Project

New Project
Android Studio

Configure your new project

Application name:

Company Domain: saira_000.example.com

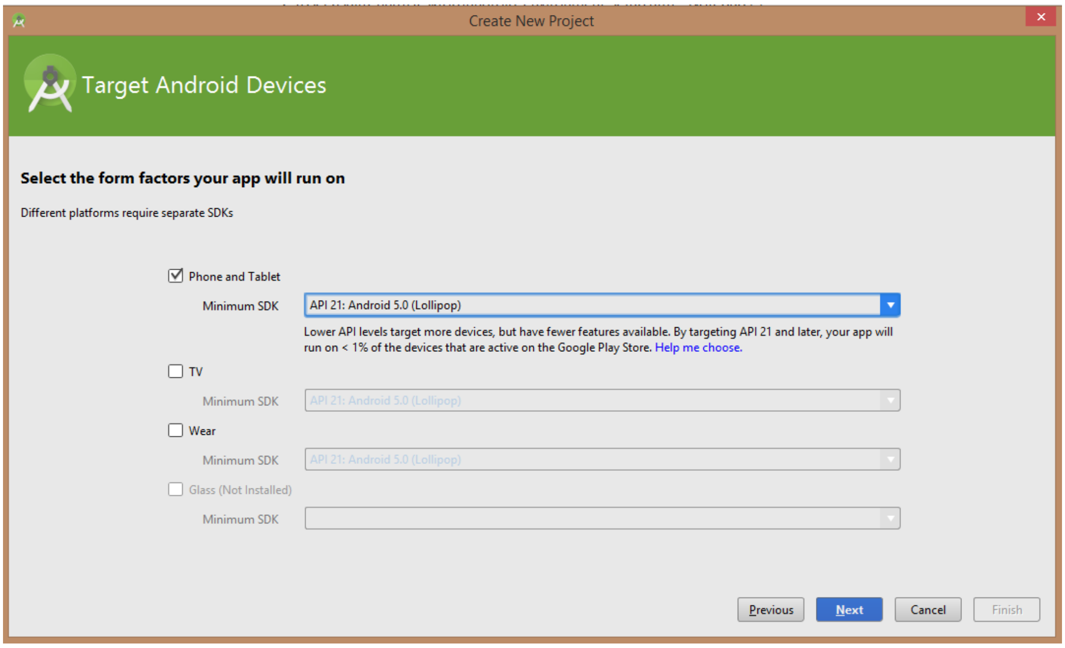
Package name: com.example.saira_000. Edit

Project location: C:\Users\saira_000\AndroidStudioProjects

Please enter an application name (shown in launcher)

Fig 1.9 Project Screen

Once all that is entered, select form factors for an application to run.



Create New Project

Target Android Devices

Select the form factors your app will run on

Different platforms require separate SDKs

☑ Phone and Tablet

Minimum SDK: API 21: Android 5.0 (Lollipop)

Lower API levels target more devices, but have fewer features available. By targeting API 21 and later, your app will run on < 1% of the devices that are active on the Google Play Store. Help me choose.

☐ TV

Minimum SDK: API 21: Android 5.0 (Lollipop)

☐ Wear

Minimum SDK: API 21: Android 5.0 (Lollipop)

☐ Glass (Not Installed)

Minimum SDK:

Fig 1.10 Target Android device

Further, select activity to mobile showing default layout for Applications

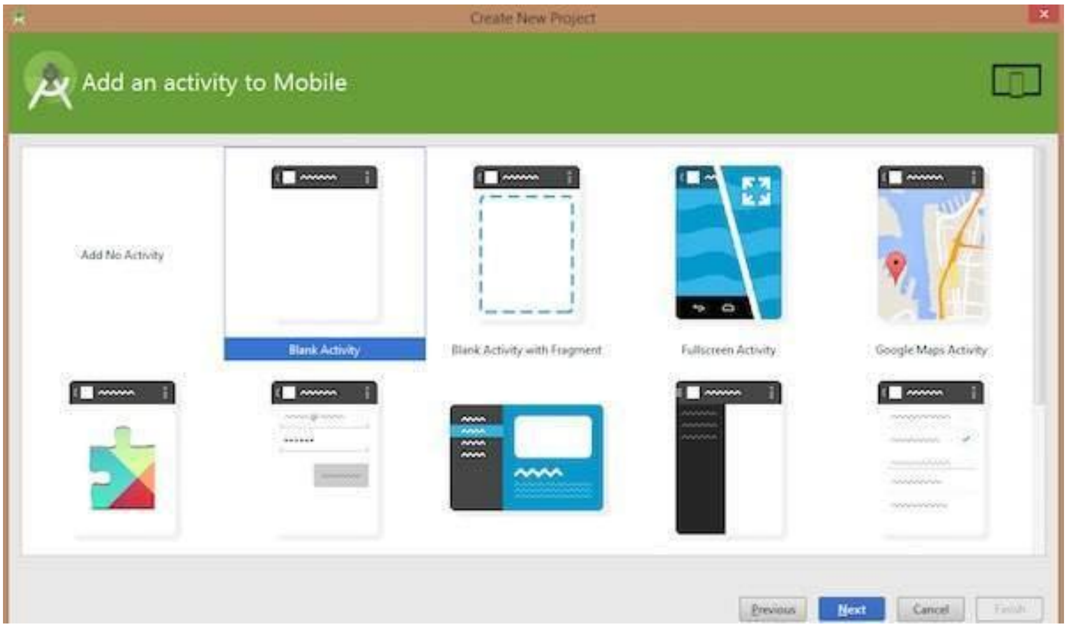


Fig 1.11 Adding Activity screen

Finally, open development tool to write application code.

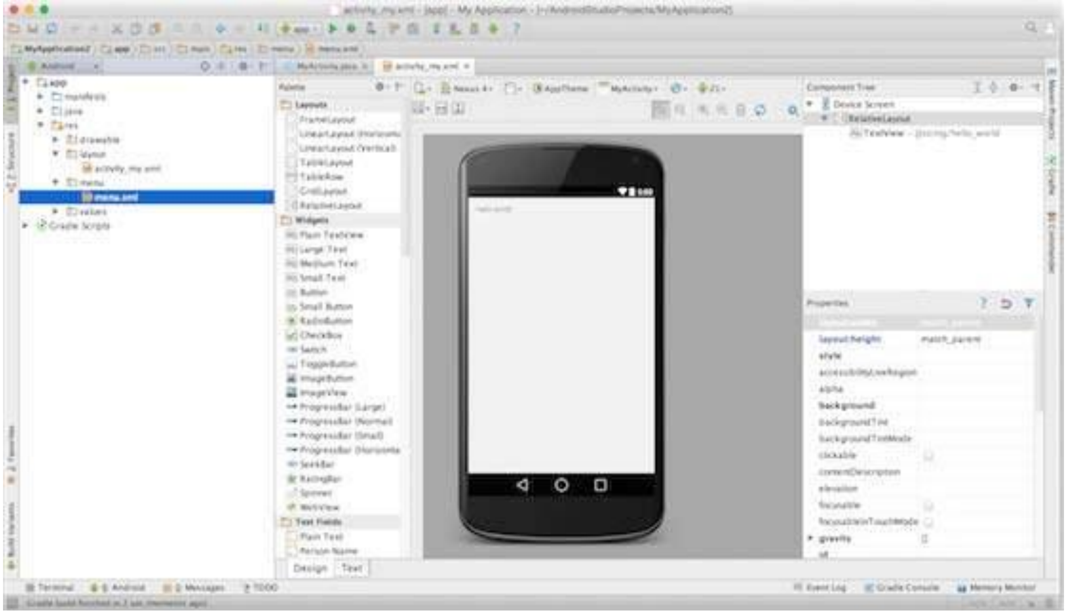
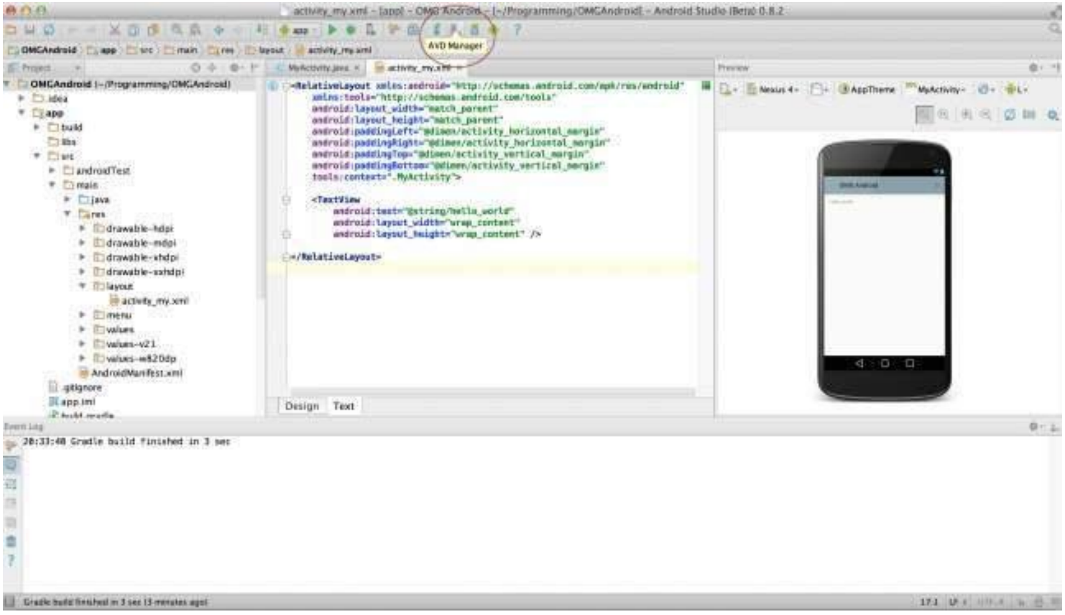


Fig 1.12

Creating Android Virtual Device

In order to check Android applications, you require virtual Android device. Now launch Android AVD Manager by choosing AVD_Manager icon:

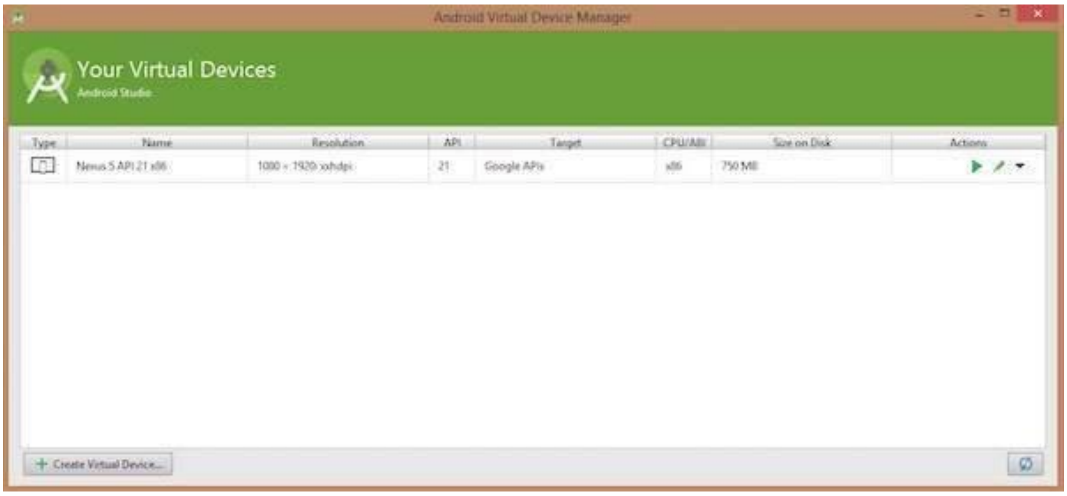


```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context=".MyActivity">

    <TextView
        android:text="@string/hello_world"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</RelativeLayout>
```

```
20:33:48 Gradle build finished in 3 sec
```

Fig 1.13 Testing of android application

Once the virtual device icon is selected, the default virtual devices appears on SDK and if not, then you have to create it by clicking on Create new Virtual device button:



Type	Name	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
	Nexus 5 API 21 x86	1080 × 1920: xxhdpi	21	Google APIs	x86	750 MB	

Fig 1.14 Virtual device

If your AVD is created successfully it means your environment is ready for Android application development.

Check your progress 2

1. AVD stands for_____________.
   a. Android Virtual Driver
   b. Android Virtual Device
   c. Application Virtual Device
   d. None of these
2. To install Android SDK on system , we need________________.
   a. Jdk 5 or above
   b. Java Runtime Environment (JRE) 6
   c. 512 MB RAM
   d. All of these

1.4 Java for Android

Java continues the elementary language for enduring Android apps. You desire to grasp Java for Android development. This orientation will educate you how to approximate in the Java Programming Language cantering on Java concepts that you will wish in order to construct Android apps. There are numerals of approaches to develop apps for Android appliances, but the felicitated method for most developers is to write native apps using Java and the Android SDK. Java for Android apps is both comparable as well as quite contrary from other types of Java applications.

Programming jargons, similar regular dialects, are contrary ways to broadcast to a computer how you desire it to behave. Programming languages apportion us to encourage a computer step-by-step how to contact data, assemble input from consumers, as well as demonstrate materials on a screen, among external things. Java's central arrays as well as core classes in the Java assemblages Framework, comparable as ArrayList along with HashMap), categorize associated operations as well as data into classes additionally interfaces (such as Java's primitive also user-defined categories, fields, approaches, collective parameters, and absolutions), customize the carriage of enduring classes via legacy along with polymorphism (imitative as subclassing as well as overriding virtual approaches). Learners' discretion conduct these Java constituents in the context of core Android elements (such as Activities as well as elementary UI components) by exercising normal tools (such as Android Studio as well as Git) desired to construct Java programs as well as useful Android apps.

Android doesn't conduct "pure" Java! This may cry out extraneous, due to when you match code from a conventional Java program to comparable code from an Android app, you'd scramble to see the contrast.

Furthermore writing as well as constructing an Android app will experience somewhat canonical to learned Java developers, the conventionality ends compactly when you construct along with run. The inference you'll consider yourself in uncharted domain is the approach Android triggers its apps in the course of the formation process.

Java's major approach endures its competence to "Write once, run everywhere". This jargon is marketed as the silver trigger to the costly mechanism of porting software from one platform to another.

This veritable marvel of software engineering is made possible thanks to what happens when a Java program compiles.

For the time being the fabrication approach for eminent other jargons, the compiler links additionally optimizes the program, furthermore that time it condenses it into Machine Code, which is a set of codes a computer can comprehend as well as develop when you run the program.

Furthermore development of machine code is expeditious, it’s bordered due to it searches the platform on which it plunges. If you ever astonished why a program composed for iOS platform doesn’t equitable work on Windows, this is one of inducts.

Java, in diverge, conducts something contrary; instead of decoding a program into machine code, the Java compiler interprets it an halfway form named Bytecode. It constructs a bunch of catchwords that are comparable to machine code, furthermore are targeted to run on a Virtual Machine (VM) instead of many distinguished arrangement.

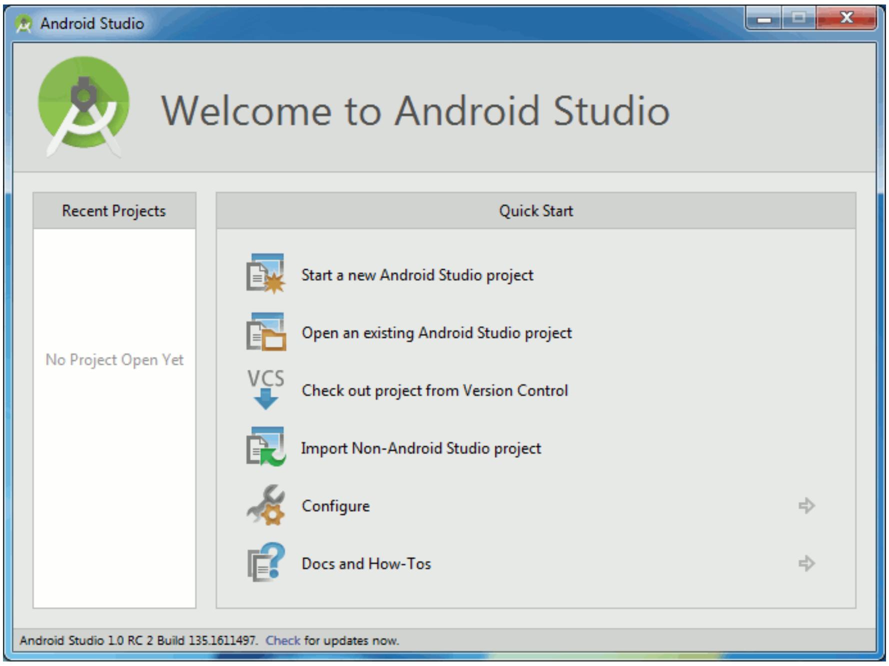
Facilitating a VM denotes that as long as it can read as well as transcribe the Bytecode’s catchwords, the program will satisfyingly run on its host platform, guaranteeing cross-platform compatibility.

Check your progress 3

1. Which language does android support to create an application?
   a. C++
   b. C
   c. Java
   d. .NET
2. Why Java Byte Code cannot be run on android?
   a. Because android use JVM
   b. Because Android use DVM
   c. Both of these
   d. None of these

1.5 Android Studio for Android Software Development

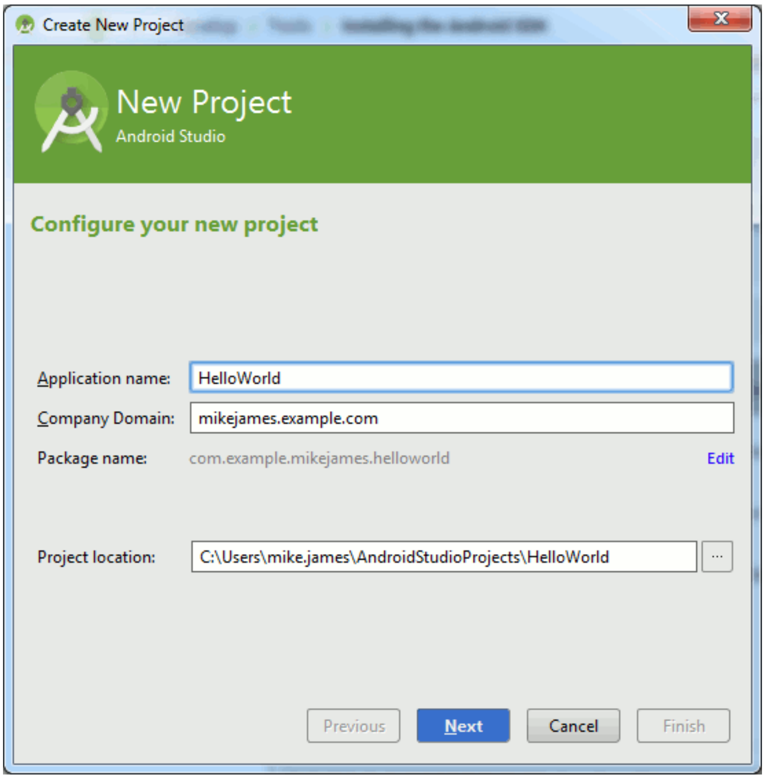
You can begin working on Android Studio once its installation is done on computer. Initially, you will find that the Android Studio will not start as it will first download its updates for Android SDK. On loading everything, you will find the Welcome screen as shown:


Android Studio
Welcome to Android Studio
Recent Projects
No Project Open Yet
Quick Start
Start a new Android Studio project
Open an existing Android Studio project
VCS
Check out project from Version Control
Import Non-Android Studio project
Configure
Docs and How-Tos

Android Studio 1.0 RC 2 Build 135.1611497. Check for updates now.

Fig 1.15 android studio

In this screen, you will see some programs already created in recent projects. If, this is your first project, in such case you have to select Start new Android Studio project option as shown:



Application name: HelloWorld

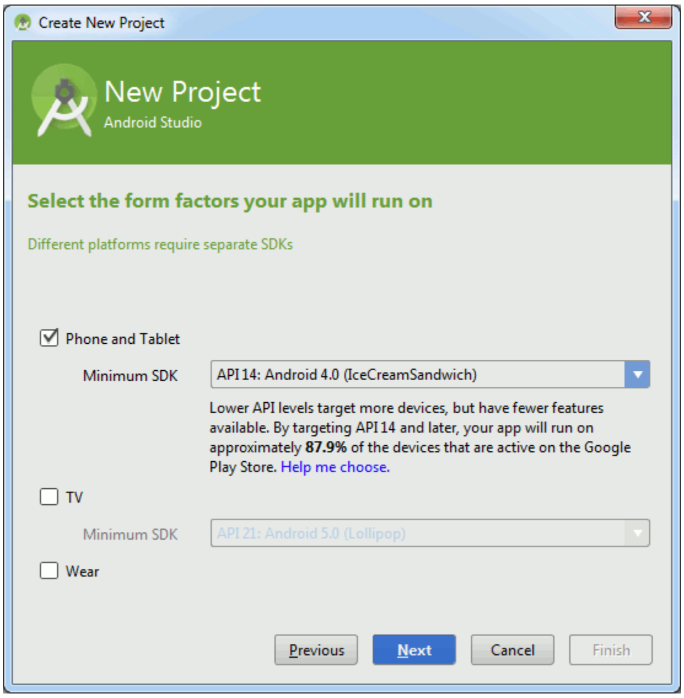
Company Domain: mikejames.example.com

Package name: com.example.mikejames.helloworld

Project location: C:\Users\mike.james\AndroidStudioProjects\HelloWorld

Fig 1.16 New projects

Here you need to adjust your details of new project by simply giving name for your application. Apart from this, you will find some standard details which Android Studio autofills itself. On clicking next you, you will find that a selection of device column will be there where you have to choose the device you are targeting:



Create New Project

New Project
Android Studio

Select the form factors your app will run on

Different platforms require separate SDKs

☑ Phone and Tablet

Minimum SDK: API 14: Android 4.0 (IceCreamSandwich)

Lower API levels target more devices, but have fewer features available. By targeting API 14 and later, your app will run on approximately **87.9%** of the devices that are active on the Google Play Store. Help me choose.

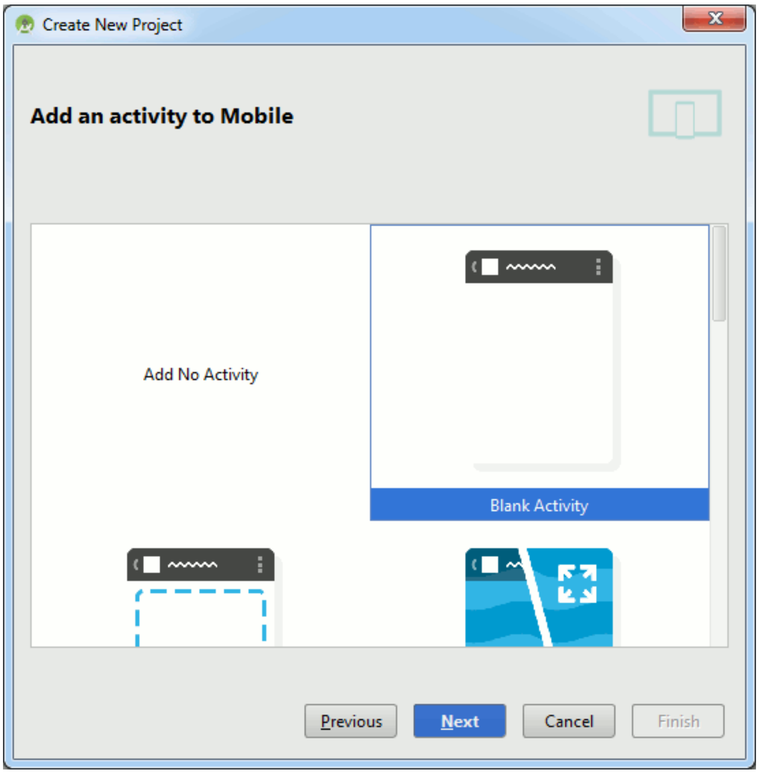
☐ TV

Minimum SDK: API 21: Android 5.0 (Lollipop)

☐ Wear

Fig 1.17 New project screen

Further, you have selected a template for a project. You will be given with default Blank Activity which all Android application has which will form a project having single activity.

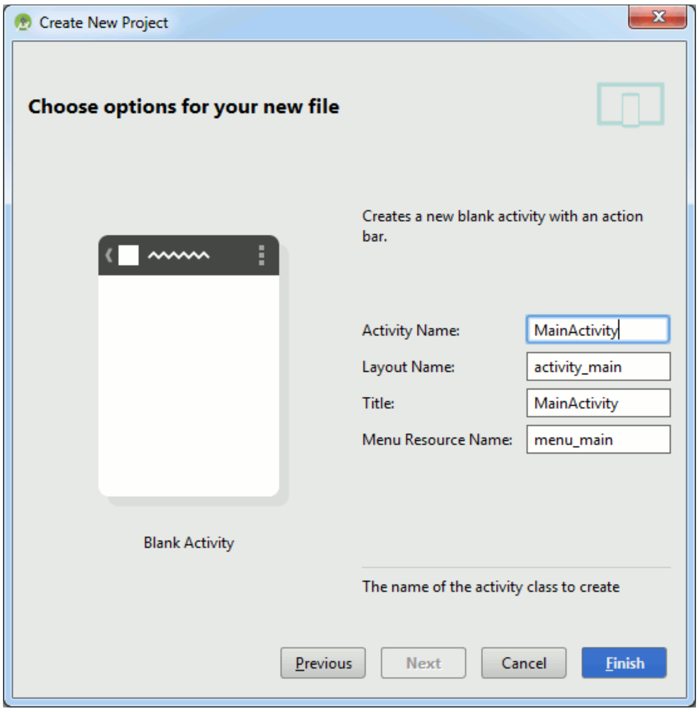


Create New Project

Add an activity to Mobile

Fig 1.18 Adding Activity to Mobile

In fig 1.19, you have to assign custom names for components of a project whose template appears. You can do that by giving name with default settings:



Create New Project

Choose options for your new file

Creates a new blank activity with an action bar.

Activity Name: MainActivity

Layout Name: activity_main

Title: MainActivity

Menu Resource Name: menu_main

The name of the activity class to create

Fig 1.19 Selecting Activity

After doing all this, you have to click on Finish button and wait for files that are created by Android Studio.

Check your progress 4

1. Name the tools that are present in Android SDK:

   a. Android Emulator

   b. DDMS(Dalvik Debug Monitoring Services),A

   c. APT(Android Asset Packaging tool)

   d. All of these

1.6 Building a sample Android application

Create a Project with Android Studio

Step 1: In Android Studio, create a new project. If you have no project shown in Welcome screen, then you have to click on New Project option. In case you have project opened, then in such case in File menu, you have to choose New Project which allows creating New Project from the screen which appears on desktop.

Step 2: Once the screen is there, you need to fill all fields and click on the Next.

- You have to give the Application Name in application name column.
- In Company domain qualifier, you have to write Company Name so that Android Studio will remember for each new project you create.
- In Package name, you have to fill name for the project which should be unique. You can Edit this independently from application name or company domain.
- In Project location, you have to specify the directory path of your system where your project files gets stored.

Step 3: Under Select the form factors your app will run on, check the box for Phone and Tablet.

Step 4: For Minimum SDK, you have to choose API 8: Android 2.2 (Froyo).

For Minimum Required SDK, you have to refer to earliest version of Android which is supported by app with API level.

Step 5: Leave options as TV, Wear, and Glass unchecked and click on Next.

Step 6: In Add an activity to <template>, select Blank Activity and click Next.

Step 7: In Customize the Activity, you can change Activity Name to MyActivity. In this, the Layout Name will change to activity_my and Title to MyActivity. The Menu Resource Name is menu_my.

Step 8: Now finally, choose the Finish button to create the project.

Check your progress 5

1. What is an Activity?

a. Activity performs actions on the screen

b. process

c. Both of these

d. None of these

1.7 Let Us Sum Up

In this unit we have learnt that Android continues as an open source as well as Linux-based operating system for mobile instruments comparable as smartphones along with tablet computers. It is seen that Android endures as an Operating System for mobile appliances constructed proximate Google, which continues developed upon Linux kernel.

It is stated that Android Studio IDE subsists impelled at Google I/O 2013. Android Studio is based on Intellij IDEA which contrivances to exchange Eclipse + Android Developer Tools for Android creation. Java continues the elementary language for enduring Android apps. You desire to grasp Java for Android development.

We see that you can start working on Android Studio once its installation is done on computer. For this, you will find that Android Studio will not start as it will first download its updates for Android SDK.

1.8 Answers for Check Your Progress

Check your progress 1

Answers: (1 –c), (2-a)

Check your progress 2

Answers: (1 -b), (2 -d)

Check your progress 3

Answers: (1 –c), (2 –b)

Check your progress 4

Answers: (1 -d)

Check your progress 5

Answers: (1 –a)

1.9 Glossary

1. **Activity -** An application screen that supports Java code from Activity class.
2. **Application -** In Android application, there are activities, services, listeners and intent receivers.

1.10 Assignment

Explain the architecture of Android?

1.11 Activities

Install Android SDK on your machine

1.12 Case Study

Try to create a project with activity in Android SDK

1.13 Further Reading

1. Learn Java for Android Development (2nd edition), Jeff Friesen, 2013

UNIT 2: ANDROID APPLICATION DESIGN ESSENTIALS - I

Unit Structure

2.0 Learning Objectives

After learning this unit, you will be able to understand:

- About Framework for Application
- About Application Context
- About Permissions
- About Receiving and Broadcasting Intents

2.1 Introduction

Android exercises a fresh design analogy catapulted by paper as well as ink that ascribes a commiserating feel of tactility. The awesome thing about Android is that it rushes on a bunch of facilities. The defective thing about Android is that it runs on an aggregate of appliances. Furthermore get devise designers: this alternate is individual going to get better/worse as more as well as more devices pop up on the market. The thing to remember when you design for Android is to NOT facilitate fixed-width arrangements. If you come from a web design experience, analyze in terms of "fluid layouts". Arrangements that will exaggerate based on the wideness of the browser, or in this case, the screen.

2.2 A Framework for a Well-Behaved Application

Android is the foremost platform for mobile conveniences which has aptitude to canter millions of mobile phones in about grown than 200 countries. It not sole ascribes a awesome backing for people around globe to facilitate hundreds of games as well as apps in their phones furthermore additionally assigns a big open marketplace to developers for Android App creation. Android attributes one partition along with distinctive configuration which assigns all Android developers to contribute their apps along with enduring them in arrangement to drive them feasible on numerous Smartphone, tablet users.

The main endeavour of arrangements is to advance productivity by easing works which incidentally preserves lot of time for developers to endure any other considerable consequences in the app or game. These arrangements provide inbuilt tools for developers to work immediately on complex as well as lengthy part of coding.

In today's fast execution world, almost comprehensive programming jargons have arrangements to help the developers. With present elevation of mobile devices a lot of compositions are additionally expanded whether it is for Android, Windows or iOS.

Android is a completely power lined operating system that ascribes energetic base to the world upholding lakes of approaches as well as games for android consumers as excellently as an open marketplace upholding Android App Development. It assigns you an individual as well as a abnormal application sample which authorizes you to contribute your apps expansively for Application development as well as App formation to hundreds of millions of consumers across a broad magnitude of devices that is from phones to tablets as well as offing. Android has undertaken 15 competent, open source as well as cross platform arrangements. These arrangements enrich Android App formulation along with Mobile App formulation.

- Basic4Android - www.basic4ppc.com
- Corona SDK - www.coronalabs.com
- DHTMLX Touch - www.dhtmlx.com
- Dojo Mobile - www.dojotoolkit.org
- iUI - www.code.google.com
- jQuery Mobile - www.jquerymobile.com
- kendo UI - www.kendoui.com
- Mono for Android - www.xamarin.com
- Mo Sync SDK - www.mosync.com
- PhoneGap - www.phonegap.com
- RhoMobile Suite - www.docs.rhomobile.com
- Sencha Touch 2 - www.sencha.com
- SproutCore - www.sproutcore.com
- TheAppBuilder - www.theappbuilder.com
- Titanium - www.appcelerator.com

Check your progress 1

1. Why Android is so popular?
    a. Open Source code
    b. Application framework
    c. Java support
    d. All of these

2.3 Application Context

An Android service continues a bunch of efforts, each of which is designated an action. Each action within a service has an exceptional action as well as user interface. To discover this more completely, consider a scholarly game practice designated Chippy's Revenge. The practice context is the core position for complete top-level operation benefit. You use the function context to approach settings as well as resources participated across multiple action instances. You can retrieve the application context for the current process by using the getApplicationContext() method, like this:

```
Context context = getApplicationContext();
```

Since the action class is derived from the Context class, you can facilitate this instead of acquiring the application context authentically. The operation context assigns approach to a number of top-level operation characteristics. Here are little larger things you can do with the application context:

- Launch Activity instances
- Retrieve assets packaged with the application
- Request a system-level service provider (for example, location service)
- Manage private application files, directories and databases
- Inspect and enforce application permissions

The first item on this list - launching Activity instances - is perhaps the most common reason you will use the application context.

Working with Activities

The action class is core to every Android function. Much of the time, you'll describe as well as exercise an activity for each cover in your operation. In the Chippy's Revenge game service, you possess to employ five contrary Activity classes. In the direction of practicing the game, the user adjustments from one activity to the following, interlacing with the layout commands of each activity.

Launching Activities

There are a number of ways to launch an activity, including the following:

- Designating a launch activity in the manifest file
- Launching an activity using the application context
- Launching a child activity from a parent activity for a result

- Designating a launch activity in the manifest file

In this, every Android application gives default activity inside Android manifest file where Droid1 project with DroidActivity describes default activity. Also, apart from this, other Activity classes should launch under particular situations. Here you have to handle secondary entry points simply by configuring Android manifest file with custom filters.

Launching activities using the application context

The easy and simple way to start an activity is to apply startActivity() method of application context which uses single parameter known as intent. Consider simple startActivity() call which will call following code with explicit intent:

startActivity(new Intent(getApplicationContext(), MenuActivity.class));

Such intent will request launch of target activity called MenuActivity by its class which can be implemented elsewhere inside the package. As MenuActivity class is explained inside an application's package, it has to be register as an activity inside Android manifest file. Such method is applicable in order to launch activity in theoretical game application.

Check your progress 2

1. Which of the following is the parent class of Activity?
   a. Object class
   b. Context
   c. Main Class
   d. Context Theme Wrapper
2. Start Activity() is the method of_______.
   a. Activity class
   b. Manifest file
   c. Application context
   d. None of these

2.4 Activities, Services

Activity

An Activity is a function constituent that delivers a screen with which consumers can reciprocate in array to do thing, alike as dial the phone, acquire a photo, furnish an email, or glance a map. Each exercise is assigned a window in which to illustrate its user interface. The window surely gluts the screen, but may be smaller than the screen and float on top of other windows.

An approach conventionally consists of multiple operations that are slackly connected to each other. Unconditionally, one activity in an exercise is individualized as the "core" activity, which is displayed to the consumer when impelling the application for the novel time. Each activity can that time commence another activity in normality to play contrary actions. Each duration a fresh activity starts, the old activity is halted, but the system conserves the activity in a stack. When a fresh activity commences, it is encouraged onto the back stack additionally takes consumer intersection. The back stack continues to the elementary "last in, first out" stack approach, so, when the consumer is finished with the immediate activity further presses the Back button, it is popped from the stack (and destroyed) and the old activity begins again.

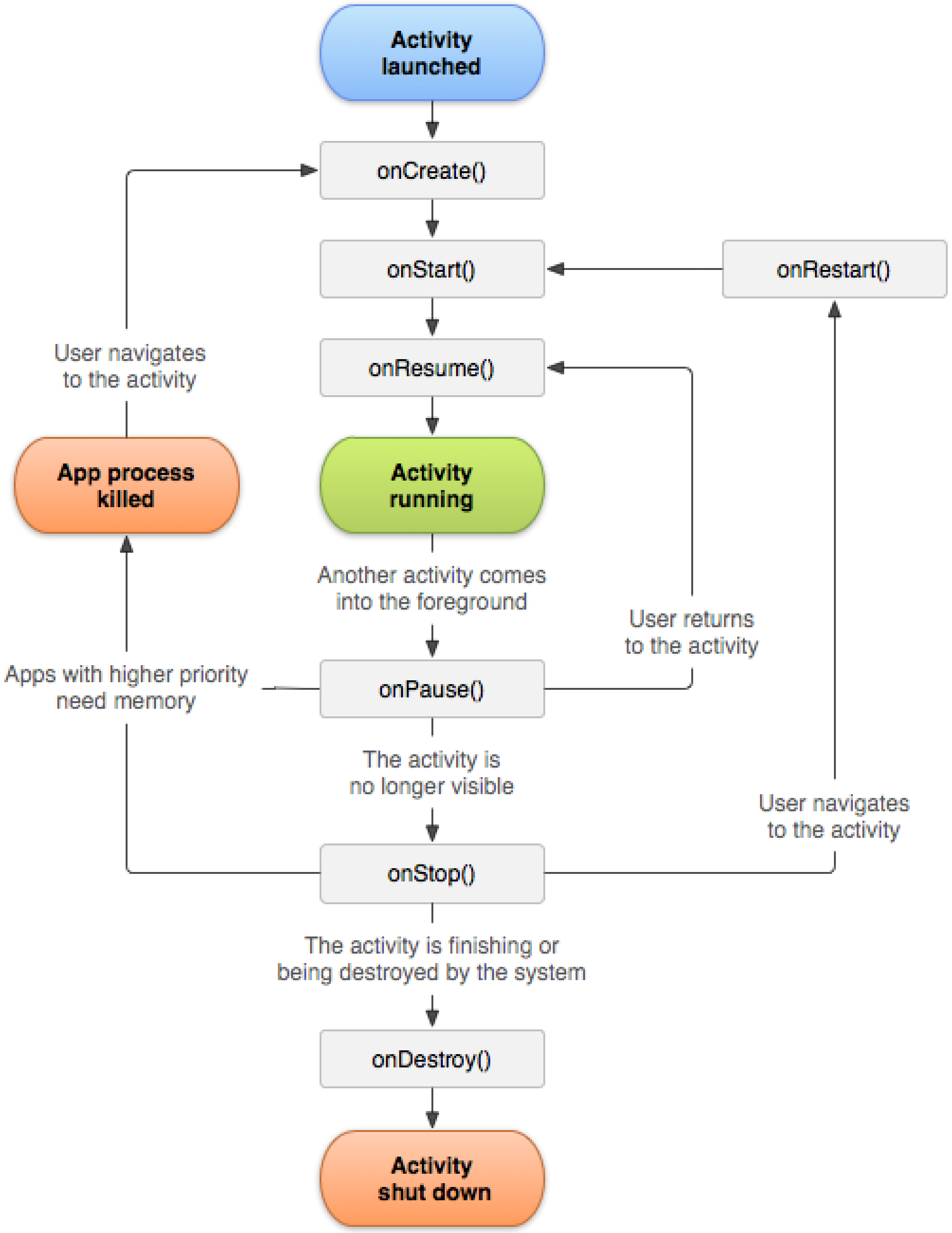

Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Fig 2.1 activity transitions between states

When an exercise is halted on account of fresh activity commencements, it is denoted of this adjustment in state through the activity's lifecycle callback approximations. There are numerous callback approximations that an activity might take, due to a adjustment in its state—whether the mechanism is formulating it, freezing it, reopening it, or demolishing it—and each callback supplies you the convenience to behave definite work that's admissible to that state change. For exemplary, when ceased, your activity should ease numerous large objects, comparable as network or database attachments. When the activity begins again, you can reacquire the essential reserve as well as recommence actions that were paused. These state adjustments are all portion of the activity lifecycle.

To start an activity:

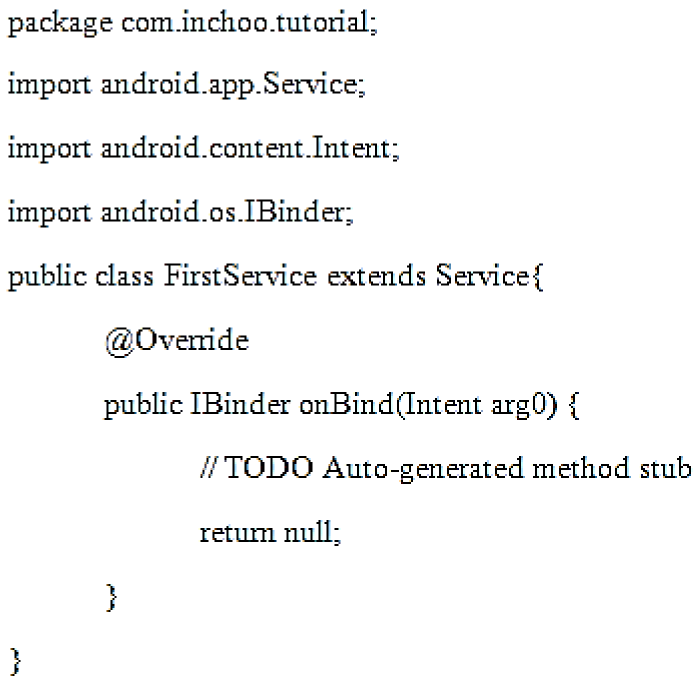
An Activity is a single screen in an app which begins with new instance by passing Intent to startActivity(). The Intent will show activity to start and carries required data. To have a result from activity on completion, you need to call startActivityForResult(), which will send result in activity's onActivityResult() callback.

Services

Service serves as operation element showing if an application's wish to do longer-running operation if not interacting with user or gives functionality for other applications for use.

Creating Simple Service:

You can create simple service request from FirstService in your namespace. It is known that Service class and Eclipse will help in importing required namespaces and helps in adding unimplemented methods (onBind):

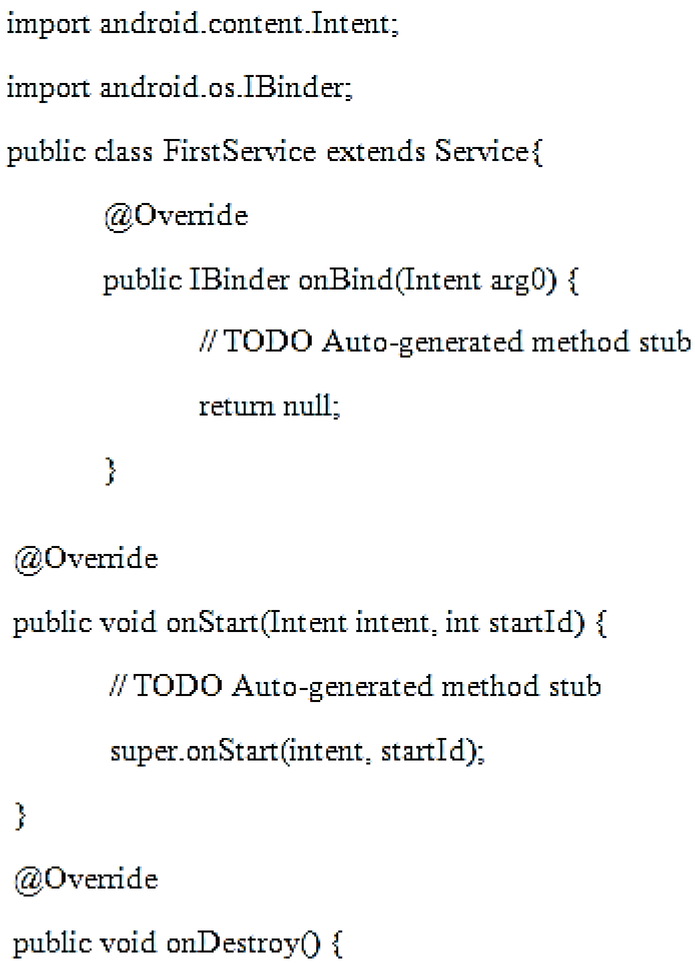


```
package com.inchoo.tutorial;

import android.app.Service;

import android.content.Intent;

import android.os.IBinder;

public class FirstService extends Service{

        @Override

        public IBinder onBind(Intent arg0) {

                // TODO Auto-generated method stub

                return null;

        }

}
```

Now we will override two more methods: onStart and onDestroy, so our FirstService class looks like this:

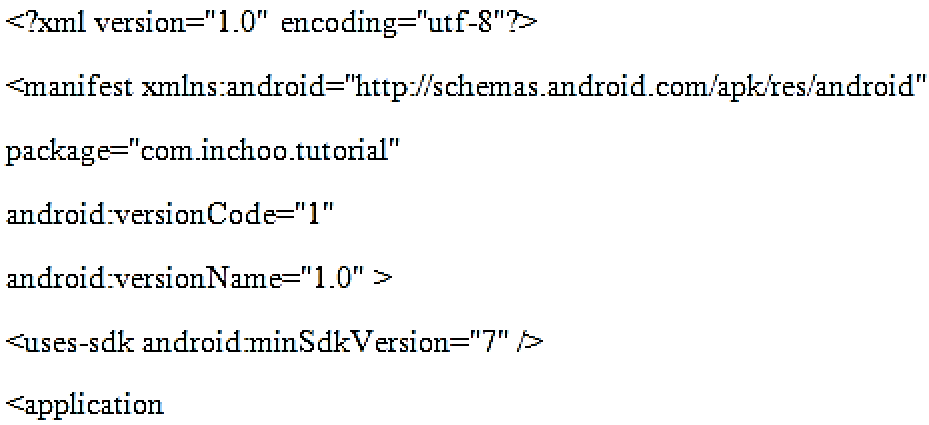
package com.inchoo.tutorial;

import android.app.Service;

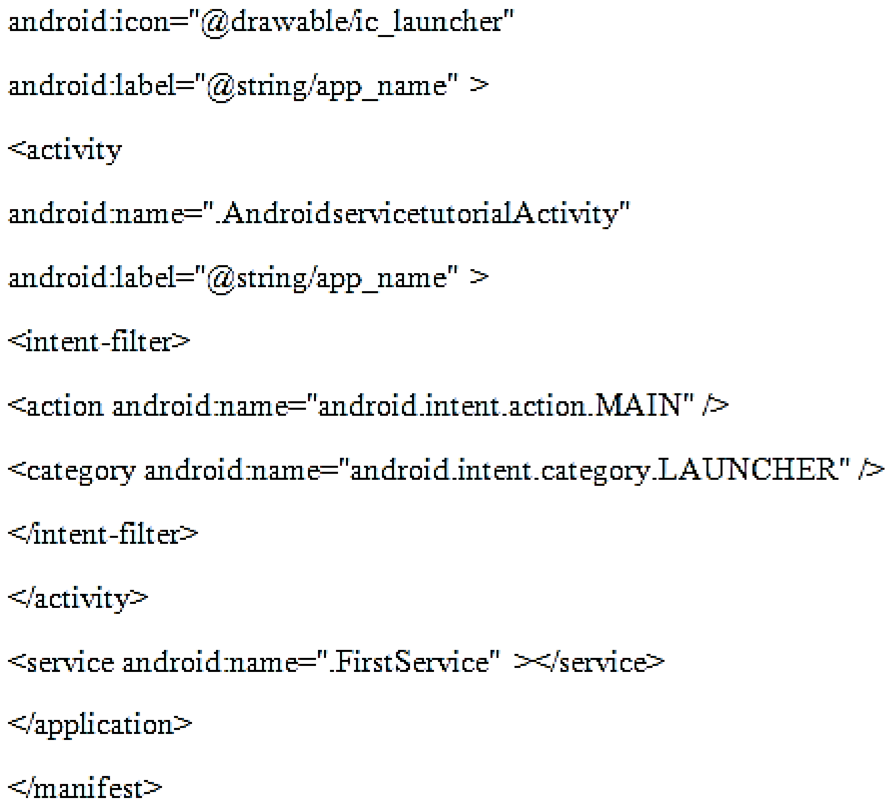


```
import android.content.Intent;

import android.os.IBinder;

public class FirstService extends Service{

        @Override

        public IBinder onBind(Intent arg0) {

                // TODO Auto-generated method stub

                return null;

        }

 @Override

 public void onStart(Intent intent, int startId) {

        // TODO Auto-generated method stub

        super.onStart(intent, startId);

 }

 @Override

 public void onDestroy() {

            // TODO Auto-generated method stub

            super.onDestroy();

      }

}
```

It is noted that you have to add service declaration in AndroidManifest.xml to have it properly as:

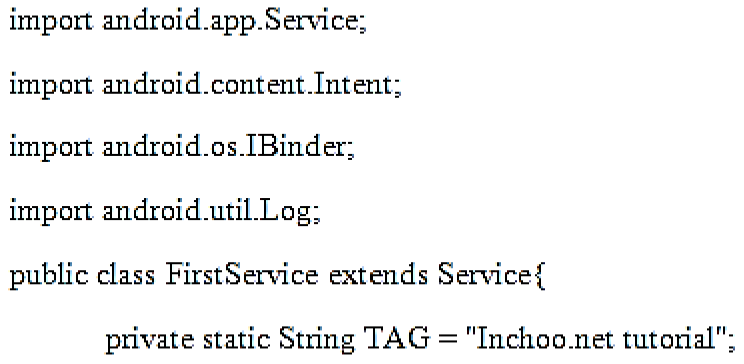


```
<?xml version="1.0"  encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="com.inchoo.tutorial"

android:versionCode="1"

android:versionName="1.0" >

<uses-sdk android:minSdkVersion="7" />

<application
```

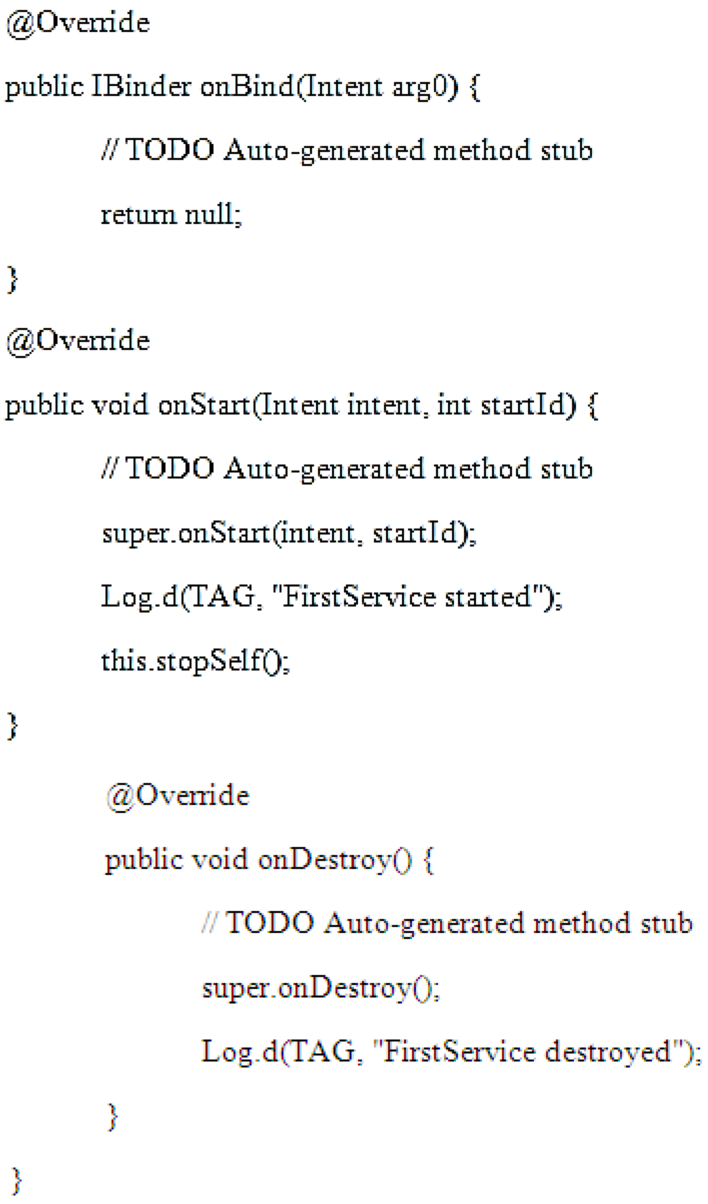


```
android:icon="@drawable/ic_launcher"
android:label="@string/app_name" >
<activity
android:name=".AndroidservicetutorialActivity"
android:label="@string/app_name" >
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
<service android:name=".FirstService"  ></service>
</application>
</manifest>
```

Now add log to see when service will start and vanish further can able to stop service immediately once started. We see that the FirstService.java class will look as shown:



```
import android.app.Service;
import android.content.Intent;
import android.os.IBinder;
import android.util.Log;
public class FirstService extends Service{
        private static String TAG = "Inchoo.net tutorial";
```



```
@Override
public IBinder onBind(Intent arg0) {
        // TODO Auto-generated method stub
        return null;
}
@Override
public void onStart(Intent intent, int startId) {
        // TODO Auto-generated method stub
        super.onStart(intent, startId);
        Log.d(TAG, "FirstService started");
        this.stopSelf();
}
        @Override
        public void onDestroy() {
                // TODO Auto-generated method stub
                super.onDestroy();
                Log.d(TAG, "FirstService destroyed");
        }
}
```

Check your progress 3

1. Which component is not activated by an Intent?

   a. Activity

   b. ContentProvider

   c. Content

   d. Services

2. What do you mean by services?

   a. It will perform background functionalities

   b. It will provide connection between activities and the data

   c. It will share the data between applications

   d. None of these

2.5 Intents and Intent Filter

Intent

Intent is a type of messaging object that are used to request an action from other app component. It facilitates communication among components in various manners. There are two types of intents present in the Android:

- Implicit Intents
- Explicit Intents

Explicit Intents

It is an intent which describes target component by its name which are mostly applied for application internal messages such as activity beginning with subordinate service having duplicate activity. It connects internal world of application, by connecting one activity with other.

Implicit Intents

It is an intent which will not name a target and field and in this the component name is left blank. It is applied to activate components in other applications.

To start a service:

Service serves as component which does operations in background without using user interface. It can be started for specific work by passing Intent to startService() function which shows the service to start with required data.

If service is framed as per client-server interface, in case, it will bind with other component using Intent to bindService().

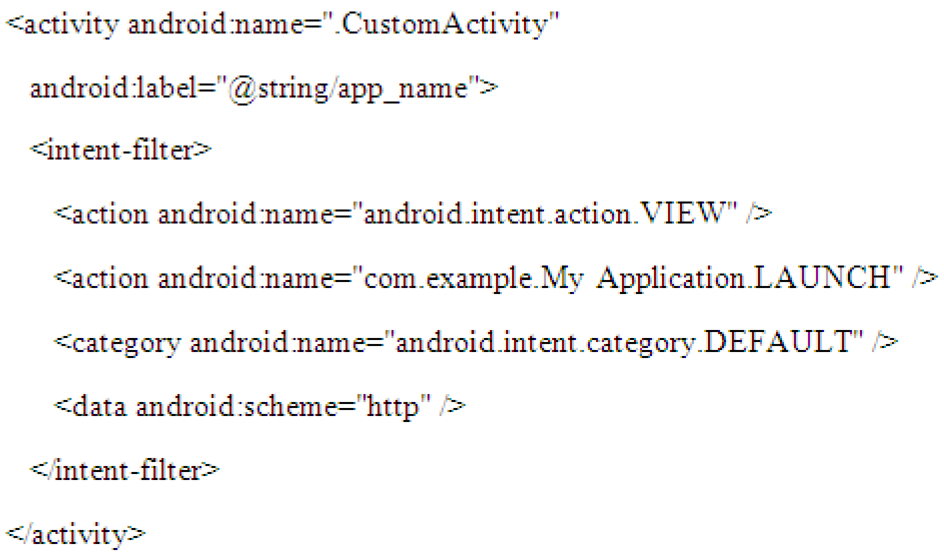
To deliver a broadcast:

Broadcast is a message that any app can receive. The system will show many broadcasts for system events which can be system booting or device charging. It can be delivered to other apps by passing Intent to sendBroadcast(), sendOrderedBroadcast(), or sendStickyBroadcast().

Intent Filters

It is seen earlier that Intent is applied to call other activity in an Android. The Android OS uses filters in order to pinpoint set of Activities, Services and Broadcast receivers will take care the Intent through particular set of action, categories, data scheme linked with Intent. You can apply <intent-filter> element in manifest file in order to list down actions, categories and data types which gets linked with activity, service or broadcast receiver.

The example below shows AndroidManifest.xml file which shows activity com.example.My Application.CustomActivity which can be call upon by certain actions such as:



```
<activity android:name=".CustomActivity"
  android:label="@string/app_name">
  <intent-filter>
    <action android:name="android.intent.action.VIEW" />
    <action android:name="com.example.My Application.LAUNCH" />
    <category android:name="android.intent.category.DEFAULT" />
    <data android:scheme="http" />
  </intent-filter>
</activity>
```

After the activity is defined with required filters and activities, it will call upon this activity with android.intent.action.VIEW or com.example.My Application.LAUNCH with android.intent.category.DEFAULT.

Check your progress 4

1. Which of the following is an example of Explicit intent?

   a. connecting one activity to another activity.

   b. connecting to web browser

   c. Both of these

   d. None of these

2. <intent-filter> element is used in_________.

a. content

b. startActivity()

c. sendBroadcast(),

d. Android Manifest File

2.6 Permissions

To have an access to certain features or applications, you need to have permission either to download or run an application in Android. To request for permission, follows simple, transparent and understandable rules. On request for access, application will make sure about feature which is clear for granting permission.

Usage

Permissions are bifurcated in many categories that allow consumers to have every permission which is carried in single action. It is seeing that permission is required by consumer to contacts with features to view and edit it. A user may perform an action that demonstrates clear intent, such as:

- Taking a picture
- Selecting a contact
- Starting a call or text message

In these cases, the user's action clearly demonstrates their intent, and no permission dialog is needed or authorized.

Runtime permissions

Application needs permission in order to access information or to use device capabilities any time once the installation is carried out. To work by user in an application, which could be operating of device camera, the application needs request permission for particular moment. Here, there are possibilities that the user may allow or deny permissions of any app from Android Settings anytime after installation.

Request patterns

The request for permissions relies continuously on clarity and importance of permission type which depends on ways of introducing permissions to user. In case of any serious permission, the request to be made at up-front, which alternate permission may be granted in-context. There needs to be more education on less clear permission which can be handled at either done up-front or in context. Giving feedback when permission is denied as this will save a feature from working as intended and that should be explained to the user.

Critical permissions

If the app can no longer run because a critical permission has been denied, explain why that permission must be allowed and offer a button to open Settings so the user can allow it.

Check your progress 5

1. What types of permissions are used in android?
   a. Critical Permission
   b. Secondary Permission
   c. Both of these
   d. None of these

2.7 Receiving and Broadcasting Intents

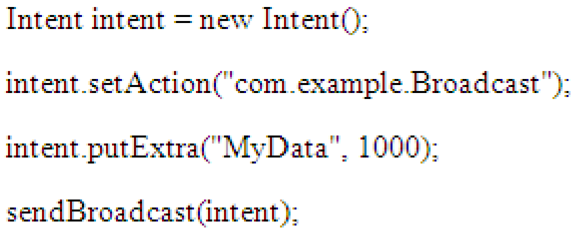
Intent gives facility for doing late runtime binding which exists aming coding in various applications. It is mainly used in creating activities where it sticks among certain activities. This is passive data structure holding that serves as an abstract showing action to be done which are:

- Action -- The general action to be performed, such as ACTION_VIEW, ACTION_EDIT, ACTION_MAIN, etc.
- Data -- The data to operate on, such as a person record in the contacts database, expressed as a Uri.

Broadcast Intents

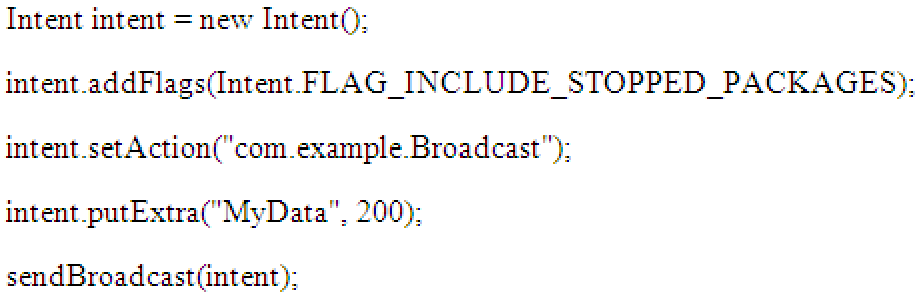
These are Intent objects which broadcast through call via sendBroadcast(), sendStickyBroadcast() or sendOrderedBroadcast() methods of Activity class. Apart from sending messages and event system among application components, these intents applied by Android system to show desired applications of key system events.

In order of creation of broadcast intent, it should carried action string with optional data and category string. With standard intents, data gets added to broadcast intent with key-value pairs in conjunction with putExtra() method of intent object. In this, the optional category string gets assigned to broadcast intent by calling to addCategory() method. Here the action string that shows broadcast event is different and normally uses application's Java package name syntax as shown:



```
Intent intent = new Intent();

intent.setAction("com.example.Broadcast");

intent.putExtra("MyData", 1000);

sendBroadcast(intent);
```

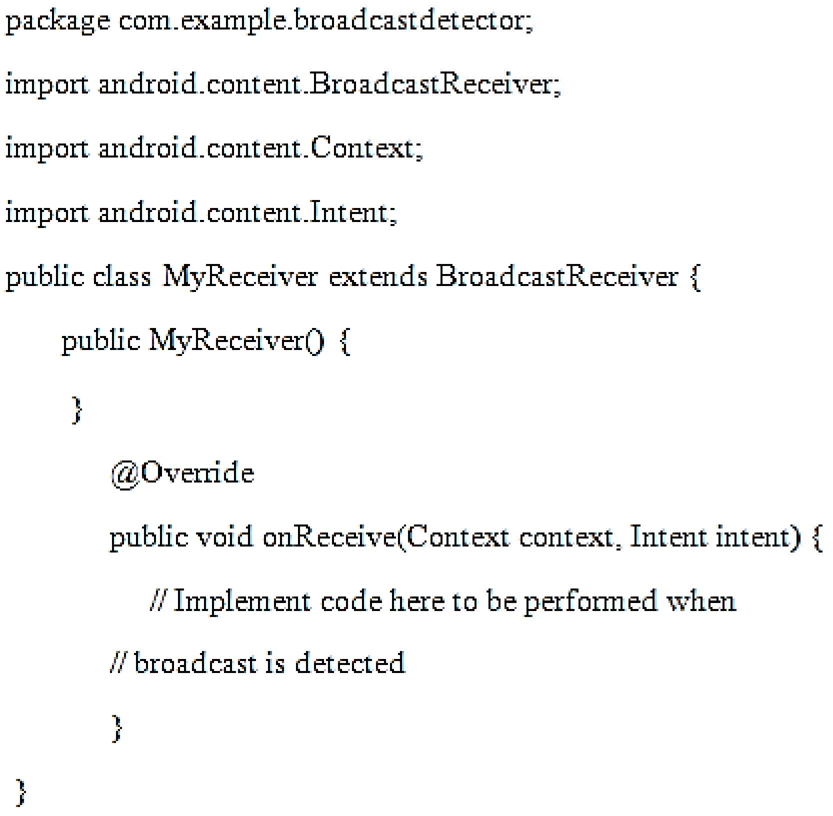
The above program will comfortably launch the required broadcast receiver on device which runs Android version earlier than 3.0. On more recent versions of Android, however, the intent would not be received by the broadcast receiver. This is because Android 3.0 introduced a launch control security measure that prevents components of stopped applications from being launched via an intent. To get around this, however, a flag can be added to the intent before it is sent to indicate that the intent is to be allowed to start a component of a stopped application. This flag is FLAG_INCLUDE_STOPPED_PACKAGES and would be used as outlined in the following adaptation of the previous code fragment:



```
Intent intent = new Intent();

intent.addFlags(Intent.FLAG_INCLUDE_STOPPED_PACKAGES);

intent.setAction("com.example.Broadcast");

intent.putExtra("MyData", 200);

sendBroadcast(intent);
```

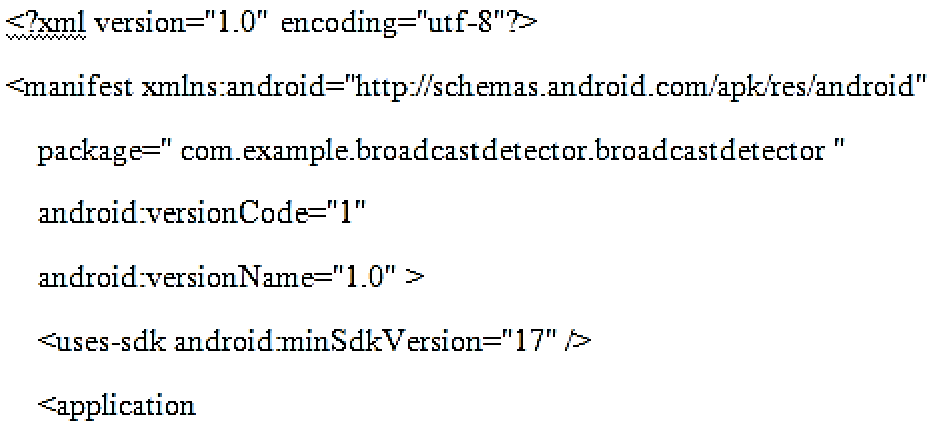
Broadcast Receivers

Broadcast receivers are implemented by extending the Android BroadcastReceiver class and overriding the onReceive() method. The broadcast receiver may then be registered, either within code or within a manifest file. When a matching broadcast is detected, the onReceive() method of the broadcast receiver is called, at which point the method has 5 seconds within which to perform any necessary tasks before returning. The following code outlines a template Broadcast Receiver subclass:

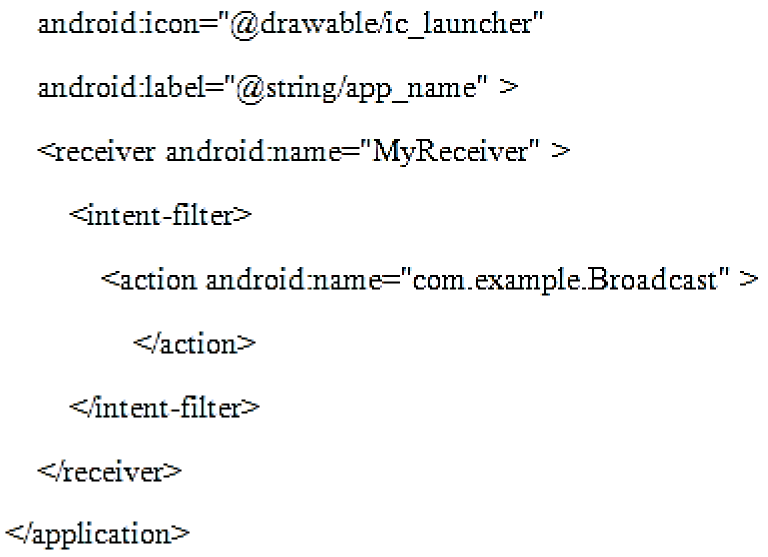


```
package com.example.broadcastdetector;

import android.content.BroadcastReceiver;

import android.content.Context;

import android.content.Intent;

public class MyReceiver extends BroadcastReceiver {

       public MyReceiver() {

        }

            @Override

            public void onReceive(Context context, Intent intent) {

                // Implement code here to be performed when

            // broadcast is detected

            }

 }
```

When registering a broadcast receiver inside manifest file, a <receiver> entry gets added that carries more intent filters that carry action string of broadcast intent for which receiver is required to listen. The program below will describe about broadcasting receiver which listens for broadcast intents having action string of com.example.Broadcast:



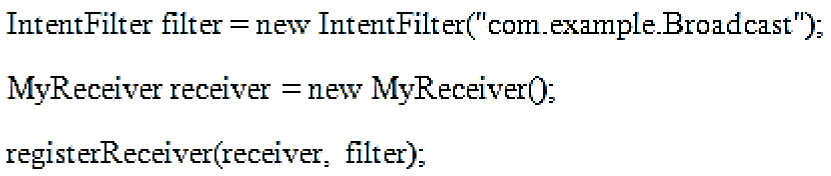
```
<?xml version="1.0"  encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

    package=" com.example.broadcastdetector.broadcastdetector "

    android:versionCode="1"

    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="17" />

    <application
```



```
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <receiver android:name="MyReceiver" >
            <intent-filter>
                <action android:name="com.example.Broadcast" >
                    </action>
            </intent-filter>
        </receiver>
    </application>
```

</manifest>

We see similar effect by registering broadcast receiver code with registerReceiver() method of Activity class along with required configured IntentFilter object:



```
IntentFilter filter = new IntentFilter("com.example.Broadcast");
MyReceiver receiver = new MyReceiver();
registerReceiver(receiver, filter);
```

On registering in code is not required, that done by unregisterReceiver() method of activity class using reference to receiver object as an argument as shown:

unregisterReceiver(receiver);

Check your progress 6

1. What is sticky intent in android?

    a. Method of intent

    b. Object of Activity class

    c. It is a type of intent which allows the communication between a function and a service

    d. None of these

2. Which of the following class is extended while implementing broadcast receivers?

    a. Broadcast Receiver class

    b. Activity class

    c. Context wrapper class

    d. None of these

2.8 Let Us Sum Up

While studying this unit, we have learnt that Android uses new design metaphor inspired by paper and ink that provides a reassuring sense of tactility. The great thing about Android is that it runs on a bunch of devices. Android is the greatest platform for mobile devices which has capability to run millions of mobile phones in about more than 200 countries. It not only provides a great support for people around world to use hundreds of games and apps in their phones but also provides a huge open marketplace to developers for Android App Development.

An Activity is an application component that provides a screen with which users can interact in order to do something, such as dial the phone, take a photo, send an email, or view a map. A Service is an application component representing either an application’s desire to perform a longer-running operation while not interacting with the user or to supply functionality for other applications to use. An Intent is a messaging object you can use to request an action from another app component. Although intent facilitate communication between components in several ways.

Permission requests should be simple, transparent, and understandable. When requesting access, apps should ensure that either the feature itself or an explanation provided makes it clear why permission is needed.

2.9 Answers for Check Your Progress

Check your progress 1

Answers: (1 –d)

Check your progress 2

Answers: (1 -d), (2-c)

Check your progress 3

Answers: (1 –b), (2-a)

Check your progress 4

Answers: (1 -b), (2-d)

Check your progress 5

Answers: (1 –c)

Check your progress 6

Answers: (1 –c), (2-a)

2.10 Glossary

1. **Activity -** An application screen that supports Java code from Activity class.
2. **Application -** In Android application, there are activities, services, listeners and intent receivers.
3. **Intent -** Message object which uses to communicate with certain applications/activities asynchronously.
4. **Service -** It is application component representing either an application's desire to perform a longer-running operation while not interacting with the user or to supply functionality for other applications to use.

2.11 Assignment

Write short note on Activities in android.

2.12 Activities

Try to create activity and services using Android SDK.

2.13 Case Study

Study Android Manifest File of your project.

2.14 Further Readings

1. Learn Java for Android Development (2nd edition), Jeff Friesen, 2013.
2. Android Application Development for Java Programmers, James C. Sheusi, 2012.

Block Summary

In this block, you have learnt and understand about the basic of Intents along with their working. The block gives an idea on the study and concept of various permission strategies in building an application in android. You have been well explained on the concepts of various android application formats.

The block detailed about the basic of Java in designing of Android applications with study about its targeted components. The concept related to installation regarding Java studio along with screenshots helps the students to understand better about development of application. You will be demonstrated practically with app component.

Block Assignment

Short Answer Questions

1. What do you understand by Activity in Android?
2. Explain Intents
3. Create a new Project in Android SDK and describe it briefly
4. What is Android Manifest File?

Long Answer Questions

1. Explain Services in Android in detail
2. Describe Android Intent filters
3. Create an activity in Android SDK to register user and shows his/her detail.

Enrolment No. ☐

1. How many hours did you need for studying the units?

Unit No	1	2	3	4
Nos of Hrs				

2. Please give your reactions to the following items based on your reading of the block:

Items	Excellent	Very Good	Good	Poor	Give specific example if any
Presentation Quality	□	□	□	□	____________ ____________
Language and Style	□	□	□	□	____________ ____________
Illustration used (Diagram, tables etc)	□	□	□	□	____________ ____________
Conceptual Clarity	□	□	□	□	____________ ____________
Check your progress Quest	□	□	□	□	____________ ____________
Feed back to CYP Question	□	□	□	□	____________ ____________

3. Any Other Comments

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

…………………………………………………………………………………………

> *Education is something which ought to be brought within the reach of every one.*

- Dr. B. R. Ambedkar

Dr. Babasaheb Ambedkar Open University
Jyotirmay' Parisar, Opp. Shri Balaji Temple, Sarkhej-Gandhinagar Highway, Chharodi,
Ahmedabad-382 481.

MOBILE APPLICATION DEVELOPMENT

PGDCA 203

BLOCK 2:
ANDROID APPLICATION AND USER INTERFACE DESIGN

Dr. Babasaheb Ambedkar Open University
Ahmedabad

MOBILE APPLICATION DEVELOPMENT

KNOWLEDGE MANAGEMENT & RESEARCH ORGANIZATION

Knowledge Management and
Research Organization
Pune

Editorial Panel

Author
Er. Nishit Mathur

Language Editor
Prof. Jaipal Gaikwad

Graphic and Creative Panel
Ms. K. Jamdal
Ms. Lata Dawange
Ms. Pinaz Driver
Ms. Ashwini Wankhede
Mr. Kiran Shinde
Mr. Prashant Tikone
Mr. Akshay Mirajkar

Copyright © 2015 Knowledge Management and Research Organization.
All rights reserved. No part of this book may be reproduced, transmitted or utilized in any form or by a means, electronic or mechanical, including photocopying, recording or by any information storage or retrieval system without written permission from us.

Acknowledgment
Every attempt has been made to trace the copyright holders of material reproduced in this book. Should an infringement have occurred, we apologize for the same and will be pleased to make necessary correction/amendment in future edition of this book.
The content is developed by taking reference of online and print publications that are mentioned in Bibliography. The content developed represents the breadth of research excellence in this multidisciplinary academic field. Some of the information, illustrations and examples are taken "as is" and as available in the references mentioned in Bibliography for academic purpose and better understanding by learner.'

ROLE OF SELF INSTRUCTIONAL MATERIAL IN DISTANCE LEARNING

The need to plan effective instruction is imperative for a successful distance teaching repertoire. This is due to the fact that the instructional designer, the tutor, the author (s) and the student are often separated by distance and may never meet in person. This is an increasingly common scenario in distance education instruction. As much as possible, teaching by distance should stimulate the student's intellectual involvement and contain all the necessary learning instructional activities that are capable of guiding the student through the course objectives. Therefore, the course / self-instructional material are completely equipped with everything that the syllabus prescribes.

To ensure effective instruction, a number of instructional design ideas are used and these help students to acquire knowledge, intellectual skills, motor skills and necessary attitudinal changes. In this respect, students' assessment and course evaluation are incorporated in the text.

The nature of instructional activities used in distance education self-instructional materials depends on the domain of learning that they reinforce in the text, that is, the cognitive, psychomotor and affective. These are further interpreted in the acquisition of knowledge, intellectual skills and motor skills. Students may be encouraged to gain, apply and communicate (orally or in writing) the knowledge acquired. Intellectual-skills objectives may be met by designing instructions that make use of students' prior knowledge and experiences in the discourse as the foundation on which newly acquired knowledge is built.

The provision of exercises in the form of assignments, projects and tutorial feedback is necessary. Instructional activities that teach motor skills need to be graphically demonstrated and the correct practices provided during tutorials. Instructional activities for inculcating change in attitude and behavior should create interest and demonstrate need and benefits gained by adopting the required change. Information on the adoption and procedures for practice of new attitudes may then be introduced.

Teaching and learning at a distance eliminates interactive communication cues, such as pauses, intonation and gestures, associated with the face-to-face method of teaching. This is particularly so with the exclusive use of print media. Instructional activities built into the instructional repertoire provide this missing interaction between the student and the teacher. Therefore, the use of instructional activities to affect better distance teaching is not optional, but mandatory.

Our team of successful writers and authors has tried to reduce this.

Divide and to bring this Self Instructional Material as the best teaching and communication tool. Instructional activities are varied in order to assess the different facets of the domains of learning.

Distance education teaching repertoire involves extensive use of self-instructional materials, be they print or otherwise. These materials are designed to achieve certain pre-determined learning outcomes, namely goals and objectives that are contained in an instructional plan. Since the teaching process is affected over a distance, there is need to ensure that students actively participate in their learning by performing specific tasks that help them to understand the relevant concepts. Therefore, a set of exercises is built into the teaching repertoire in order to link what students and tutors do in the framework of the course outline. These could be in the form of students' assignments, a research project or a science practical exercise. Examples of instructional activities in distance education are too numerous to list. Instructional activities, when used in this context, help to motivate students, guide and measure students' performance (continuous assessment)

PREFACE

We have put in lots of hard work to make this book as user-friendly as possible, but we have not sacrificed quality. Experts were involved in preparing the materials. However, concepts are explained in easy language for you. We have included may tables and examples for easy understanding.

We sincerely hope this book will help you in every way you expect.

All the best for your studies from our team!

Mobile Application Development

Contents

BLOCK 3: ANDROID NETWORKING AND DEVELOPMENT

UNIT 1 ADVANCED TOPICS - I

Android Networking, Web and Telephony API, Search, Location and Mapping, Sensors, NFC, Speech, Gestures, and Accessibility,

UNIT 2 ADVANCED TOPICS - II

Communication, Identity, Sync, and Social Media, The Android Native Development Kit (NDK)

BLOCK 4: ANDROID APPLICATION PUBLISHING AND CONTENT PROVIDERS

UNIT 1 MORE ON ANDROID

Handling and Persisting Data, A Content Provider as a Facade for a RESTful Web Service, Using Content Providers

UNIT 2 PUBLISHING ANDROID APPLICATION

Deploying Android Application to the World, Selling your Android application

Dr. Babasaheb
Ambedkar
Open University

PGDCA-203

MOBILE APPLICATION DEVELOPMENT

BLOCK 1: ANDROID APPLICATION AND USER INTERFACE DESIGN

BLOCK 2: ANDROID APPLICATION AND USER INTERFACE DESIGN

Block Introduction

Intents are asynchronous messages which permit application components to request functionality from other android components. Intents allow you to interact with components from the same applications additionally like components contributed by other applications. The new Fragment API for android, introduced in android 3.0, permits for easier dynamic user interfaces. API that is application program interface may be a set of routines, protocols, and tools for building software applications.

In this block, we will detail about the basic of Intent messages and usability of application components. The block will focus on the study and concept of running android 6.0 with API level 23 with permission requiring for access of additional features. You will get an idea on application in Android with necessity of Manifest.xml file.

In this block, you will make to learn and understand about new Fragment API for android for easy dynamic user interfaces. The concept related to interface elements in android app using view and ViewGroup objects will also be explained to the students. The student will be demonstrated practically about animations.

Block Objective

After learning this block, you will be able to understand:

- The basic of applying Intent Filter
- Study features of Permissions
- Concept of Android Manifest File
- Idea about managing different types application resources
- Features of Android User Interface Design
- Idea about User Interface Screen elements
- Concept of Layouts
- Working characteristics of Graphics and Multimedia

Block Structure

UNIT 1: ANDROID APPLICATION DESIGN ESSENTIALS - II

Unit Structure

1.0 Learning Objectives

After learning this unit, you will be able to understand:

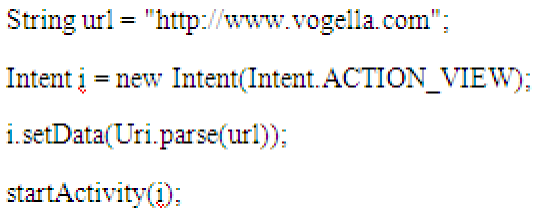
- About Intent Filter
- About Android Manifest File
- About application resources

1.1 Introduction

Your app's user interface is collectively that the user can see and collaborate with. Android provides a variety of pre-built UI components such as structured layout objects and UI controls that allow you to build the graphical user interface for your app. Android also provides other UI modules for different interfaces such as dialogs, notifications, and menus.

1.2 Using Intent Filter

Intents are asynchronous messages which permit application components to request functionality from other android components. Intents allow you to interact with components from the same applications additionally like components contributed by other applications. Intents are used to signal to the android system that a precise event has occurred. Intents typically describe the action that should be performed and provide data upon that such an action should be done. for example, your application will begin a browser component for a certain url via an intent. This is demonstrated by the following example.



```
String url = "http://www.vogella.com";

Intent i = new Intent(Intent.ACTION_VIEW);

i.setData(Uri.parse(url));

startActivity(i);
```

A component will register itself via an intent filter for a selected action and specific data. an intent filter specifies the kinds of intents to that an activity, service, or broadcast receiver will respond to by declaring the capabilities of a component.

Android components register intent filters either statically within the androidManifest.xml or just in case of a broadcast receiver also dynamically via code. An intent filter is defined by its category, action and data filters. It also can contain additional meta-data.

If intent is sent to the android system, the android platform runs a receiver determination. It uses the data included within the intent. If many components have registered for the same intent filter, the user will decide that component should be started.

Intent filter

You can register your android components via intent filters for certain events. If a component doesn't define one, it will solely be referred to as by explicit intents. This chapter offers an example for registering a component for intent. The key for this registration is that your component registers for the correct action, mime-type and specifies the correct meta-data.

To inform the system those implicit intents they will handle, activities, services, and broadcast receivers will have one or more intent filters. each filter describes a capability of the component, a set of intents that the component is willing to receive. It, in effect, filters in intents of a desired kind, whereas filtering out unwanted intents — however solely unwanted implicit intents (those that do not name a target class). a certain intent is usually delivered to its target, no matter what it contains; the filter isn't consulted. However an implicit intent is delivered to a component as long as it will undergo one of the component's filters.

An intent filter is an instance of the IntentFilter class. However, since the android system should realize the capabilities of a part before it will launch that part, intent filters are typically not set up in Java code, however within the application's manifest file (AndroidManifest.xml) as &lt;intent-filter&gt; elements. (The one exception would be filters for broadcast receivers that square measure registered dynamically by calling Context.registerReceiver(); they're directly created as IntentFilter objects.)

A filter has fields that parallel the action, data, and category fields of an Intent object. An implicit intent is tested against the filter in all 3 areas. To be delivered to the component that owns the filter, it must pass all 3 tests. If it fails even one in all them, the android system will not deliver it to the part — at least not on the basis of that filter. However, since a component can have multiple intent filters, an intent that doesn't pass through one of a component's filters may make it through on another.

Check your progress 1

1. What is android intent?

   a. It is an abstract description of an operation to be performed

   b. activity

   c. process

   d. None of these

2. The Primary elements of intent are____________________.

   a. Action

   b. Data

   c. Both of these

   d. None of these

1.3 Permissions

It's easy for an app to overwhelm a user with permission requests. If a user finds the app frustrating to use, or the user is worried regarding what the app could be doing with the user's information, they will avoid using the app or uninstall it entirely. Whenever you ask for permission, you force the user to form a decision. You must minimize the amount of times you make these requests. If the user is running android 6.0 (API level 23) or later, whenever the user tries some new app feature that needs a permission, the app needs to interrupt the user's work with a permission request. If the user is running an earlier version of golem, the user needs to grant all of the app's permissions when installing the app; if the list is just too long or appears inappropriate, the user could decide not to install your app at all. For these reasons, you must minimize the amount of permissions your app wants.

If the user is running android 6.0 (API level 23) or later, the user needs to grant your app its permissions whereas they're running the app. If you confront the user with a lot of requests for permissions promptly, you'll overwhelm the user and cause them to quit your app. Instead, you should invite permissions as you need them.

In some cases, one or additional permissions could be absolutely essential to your app. it would add up to ask for all of these permissions as soon because the app launches. For example, if you make a photography app, the app would wish access to the device camera. Once the user launches the app for the first time, they will not be surprised to be asked for permission to use the camera. However if the same app also had a feature to share photos with the user's contacts, you probably mustn't invite the READ_CONTACTS permission at first launch. Instead, wait till the user tries to use the "sharing" feature and ask for the permission.

The permissions dialog shown by the system when you call requestPermissions() says that what permission your app wants, but doesn't tell why. In some cases, the user may find it confusing. It's has been a good idea to explain to the user why your app wants the permissions before calling requestPermissions().

For example, a photography app might want to use location services so it can geotag the photos. A typical user might not understand that a photo can contain location information, and would be puzzled why their photography app wants to know the location. So in this case, it's a good idea for the app to tell the user about this feature before calling requestPermissions().

Check your progress 2

1. Which method is called when the app seeks permission from the user?

   a. startActivitity()

   b. requestPermission()

   c. broadcastReceiver()

   d. None of these

1.4 Android Manifest File and its common settings

Every application should have an AndroidManifest.xml file (with precisely that name) in its root directory. The manifest file presents essential info regarding your app to the android system, info the system must have before it will run any of the app's code. Among other things, the manifests will the following:

- It names the Java package for the application. The package name is a unique identifier for the application.
- It describes the components of the application — the activities, services, broadcast receivers, and content providers that the application consists of. It names the classes that implement every of the elements and publishes their capabilities (for example, that Intent messages they will handle). These declarations let the android system know what the components are and under what conditions they will be launched.
- It determines that processes can host application components.
- It declares that permissions the application should have so as to access protected parts of the API and interact with other applications.
- It additionally declares the permissions that others are needed to have so as to move with the application's components.
- It lists the Instrumentation classes that provide profiling and other info because the application is running. These declarations are present within the manifest solely whereas the application is being developed and tested; they are removed before the application is published.
- It declares the minimum level of the android API that the application needs.
- It lists the libraries that the application should be linked against.

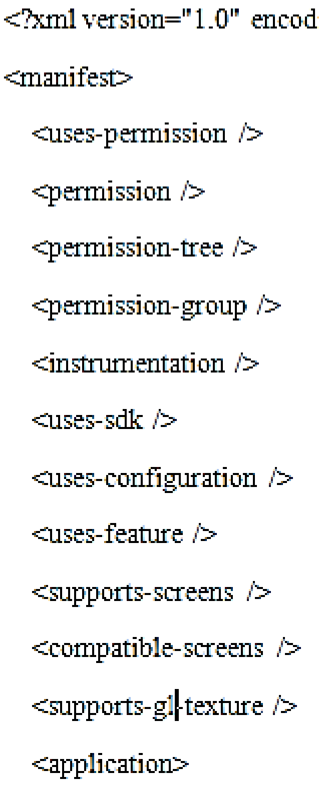
AndroidManifest.xml could be a powerful go in the android platform that enables you to describe the functionality and needs of your application to android.

However, working with it's not easy. Xamarin.Android helps to reduce this problem by permitting you to feature custom attributes to your classes, which can then be wont to automatically generate the manifest for you. Our goal is that 99 of our users should never need to manually modify AndroidManifest.xml.

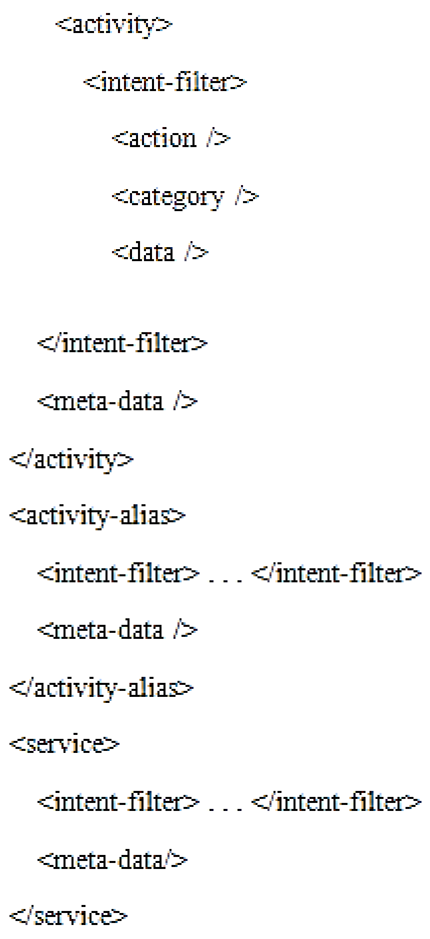
AndroidManifest.xml is generated as a part of the build method, and also the XML found inside Properties/AndroidManifest.xml is merged with XML that's generated from custom attributes. The resulting merged AndroidManifest.xml resides within the obj subdirectory; for example, it resides at obj/Debug/android/AndroidManifest.xml for debug builds. The merging process is trivial: it uses custom attributes within the code to generate XML elements, and inserts those elements into AndroidManifest.xml.

Structure of Manifest File

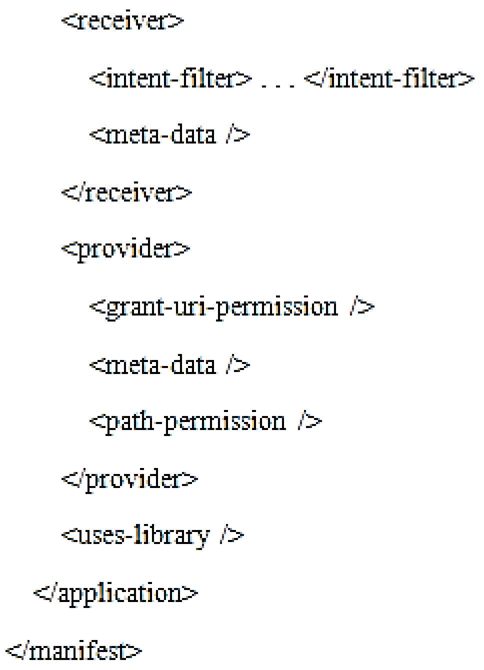
The code below shows the structure of manifest file and every element that it can contain. Each element, along with all of its attributes, is documented in full in a separate file.



```
<?xml version="1.0" encoding="utf-8"?>
<manifest>
   <uses-permission />
   <permission />
   <permission-tree />
   <permission-group />
   <instrumentation />
   <uses-sdk />
   <uses-configuration />
   <uses-feature />
   <supports-screens />
   <compatible-screens />
   <supports-gl-texture />
   <application>
```

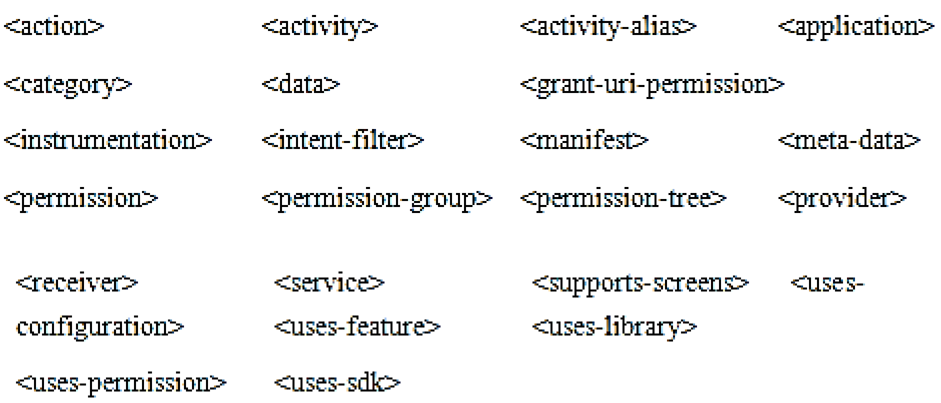


```
    <activity>
      <intent-filter>
        <action />
        <category />
        <data />

   </intent-filter>
   <meta-data />
 </activity>
 <activity-alias>
   <intent-filter> . . . </intent-filter>
   <meta-data />
 </activity-alias>
 <service>
   <intent-filter> . . . </intent-filter>
   <meta-data/>
 </service>
```



```
    <receiver>
      <intent-filter> . . . </intent-filter>
      <meta-data />
    </receiver>
    <provider>
      <grant-uri-permission />
      <meta-data />
      <path-permission />
    </provider>
    <uses-library />
  </application>
</manifest>
```

All the elements that appear in manifest file are highlighted below. These are the standard elements that are used in manifest files.



<action> <activity> <activity-alias> <application>

<category> <data> <grant-uri-permission>

<instrumentation> <intent-filter> <manifest> <meta-data>

<permission> <permission-group> <permission-tree> <provider>

<receiver> <service> <supports-screens> <uses-configuration> <uses-feature> <uses-library>

<uses-permission> <uses-sdk>

Check your progress 3

1. What is android manifest file?

   a. java file

   b. drivers

   c. It is a file that application code, resources, permissions, icons and themes

   d. None of these

2. What is the use of manifest file?

   a. It names the Java package for the application

   b. It determines which processes will host application components

   c. It lists the libraries of the application

   d. All of these

1.5 Managing different types application resources in a hierarchy

The well-written application accesses its resources programmatically rather than hard coding them into the source code. this is often done for a variety of reasons. Storing application resources in a very single place is a additional organized approach to development and makes the code more legible and maintainable. Externalizing resources like strings makes it easier to localize applications for various languages and geographic regions. All android applications are composed of 2 things:

- Functionality or code instructions

- Data or resources

The functionality is that the code that determines however your application behaves. This includes any algorithms that make the application run. Resources include text strings, images and icons, audio files, videos, and other data employed by the application.

Android resource files are stored separately from the java class files within the android project. Most common resource types are stored in XML. You'll additionally store raw data files and graphics as resources. Resources are organized in a very strict directory hierarchy within the android project. All resources should be stored under the /res project directory in specially named subdirectories that has to be lowercase.

Application resources are created and stored within the android project files under the /res directory. Employing a well-defined however flexible directory structure, resources are organized, defined, and compiled with the appliance package. Application resources are not shared with the rest of the android system.

Storing Application Resources

Defining application data as resources (as opposed to at runtime in code) is good programming practice. Grouping application resources together and compiling them into the application package has the following benefits:

- Code is cleaner and easier to read, resulting in fewer bugs.
- Resources are organized by type and guaranteed to be unique.
- Resources are conveniently set for handset customization.
- Localization and internationalization are simple.

The android platform supports a range of resource sorts, which might be combined to create differing kinds of applications. Android applications will include many different varieties of resources. The following are a number of the most common resource types:

- Strings, colors, and dimensions
- Drawable graphics files
- Layout files
- Raw files of all types

Resource sorts are defined with special XML tags and organized into specially named project directories. Some /res subdirectories, like the /drawable,

/layout, and /values directories, are created by default once a new android project is formed, however others should be added by the developer once required.

Resource files stored within /res subdirectories must abide by the following rules:

- Resource filenames must be in lowercase.
- Resource filenames may contain letters, numbers, underscores, and periods only.
- Resource filenames must be unique.

When resources are compiled, their name dictates their variable name. For example,a graphics file saved within the /drawable directory as mypic.jpg is referenced as @drawable/mypic. It's necessary to name resource names intelligently and be aware of character limitations that are stricter than file system names. Consult the android documentation for specific project directory naming conventions.

Simple resources like string, color, and dimension values ought to be defined in XML files under the /res/values project directory in XML files. These resource files use special XML tags that represent name/value pairs. These types of resources are compiled into the application package at build time.

All application resources are keep within the /res project directory structure and are compiled into the project at build time. Application resources may be used programmatically. They'll even be referenced in different application resources. Application resources may be accessed programmatically using the generated class file called R.java. To reference a resource from within your Activity class, you must retrieve the application's Resources object using the getResources() method then build the appropriate method call, supported the kind of resource you want to retrieve.

Check your progress 4

1. In android resources are stored inside_________________.
   a. Java file
   b. Layout folder
   c. Res Folder of XML file
   d. None of these

1.6 Let Us Sum Up

In this unit we have learnt that Intents are asynchronous messages which permit application components to request functionality from other android components that allow interacting with components from similar applications with components that are contributed by other applications.

If user is running android 6.0 or later, whenever the user tries some new app feature that needs permission, the app needs to interrupt the user's work with a permission request.

It is noted that all application should have AndroidManifest.xml file in its root directory that shows required info regarding an app for android system.

It is found that a well-written application accesses its resources programmatically rather than hard coding them into the source code which often is carried out for variety of reasons.

1.7 Answers for Check Your Progress

Check your progress 1

Answers: (1 –a), (2-c)

Check your progress 2

Answers: (1 -b)

Check your progress 3

Answers: (1 –c), (2 –d)

Check your progress 4

Answers: (1 -c)

1.8 Glossary

1. **Activity -** An application screen that supports Java code from Activity class.

2. **Application -** In Android application, there are activities, services, listeners and intent receivers.
3. **Intent -** Message object which uses to communicate with certain applications/activities asynchronously.

1.9 Assignment

Explain intent and intent filters with the help of an example.

1.10 Activities

Study android manifest file of your project.

1.11 Case Study

Study android application resources.

1.12 Further Readings

1. Learn Java for Android Development (2nd edition), Jeff Friesen, 2013
2. Android Application Development for Java Programmers, James C. Sheusi, 2012

UNIT 2: ANDROID USER INTERFACE DESIGN AND COMMON APIS

Unit Structure

2.0 Learning Objectives

After learning this unit, you will be able to understand:

- About User Interface Screen
- About Working with Animation
- About Drawing 2D and 3D Graphics

2.1 Introduction

The new Fragment API for android, introduced in android 3.0, permits for easier dynamic user interfaces. API that is application program interface may be a set of routines, protocols, and tools for building software applications. The API specifies however software components should interact and apis are used once programming graphical program (GUI) components. A good API makes it easier to develop a program by providing all the building blocks. A programmer then puts the blocks together.

2.2 User Interface Screen elements

All user interface elements in an android app are built using view and ViewGroup objects. A view is an object that draws one thing on the screen that the user will move with. A viewGroup is an object that holds alternative View (and ViewGroup) objects so as to define the layout of the interface.

The user interface for every component of your app is defined using a hierarchy of view and ViewGroup objects, as shown in figure 2.1. every view group is an invisible container that organizes child views, whereas the child views is also input controls or other widgets that draw some part of the UI. This hierarchy tree is often as easy or complicated as you wish it to be.

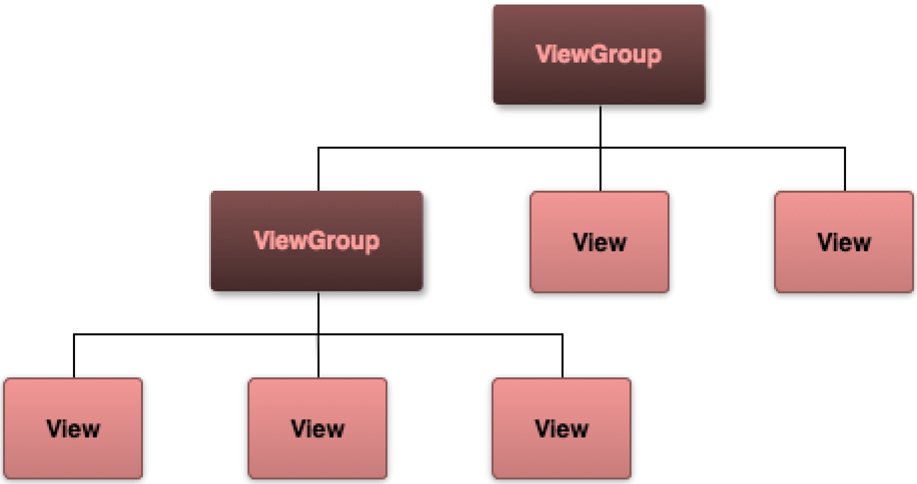


Fig 2.1 User Interface components

To declare your layout, you can instantiate View objects in code and start building a tree, but the easiest and the most effective way to define your layout is by using an XML file. XML offers a human-readable structure for the layout, similar to HTML. The name of an XML element for a view is respective to the Android class it represents. So a <TextView> element creates a TextView widget in your UI, and a <LinearLayout> element creates a LinearLayout view group.

Check your progress 1

1. What is view group?

   a. class

   b. It is a collection of views

   c. object

   d. None of these

2. What are the attributes of <TextView>?

   a. android:id

   b. android:editable

   c. android:fontFamily

   d. All of these

2.3 Designing User Interfaces with Layouts

A layout defines the visual structure for a program, like the UI for an activity or app convenience. You’ll be able to declare a layout in 2 ways:

- Declare UI elements in XML. android provides a simple XML vocabulary that corresponds to the read classes and subclasses, like those for widgets and layouts.
- Instantiate layout elements at runtime. Your application will create read and ViewGroup objects (and manipulate their properties) programmatically.

The android framework provides you the flexibility to use either or each of those methods for declaring and managing your application's UI. For example, you'll declare your application's default layouts in XML, including the screen elements which will seem in them and their properties. You’ll then add code in your application that would modify the state of the screen objects, including those declared in XML, at run time.

The advantage to declaring your UI in XML is that it allows you to better separate the presentation of your application from the code that controls its behavior. Your UI descriptions are external to your application code, which implies that you simply will modify or adapt it while not having to modify your source code and recompile. For example, you'll produce XML layouts for different screen orientations, totally different device screen sizes, and totally different languages. Additionally, declaring the layout in XML makes it easier to visualize the structure of your UI, therefore it's easier to debug problems. As such, this document focuses on teaching you how to declare your layout in XML. If you are interested in instantiating view objects at runtime, refer to the ViewGroup and view class references.

XML layout attributes named layout_something define layout parameters for the view that are acceptable for the ViewGroup in which it resides.

Every ViewGroup class implements a nested class that extends ViewGroup.LayoutParams. This subclass contains property varieties that define the size and position for every child view, as appropriate for the view group. As you'll see in figure 2.2, the parent view group defines layout parameters for every child view.

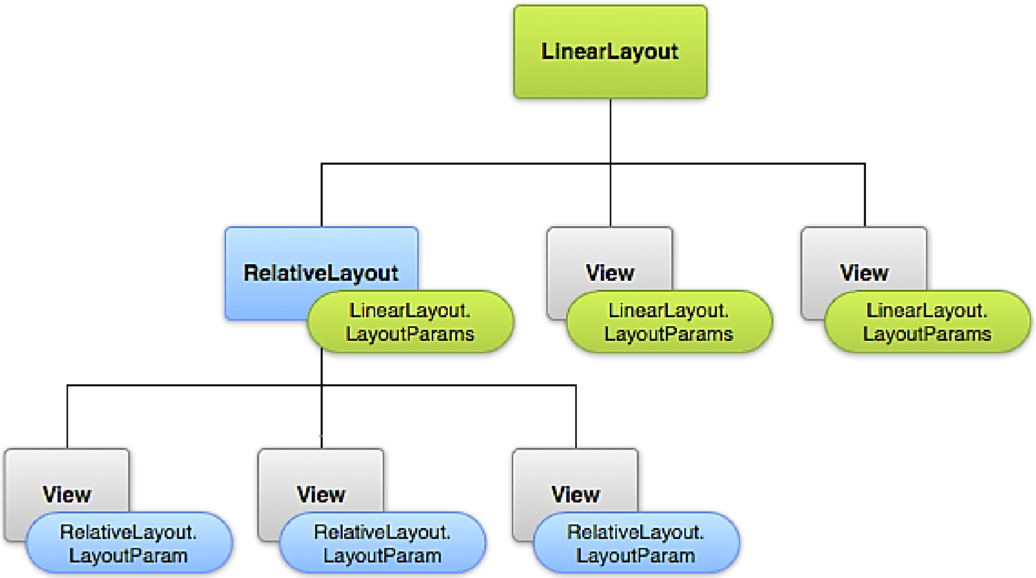

LinearLayout
RelativeLayout
LinearLayout. LayoutParams
View
LinearLayout. LayoutParams
View
LinearLayout. LayoutParams
View
RelativeLayout. LayoutParam
View
RelativeLayout. LayoutParam
View
RelativeLayout. LayoutParam

Fig 2.2 Designing Layout

The geometry of a view is that of a rectangle. A view includes a location, expressed as a pair of left and top coordinates, and 2 dimensions, expressed as a width and a height. The unit for location and dimensions is the pixel.

It is possible to retrieve the location of a view by invoking the methods getLeft() and getTop(). The previous returns the left, or X, coordinate of the rectangle representing the view. The latter returns the top, or Y, coordinate of the rectangle representing the view. These methods both return the location of the view relative to its parent. For instance, when getLeft() returns 20, which means the view is located 20 pixels to the right of the left edge of its direct parent.

In addition, many convenience strategies are offered to avoid unnecessary computations, specifically getRight() and getBottom(). These strategies come the coordinates of the correct and bottom edges of the rectangle representing the view. for instance, calling getRight() is similar to the following computation: getLeft() + getWidth().

Each subclass of the ViewGroup class provides a unique way to show the views you nest within it. Below are a number of the a lot of common layout types that are built into the android platform.

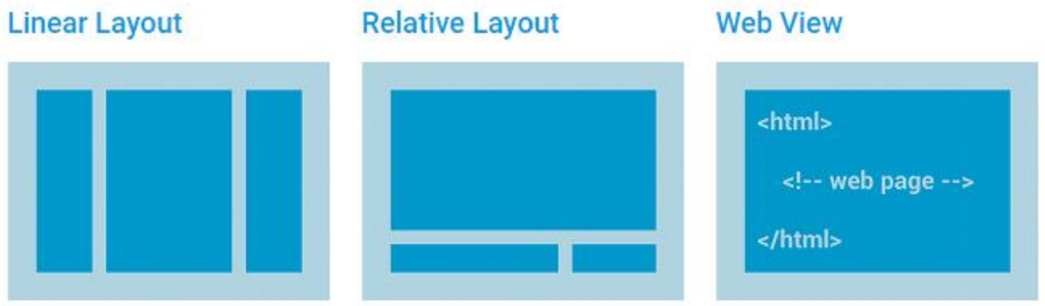


Fig 2.3 Different Layouts

Check your progress 2

1. Where does a layout placed in android?
   a. In XML file
   b. In Resource folder
   c. Not defined clearly
   d. None of these
2. What is linear layout?
   a. Placing views in rows and columns
   b. Placing views either horizontally or vertically
   c. Placing views in relative position
   d. None of these

2.4 Drawing and Working with Animation

It is found that in Android, the framework provides dual animation systems:

- property animation
- view animation

The above two animation systems are practical options, whereas the property animation system is preferred method to apply as it is flexible and offers more features. Additionally, these two systems will allow drawable animation that can be easily available to load drawable resources and will show in single frame one after another.

Notwithstanding working on what, normally such devices carried restricted power, which serves as highest rated mobile applications that carries sophisticated User Experience (UX) that carries full high quality graphics as well as animations which shows an perceptive, thus far approachable and energetic outlook. With increase in mobile applications and leading to more and more sophisticatedness, consumers will expect even high from applications.

Animation

Animation is the base of Android application as users like things which tend to move here and there on the screen. Animations appear as advance way which helps in improving user expertise for particular application that will help in standing out. The best animations are the ones that users don’t happen to see as it looks natural. Android provides the following three API’s for:

View Animation:

It serves as genuine API which Android gives. Such type of animations is fixed to particular View and will able to do simple transformations of contents regarding particular View. Being simple and ease, such API tends to be of use for things which can be an alpha animation, rotations and further.

Property Animation:

This type of animation is introduced in Android which allows an application to animate with all its features. It is of used when an change in property of an object appears, as if particular object is not visible on screen.

Drawable Animation:

It is a special Drawable resource which on application results in simple animation effect along with its layouts.

Check your progress 3

1. To change the property of an object we use_________.

   a. View Animaton

   b. Drawable Animation

   c. Property Animation

   d. All of these

2.5 Drawing 2D and 3D Graphics and Multimedia

Android is rich and carries varied framework which supports 2D and 3D graphics along with animations. In this, the graphics splits into several approaches which are utilized for 2D and 3D graphic animations. In Android, 3D graphics are present through various built in frameworks which can be OpenGL ES for OpenGL and for 3rd party frameworks as MonoGame.

It is seen that Android will have two different API's that frames 2D graphics which can be high level declarative approach while other is of programmatic low-level API:

- Drawable Resources – Such resources applies to develop custom graphics which can be programmatical or typical that embeds drawing instructions in XML files. It is noted that Drawable resources normally are explained as XML files having instructions or actions for Android which can be utilized for 2D graphic.
- Canvas – It is a sort of low level API which involves drawing directly on a bitmap and shows very fine-grained control on display.

Along with 2D graphics techniques, Android carries numerous ways to develop animations which can be:

- Drawable Animations: It is found that Android supports frame-by-frame animations which are called as Drawable Animation. It is easy and simple animation API where Android sequentially loads and show Drawable resources one after another in series just like cartoon.
- View Animations: It seems that View Animations appears as original animation API's which is present in Android along with its versions. Suich type of API are restricted which allows work with View objects and will do

simple transformations on particular Views. There are normally explained in XML files which are available in /Resources/anim folder.

- Property Animations: It is noted that in Android, such animations exists as new set of animation API's. which introduces extensible and flexible system which can be utilized to animate properties of particular object and not simply the View objects. It is flexible which allows animations to put in a nutshell in different classes which allows sharing of code in easy way.

2D Graphics

Drawable Resources are customary as well as elevated mechanism in Android applications. As with external expedients, Drawable Resources are affirmative – they're described in XML files. This access authorizes for a filter estrangement of code from reserves. This can advance execution as well as extension since it is not essential to alter code to update or convert the graphics in an Android application.

Furthermore, while Drawable Resources are advantageous for ample elementary as well as common graphic conditions, however the default the power as well as control of the Canvas API.

The extraneous approach, applying the Canvas material, is very analogous to estranged accustomed API arrangements comparable as System. Drawing or iOS's Core Drawing. Applying the Canvas material assigns the best control of how 2D graphics are developed. It is applicable for circumstances where a Drawable Resource will not perform or will be hard to work with. For exemplary, it may be necessary to draw a custom slider control whose appearance will change based on calculations related to the value of the slider.

Canvas Drawing API

Drawables are authoritative furthermore acquire their boundaries. Assured substances are both not feasible and very complicated. For exemplary, conducting a filter to a blueprint that was acquired by a camera on the appliance. It would be very burdensome to conduct red-eye compaction by utilizing a Drawable Resource. Instead, the Canvas API authorizes an application very establish grained control to selectively adjust colors in a definite fraction of the picture.

One class that is averagely utilized with the Canvas endures the Paint class. This class accepts colour as well as style information about how to illustrate on the class. It is utilized to assign things comparable a color as well as transparency.

The Canvas API utilizes the painter's archetypal to illustrate 2D graphics. Actions are conducted in successive levels on foremost of each other. Each behavior will overlay numerous domain of the basic bitmap. When the domain interlaces a subsequently painted domain the fresh paint will fractionally or entirely darken the old. This is the equivalent approach abundant other drawing APIs comparable as System.Drawing along with iOS's Core Graphics function.

Check your progress 4

1. To create 2 – D graphics in android we can use ________.
   a. Canvas
   b. Drawable resources
   c. Both of these
   d. None of these
2. Why canvas is better than drawable resource for creating graphics?
   a. It is appropriate for situations where a Drawable Resource will not work
   b. the Canvas API allows an application very find grained control
   c. The Canvas API uses the painter's model to draw 2D graphics
   d. All of these

2.6 Let Us Sum Up

While studying this unit, we have learnt that new Fragment API for android introduces in android 3.0, which permits easier dynamic user interfaces. API is application program which interface sets the routines, protocols, and tools for building software applications.

It is studied that every user interface elements in android app that are built using view and ViewGroup objects where view serves as object that draws one thing on screen which user will move while viewGroup serves as object that holds alternative View objects as define in layout of interface.

It is seen that a layout defines visual structure for a program, like UI for an activity or app convenience that you'll be able to declare in a layout by two ways

It is noted that animations are great way to improve user experience of an application which help it stand out. The best animations are the ones that users don’t notice because they feel natural.

2.7 Answers for Check Your Progress

Check your progress 1

Answers: (1 –b), (2-d)

Check your progress 2

Answers: (1 -a), (2-b)

Check your progress 3

Answers: (1 –c)

Check your progress 4

Answers: (1 -c), (2-d)

2.8 Glossary

1. **Intent -** Message object which uses to communicate with certain applications/activities asynchronously.
2. **Intent Filter -** It is an object filter application which is present in manifest file which shows system with its types of components.
3. **Animation -** Motion of objects created with software that allows moving an image in motion graphics.

2.9 Assignment

Write differences among canvas and drawable resources for creating graphics.

2.10 Activities

Design UI of a registration system.

2.11 Case Study

Study UI and Layout in android.

2.12 Further Readings

1. Learn Java for Android Development (2nd edition), Jeff Friesen, 2013.
2. Android Application Development for Java Programmers, James C. Sheusi, 2012.

Block Summary

In this block, you have learnt and understand about the basic of asynchronous messages and use of permissions for developing Android features. The block gives an idea on use of Intent in interacting with components along with extra components. You have been well explained on the various criteria's and rules to be followed for permission of rights.

The block detailed about the basic of visual structure for a program along with usability of AndroidManifest.xml file. The concept related to User Interface with respect to activity or applications are also explained to you. You will be demonstrated practically about 2D and 3D animations in Android.

Block Assignment

Short Answer Questions

1. How animation can be applied in android?
2. What is the difference between intent and intent filters
3. What is view group?
4. What is meant by permission and how it can be provided?

Long Answer Questions

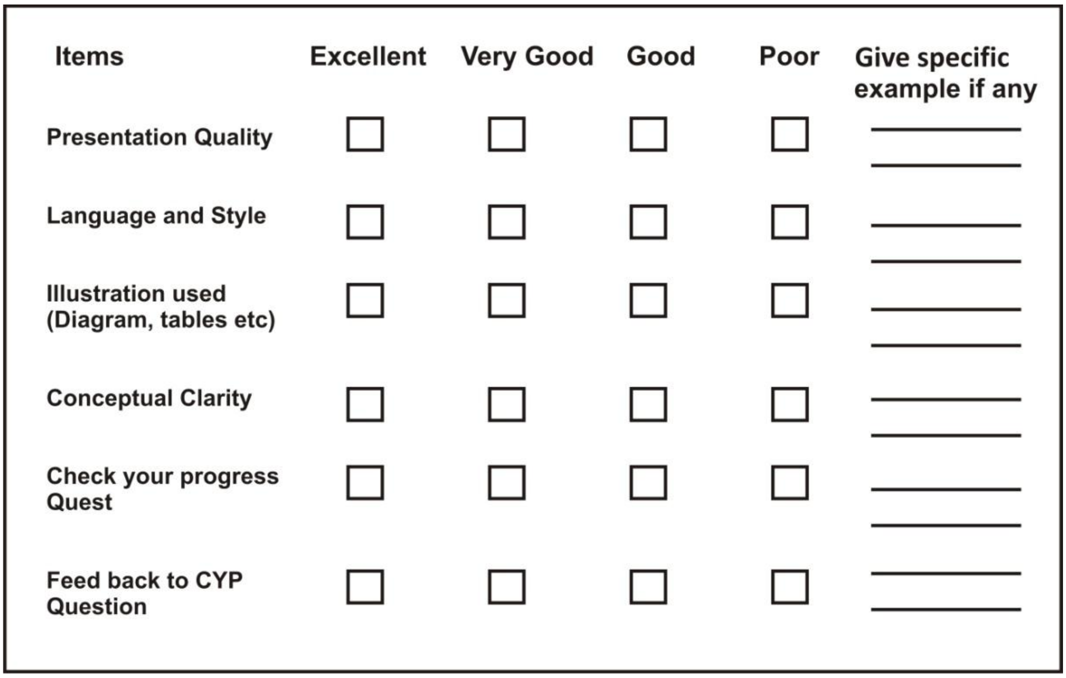
1. What is android manifest file? Explain in detail by providing its structure.
2. What is the difference between relative layout and linear layout?
3. Create a login form and study its manifest file.
4. How can we manage different type’s application resources in android?

Enrolment No. ▭

1. How many hours did you need for studying the units?

Unit No	**1**	**2**	**3**	**4**
Nos of Hrs				

2. Please give your reactions to the following items based on your reading of the block:



Items	Excellent	Very Good	Good	Poor	Give specific example if any
Presentation Quality	□	□	□	□	________ ________
Language and Style	□	□	□	□	________ ________
Illustration used (Diagram, tables etc)	□	□	□	□	________ ________
Conceptual Clarity	□	□	□	□	________ ________
Check your progress Quest	□	□	□	□	________ ________
Feed back to CYP Question	□	□	□	□	________ ________

3. Any Other Comments

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

> Education is something
> which ought to be
> brought within
> the reach of every one.

- Dr. B. R. Ambedkar

Dr. Babasaheb Ambedkar Open University
Jyotirmay' Parisar, Opp. Shri Balaji Temple, Sarkhej-Gandhinagar Highway, Chharodi,
Ahmedabad-382 481.

MOBILE APPLICATION DEVELOPMENT

PGDCA 203

BLOCK 3:
ANDROID NETWORKING AND DEVELOPMENT

Dr. Babasaheb Ambedkar Open University
Ahmedabad

MOBILE APPLICATION DEVELOPMENT

Knowledge Management and
Research Organization
Pune

Editorial Panel

Author
Er. Nishit Mathur

Language Editor
Prof. Jaipal Gaikwad

Graphic and Creative Panel
Ms. K. Jamdal
Ms. Lata Dawange
Ms. Pinaz Driver
Ms. Ashwini Wankhede
Mr. Kiran Shinde
Mr. Prashant Tikone
Mr. Akshay Mirajkar

Copyright © 2015 Knowledge Management and Research Organization.
All rights reserved. No part of this book may be reproduced, transmitted or utilized in any form or by a means, electronic or mechanical, including photocopying, recording or by any information storage or retrieval system without written permission from us.

Acknowledgment

Every attempt has been made to trace the copyright holders of material reproduced in this book. Should an infringement have occurred, we apologize for the same and will be pleased to make necessary correction/amendment in future edition of this book.

The content is developed by taking reference of online and print publications that are mentioned in Bibliography. The content developed represents the breadth of research excellence in this multidisciplinary academic field. Some of the information, illustrations and examples are taken "as is" and as available in the references mentioned in Bibliography for academic purpose and better understanding by learner.'

ROLE OF SELF INSTRUCTIONAL MATERIAL IN DISTANCE LEARNING

The need to plan effective instruction is imperative for a successful distance teaching repertoire. This is due to the fact that the instructional designer, the tutor, the author (s) and the student are often separated by distance and may never meet in person. This is an increasingly common scenario in distance education instruction. As much as possible, teaching by distance should stimulate the student's intellectual involvement and contain all the necessary learning instructional activities that are capable of guiding the student through the course objectives. Therefore, the course / self-instructional material are completely equipped with everything that the syllabus prescribes.

To ensure effective instruction, a number of instructional design ideas are used and these help students to acquire knowledge, intellectual skills, motor skills and necessary attitudinal changes. In this respect, students' assessment and course evaluation are incorporated in the text.

The nature of instructional activities used in distance education self-instructional materials depends on the domain of learning that they reinforce in the text, that is, the cognitive, psychomotor and affective. These are further interpreted in the acquisition of knowledge, intellectual skills and motor skills. Students may be encouraged to gain, apply and communicate (orally or in writing) the knowledge acquired. Intellectual-skills objectives may be met by designing instructions that make use of students' prior knowledge and experiences in the discourse as the foundation on which newly acquired knowledge is built.

The provision of exercises in the form of assignments, projects and tutorial feedback is necessary. Instructional activities that teach motor skills need to be graphically demonstrated and the correct practices provided during tutorials. Instructional activities for inculcating change in attitude and behavior should create interest and demonstrate need and benefits gained by adopting the required change. Information on the adoption and procedures for practice of new attitudes may then be introduced.

Teaching and learning at a distance eliminates interactive communication cues, such as pauses, intonation and gestures, associated with the face-to-face method of teaching. This is particularly so with the exclusive use of print media. Instructional activities built into the instructional repertoire provide this missing interaction between the student and the teacher. Therefore, the use of instructional activities to affect better distance teaching is not optional, but mandatory.

Our team of successful writers and authors has tried to reduce this.

Divide and to bring this Self Instructional Material as the best teaching and communication tool. Instructional activities are varied in order to assess the different facets of the domains of learning.

Distance education teaching repertoire involves extensive use of self-instructional materials, be they print or otherwise. These materials are designed to achieve certain pre-determined learning outcomes, namely goals and objectives that are contained in an instructional plan. Since the teaching process is affected over a distance, there is need to ensure that students actively participate in their learning by performing specific tasks that help them to understand the relevant concepts. Therefore, a set of exercises is built into the teaching repertoire in order to link what students and tutors do in the framework of the course outline. These could be in the form of students' assignments, a research project or a science practical exercise. Examples of instructional activities in distance education are too numerous to list. Instructional activities, when used in this context, help to motivate students, guide and measure students' performance (continuous assessment)

PREFACE

We have put in lots of hard work to make this book as user-friendly as possible, but we have not sacrificed quality. Experts were involved in preparing the materials. However, concepts are explained in easy language for you. We have included may tables and examples for easy understanding.

We sincerely hope this book will help you in every way you expect.

All the best for your studies from our team!

Mobile Application Development

Contents

BLOCK 3: ANDROID NETWORKING AND DEVELOPMENT

UNIT 1 ADVANCED TOPICS - I

Android Networking, Web and Telephony API, Search, Location and Mapping, Sensors, NFC, Speech, Gestures, and Accessibility,

UNIT 2 ADVANCED TOPICS - II

Communication, Identity, Sync, and Social Media, The Android Native Development Kit (NDK)

BLOCK 4: ANDROID APPLICATION PUBLISHING AND CONTENT PROVIDERS

UNIT 1 MORE ON ANDROID

Handling and Persisting Data, A Content Provider as a Facade for a RESTful Web Service, Using Content Providers

UNIT 2 PUBLISHING ANDROID APPLICATION

Deploying Android Application to the World, Selling your Android application

Dr. Babasaheb
Ambedkar
Open University

PGDCA-203

MOBILE APPLICATION DEVELOPMENT

BLOCK 3: ANDROID NETWORKING AND DEVELOPMENT

BLOCK 3: ANDROID NETWORKING AND DEVELOPMENT

Block Introduction

Android accommodates the common Java mesh java.net package which can be facilitated to approach mesh reserves. The Android mesh commenced almost three ages ago further as we accomplish the foremost work plan, this after-effect summaries on several of our key outputs as well as effects, while additionally highlighting the network's existence at a major comprehensive event that will authorize adoption of the post-2015 arrangement for misery risk shrinkage.

In this block, we will detail about the basic of Java mesh java.net package and Android network system. The block will focus on the study and concept of Web Telephony API had been developed by Mozilla with administering Firefox OS telephony abilities for API. You will get an idea on networking in Android with certain specifications.

In this block, you will make to learn and understand about LocationManager class that showing execution to Android location service. The concept related to LocationProvider class will also be explained to you. You will be demonstrated practically about Android phones gesture.

Block Objective

After learning this block, you will be able to understand:

- About Android Networking
- Idea about Web and Telephony API
- Features of NFC, Speech, Gestures
- Characteristics about Accessibility and Communication
- Concept of Android Native Development Kit (NDK)

Block Structure

Unit 1: Advanced Topics - I

Unit 2: Advanced Topics - II

UNIT 1: ADVANCED TOPICS - I

Unit Structure

1.0 Learning Objectives

After learning this Unit, you will be able to understand:

- About Networking with Android
- About Web API
- About Search, Location and Mapping
- About NFC, Speech and Gestures

1.1 Introduction

Android accommodates the common Java mesh java.net package which can be facilitated to approach mesh reserves. The base class for HTTP mesh approach in the java.net package continues the HttpURLConnection class. Practicing mesh actions on Android can be complex. You have to unplug as well as secure connections, so as to allow caches as well as affirm to practice the network activity in a background thread.

1.2 Android Networking

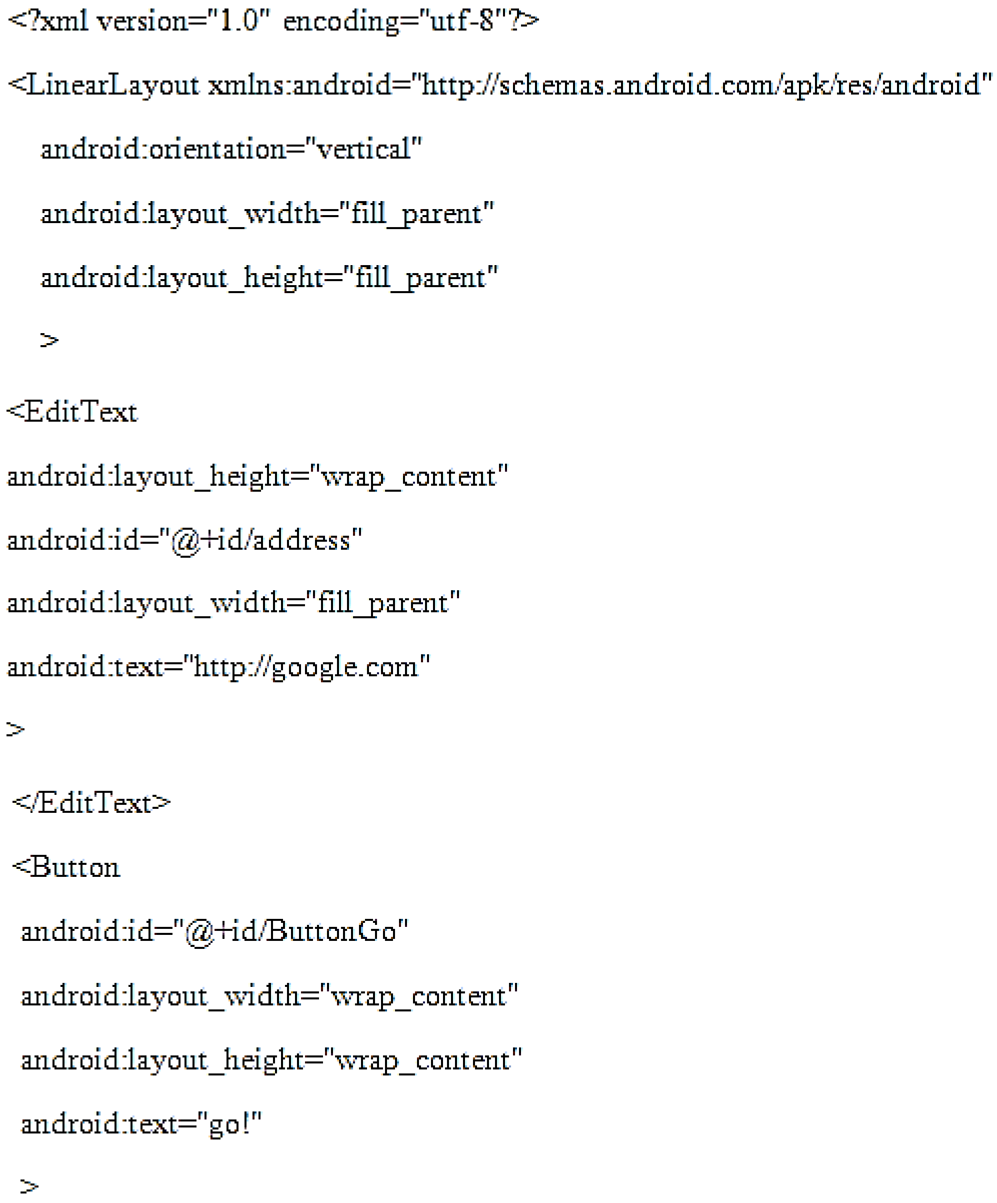
The Android mesh commenced almost three ages ago further as we accomplish the foremost work plan, this after-effect summaries on several of our key outputs as well as effects, while additionally highlighting the network's existence at a major comprehensive event that will authorize adoption of the post-2015 arrangement for misery risk shrinkage. Essential outputs of the ANDROID intersection are being demonstrated at the Third United Nations World Conference on Disaster Risk compaction that will be held from 14 to 18 March 2015 in Sendai City, Miyagi Prefecture, Japan. The consequence explains several of the essential consequences along with actions where the ANDROID mesh will be approximated. ANDROID's existence at an issue like this is hardy to assure that our task as well as considerations is conveniently depicted in major comprehensive policy formulations along with the post-2015 arrangement.

Networking in android describes the aptitude to transmit as well as accept information from remote server. This information can be coupled a plain text, xml, json, appearance or a video flow. Android originally upholds dual HTTP clients for networking, one by facilitating Apache Http Client as well as other utilizing HttpURLConnection. Applications transcribed with networking elements are beyond more enterprising as well as content wealthy than those that are not. Applications leverage the mesh for multiformity of synthesizes: to contribute current as well as updated content, to allow social networking elements of an otherwise standalone application, to offload massive processing to heavyweight servers, furthermore to authorize data accumulation beyond what the user can capture on the device.

Android allows your application adheres to the internet or several other local network furthermore authorizes you to practice network actions. A device can hold different categories of network attachments. Before you practice some network actions, you need initial examine that are you joined to that network or internet e.t.c. For this android assigns ConnectivityManager class. You expect to instantiate an object of this class by designating getSystemService() method.

Once you instantiate the object of ConnectivityManager class, you can utilize getAllNetworkInfo approach to apprehend the details of complete the networks. After checking that you are connected to the internet, you can conduct some network activity.

Android allows HttpURLConnection as well as URL class to handle these actions. You hope to instantiate an object of URL class by assigning the sequence of website. After that you expect to call openConnection mechanism of url class as well as accept it in a HttpURLConnection object. After that you need to call the connect approach of HttpURLConnection class.

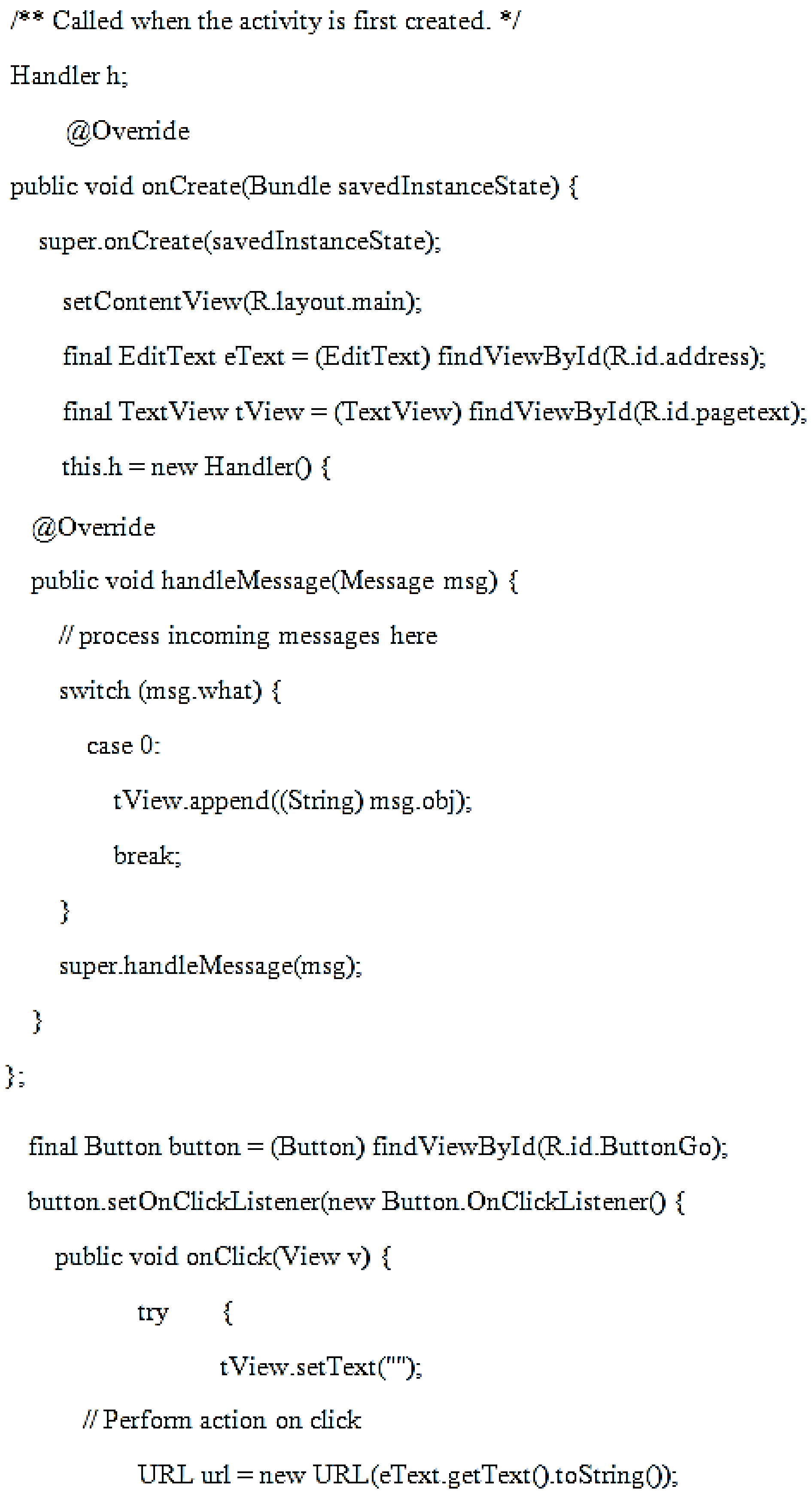


```
<?xml version="1.0"  encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<EditText
android:layout_height="wrap_content"
android:id="@+id/address"
android:layout_width="fill_parent"
android:text="http://google.com"
>
 </EditText>
 <Button
  android:id="@+id/ButtonGo"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="go!"
  >
```

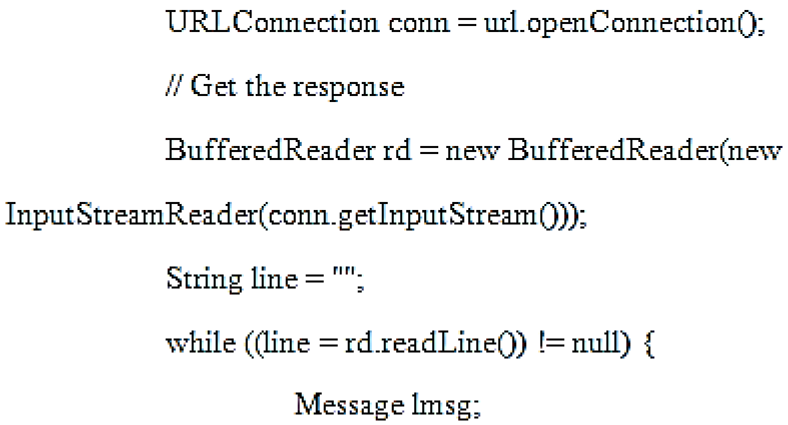
```
</Button>
<TextView
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   android:background="#ffffff"
   android:textColor="#000000"
         android:id="@+id/pagetext"
   />
</LinearLayout>
```

Listing 2 the program below describes Java code applied in above example.

```
package com.msi.getwebpage;
import android.app.Activity;
import android.os.Bundle;
// used for interacting with user interface
import android.widget.Button;
import android.widget.TextView;
import android.widget.EditText;
import android.view.View;
// used for passing data
import android.os.Handler;
import android.os.Message;
// used for connectivity
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.URL;
import java.net.URLConnection;
public class GetWebPage extends Activity {
```



```
/** Called when the activity is first created. */
Handler h;
     @Override
public void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
     setContentView(R.layout.main);
     final EditText eText = (EditText) findViewById(R.id.address);
     final TextView tView = (TextView) findViewById(R.id.pagetext);
     this.h = new Handler() {
  @Override
  public void handleMessage(Message msg) {
     // process incoming messages here
     switch (msg.what) {
       case 0:
          tView.append((String) msg.obj);
          break;
     }
     super.handleMessage(msg);
  }
};
  final Button button = (Button) findViewById(R.id.ButtonGo);
  button.setOnClickListener(new Button.OnClickListener() {
     public void onClick(View v) {
            try     {
                    tView.setText("");
       // Perform action on click
            URL url = new URL(eText.getText().toString());
```



```
                    URLConnection conn = url.openConnection();
                    // Get the response
                    BufferedReader rd = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
                    String line = "";
                    while ((line = rd.readLine()) != null) {
                            Message lmsg;
```

```
                      Message lmsg;
                   lmsg = new Message();
                   lmsg.obj = line;
                   lmsg.what = 0;
                   GetWebPage.this.h.sendMessage(lmsg);
                 }
                }
                catch (Exception e)      {
                }
             }
          });
       }
}
```

We see that code is distributed in general areas where many import statements will reference UI, data passing and networking classes. It is seen that every code with network is available in OnClick method of OnClickListener.

In this, URL and URLConnection classes team will show real connectivity to Web site of user's. Since every line is readable, text gets appended to TextView where data is assigned directly to TextView.

We see that Android application is working with HTTP Web server like Apache or Internet Information Server where an application is directly involved to TCP socket rather than HTTP.

Check your progress 1

1. Which class is required to check the availability of network connections?
    a. Main class
    b. ConnectivityManager class
    c. Activity class
    d. None of these
2. To perform network operations android needs________________.
    a. HttpURLConnection
    b. URL
    c. Both of these
    d. None of these

1.3 Web and Telephony API

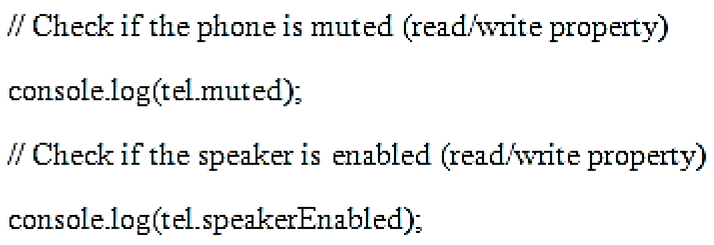
Telephony

Telephony suppliers carry data and information regarding the phone operation, which can be particularly SMS as well as MMS messages which will access the APN list, having MMSC that can be applied further. Such APIs are not present in every Android-powered devices. If your application rely on telephony characteristics like managing of SMS messages using <uses-feature> element that is present in manifest which shows "android.hardware.telephony" feature.

We see that the initial point for working with device phone features results in Navigator.mozTelephony. It is seen that when a reference to object is made, it can be started by placing and receiving calls. The syntax for Telephony object is:

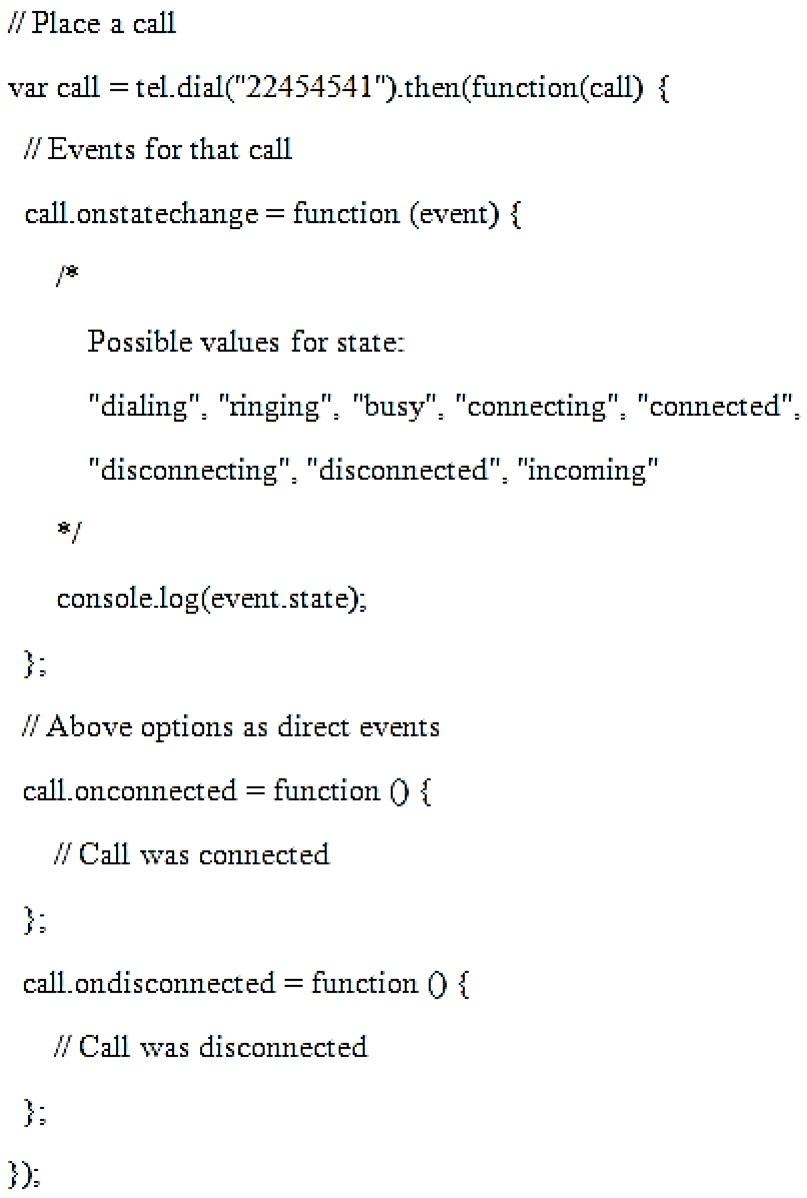
```
var tel = navigator.mozTelephony;
```

It is found that resulting Telephony object will show telephone hardware programmatically that describes quality to handle numerous features of it, such as it can be mute/unmute furthermore enable/disable the speaker phone as described from the syntax:



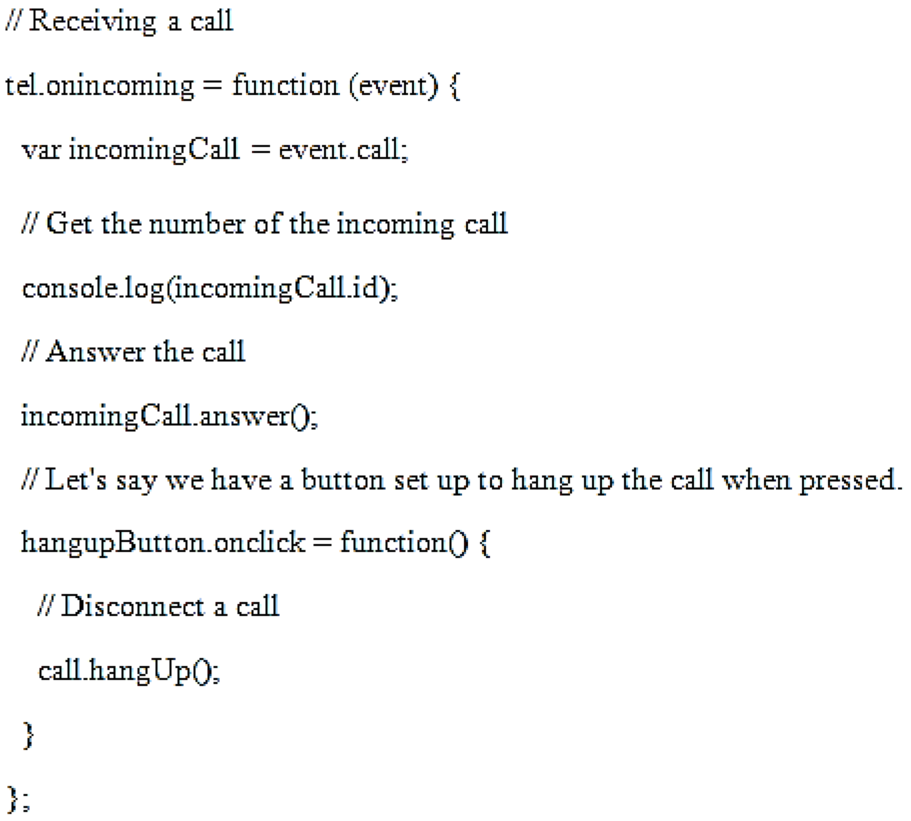
```
// Check if the phone is muted (read/write property)
console.log(tel.muted);
// Check if the speaker is enabled (read/write property)
console.log(tel.speakerEnabled);
```

It is found that doing a call is simple as calling Telephony.dial on Telephony object which is promise based API that can easily solve using TelephonyCall object showing the call. Such object has many properties, methods and allows handlers to keep record on call's properties which does actions related to hanging up and holding call along with reacting to certain changes in call state. The program below describes placing a telephone call:



```
// Place a call
var call = tel.dial("22454541").then(function(call) {
  // Events for that call
  call.onstatechange = function (event) {
    /*
      Possible values for state:
      "dialing", "ringing", "busy", "connecting", "connected",
      "disconnecting", "disconnected", "incoming"
    */
    console.log(event.state);
  };
  // Above options as direct events
  call.onconnected = function () {
    // Call was connected
  };
  call.ondisconnected = function () {
    // Call was disconnected
  };
});
```

Apart from calling, we see that receiving of calls is entirely different where we have to write Telephony.onincoming event listener which ask when call is incoming and has event object such as CallEvent having call property which gives access to call's TelephonyCall object and does work as answering the call etc. The program below describes the code for receiving of call:

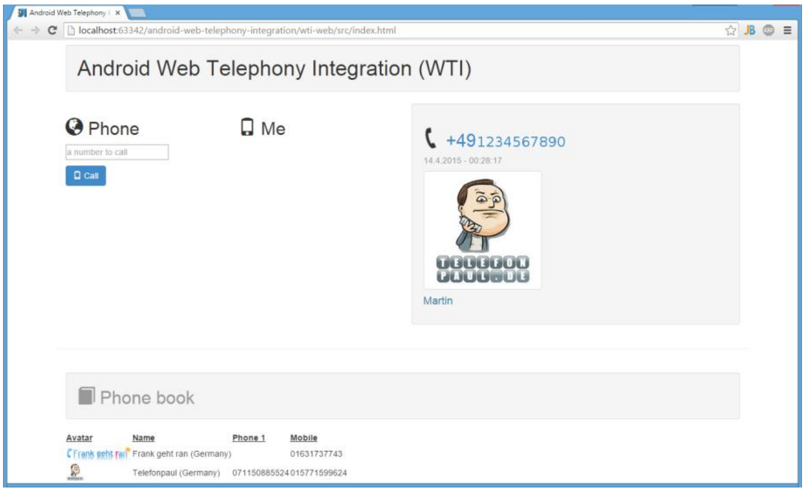


```
// Receiving a call
tel.onincoming = function (event) {
  var incomingCall = event.call;
  // Get the number of the incoming call
  console.log(incomingCall.id);
  // Answer the call
  incomingCall.answer();
  // Let's say we have a button set up to hang up the call when pressed.
  hangupButton.onclick = function() {
    // Disconnect a call
    call.hangUp();
  }
};
```

Web Telephony

The Web Telephony API had been developed by Mozilla to appease the constraints for an API to administer Firefox OS telephony abilities. It is an internal-only API, wherefore assigning 3rd party developers to benefit control up over alike a considerable device application is a major protection concern. The API's entrance point is via the Navigator.mozTelephony assets. This reverses a Telephony object that assigns advance to the device's telephony mechanism, with approaches as well as effects for directing the microphone along with speakerphone, bringing about dial tones (DTMF), starting as well as receiving calls, besides more. When a call is commenced, both commenced by the device facilitating Telephony.dial(), or received from nonexistent via the Telephony.onincoming handler, a TelephonyCall object is developed to approximate the call. This can be contacted via its effects as well as approaches to authorize methodical answering, hanging up, as well as putting on pause of calls, acknowledging to contrasting call states, additionally much more.

Web Telephony Integration (WTI) is an equivalent Computer Telephony Integration CTI furthermore for Web, which serves as proof of concept (PoC) demonstrating combination of mobile phones into web function which assists in determining numerous problems. It enables CRM consumers to call a consumer by a click in browser/web app, or vice versa where all incoming call gets automatically accessible to popup, exhibiting caller connection data. This confirmation of idea brings about utilize of continuing Internet infrastructure along with facilitates to competitive realtime app platform solutions which will advance phone based workflows.



Android Web Telephony
localhost:63342/android-web-telephony-integration/wti-web/src/index.html

Android Web Telephony Integration (WTI)

Phone

a number to call

Call

Me

+491234567890

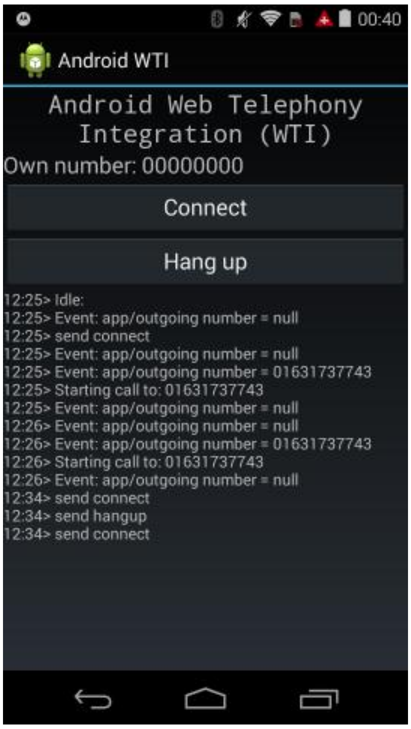
14.4.2015 - 00:28:17

Martin

Phone book

Avatar	Name	Phone 1	Mobile
Frank geht ran	Frank geht ran (Germany)		01631737743
	Telefonpaul (Germany)	071150885524	015771599624

Fig 1.1 Web Telephony integration



Android Web Telephony Integration (WTI)

Own number: 00000000

Connect

Hang up

```
12:25> Idle:
12:25> Event: app/outgoing number = null
12:25> send connect
12:25> Event: app/outgoing number = null
12:25> Event: app/outgoing number = 01631737743
12:25> Starting call to: 01631737743
12:25> Event: app/outgoing number = null
12:26> Event: app/outgoing number = null
12:26> Event: app/outgoing number = 01631737743
12:26> Starting call to: 01631737743
12:26> Event: app/outgoing number = null
12:34> send connect
12:34> send hangup
12:34> send connect
```

Fig 1.2 Web hangout

Working:

Consider the diagram shown in fig 1.3:

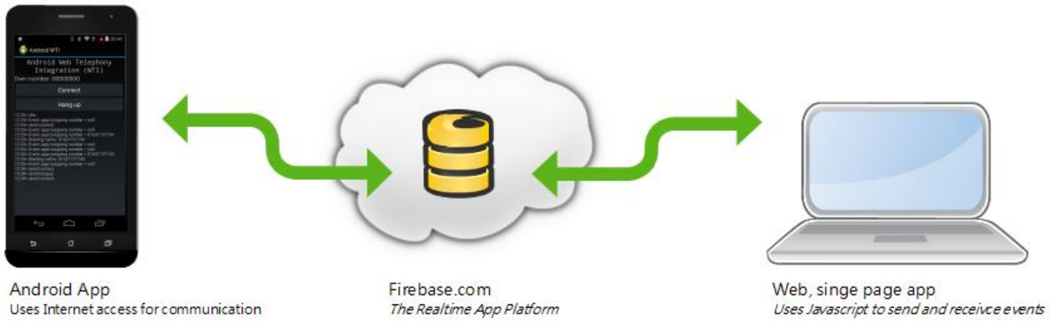

Android App
Uses Internet access for communication
Firebase.com
The Realtime App Platform
Web, singe page app
Uses Javascript to send and receivce events

Fig 1.3 Web telephony arrangement

In the above figure we see that:

- The firebase describes sharing of document
- In this, the application gets connected to Firebase and further will enroll callback handlers on nodes
- Also, the Web app connects to Firebase and registers callback handlers on certain nodes
- Here, reading and writing of document nodes needs application communication among each other
- In this, technical protocol related to publishing and subscribing events gets controlled in Firebase clients

Check your progress 2

1. Which command is used to check whether the phone is muted or not?

   a. console.log(tel.muted);

   b. console.log(tel.speakerEnabled);

   c. console.log(incomingCall.id)

   d. None of these

2. What is the full form of WTI?

   a. Web technology integration

   b. Web Telephony Integration

   c. World Telephony Integration

   d. None of these

1.4 Search, Location and Mapping

Search

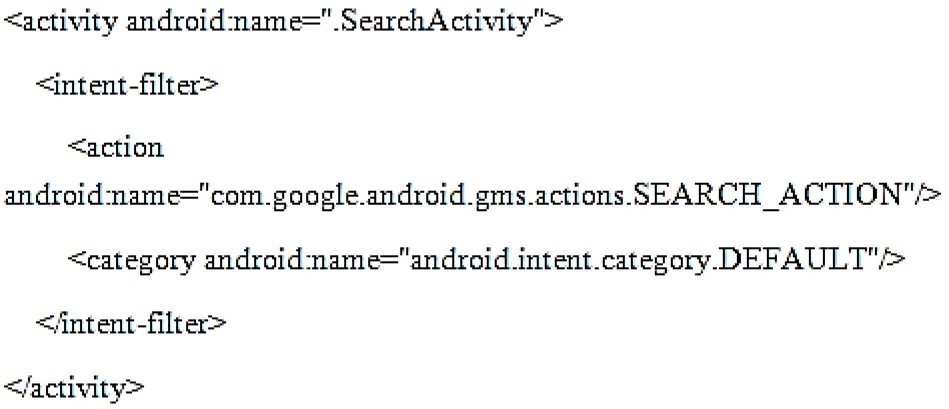
In Android, to handle search in context of application, you have to declare an intent filter in app having SEARCH_ACTION action which is described by the syntax shown below:

"com.google.android.gms.actions.SEARCH_ACTION"

Support search queries from Google Now.

QUERY

A Query is a string which carries search query. Consider the example program of an Indent filter:



```
<activity android:name=".SearchActivity">
   <intent-filter>
      <action
android:name="com.google.android.gms.actions.SEARCH_ACTION"/>
      <category android:name="android.intent.category.DEFAULT"/>
   </intent-filter>
</activity>
```

Web search:

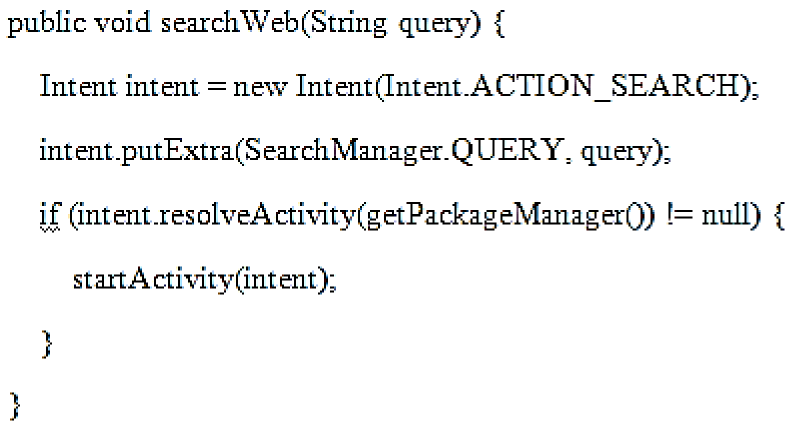
To work with web search, you have to use ACTION_WEB_SEARCH action function where you have to show search string in SearchManager.QUERY extra.

ACTION_WEB_SEARCH

- Data URI Scheme
- None
- MIME Type
- None
- Extras

SearchManager.QUERY

Consider an example of search string using an intent:



```
public void searchWeb(String query) {
    Intent intent = new Intent(Intent.ACTION_SEARCH);
    intent.putExtra(SearchManager.QUERY, query);
    if (intent.resolveActivity(getPackageManager()) != null) {
        startActivity(intent);
    }
}
```

Location:

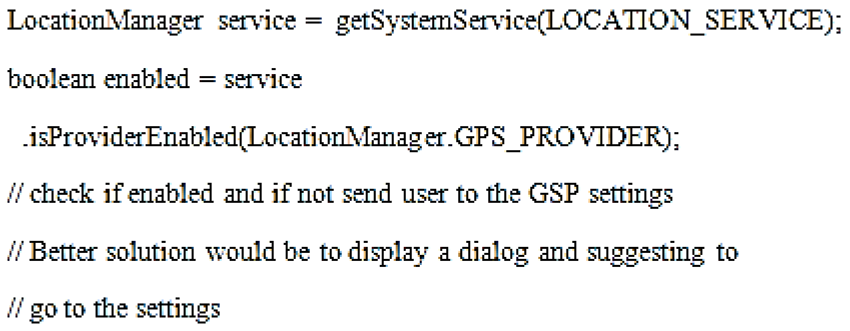
In Android there is another great feature which is about location of user. It is seen that mobile users will take their devices everywhere, which with this becomes easy to trace them with location application. For this, there is need of location APIs which is present in Google Play services that helps on adding location will allow to locate the user through automated location tracking, geofencing, and activity recognition activities.

Numerous Android devices helps in finding present geolocation of user with the use of Global Positioning System that works through tower triangulation or with wifi networks. We see that Android contains android.location package that will help in showing API to find present geo position of mobile or user.

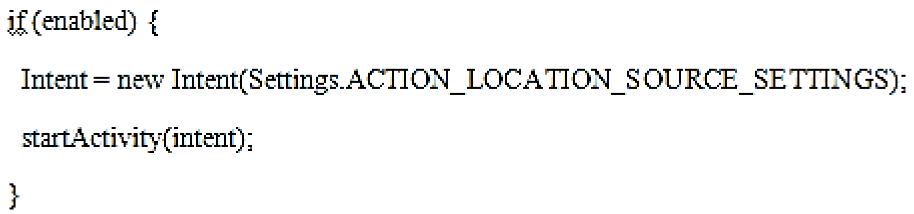
LocationManager

The LocationManager class provides access to the Android location service. This services allows to access location providers, to register location update listeners and proximity alerts and more.

You can find out, if a LocationManager is enabled via the is ProviderEnabled() method. If its not enabled you can send the user to the settings via an Intent with the Settings.ACTION_LOCATION_SOURCE_SETTINGS action for the android.provider.Settings class.



```
LocationManager  service =  getSystemService(LOCATION_SERVICE);
boolean enabled = service
  .isProviderEnabled(LocationManager.GPS_PROVIDER);
// check if enabled and if not send user to the GSP settings
// Better solution would be to display a dialog and suggesting to
// go to the settings
```



```
if (enabled) {
  Intent = new Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
  startActivity(intent);
}
```

Typically you would open an Alarm Dialog prompt the user and if he wants to enable GPS or if the application should be cancelled.

LocationProvider

The LocationProvider class is the super class of the different location providers which deliver the information about the current location. This information is stored in the Location class. The Android device might have several LocationProvider available and you can select which one you want to use. In most cases you have the followng LocationProvider available.

The Google perform activities location APIs are adopted over and above the Android composition location APIs (android.location) as a mechanism of accumulating location alertness to your app. If you are contemporarily facilitating the Android composition location APIs, you are potently catapulted to switch to the Google Play supports location APIs as shortly as feasible.

This class displays you how to utilize the Google Play activities location APIs in your app to acquire the modern location, accumulate frequent location updates, as well as behold up addresses. The class constitutes model apps along with catchword snippets that you can utilize as a beginning point for accumulating location alertness to your app.

Android assigns your applications approach to the location services acknowledged by the device pierced classes in the android.location package. The innermost element of the location arrangement is the LocationManager system operation, which assigns APIs to assess location as well as bearing of the elementary device.

As with other system services, you do not instantiate a LocationManager directly. Rather, you request an instance from the system by calling getSystemService(Context.LOCATION_SERVICE). The method returns a handle to a new LocationManager instance.

Once your application has a LocationManager, your application is able to do three things:

- Query for the list of all LocationProviders for the last known user location.

- Register/unregister for periodic updates of the user's current location from a location provider.
- Register/unregister for a given Intent to be fired if the device comes within a given proximity.

Maps:

With the Google Maps Android API, you can accumulate maps to your app that are based on Google Maps details. The API automatically undergoes approach to Google Maps servers, details downloading, map demonstrates, as well as experience motions on the map. You can additionally facilitate API calls to accumulate indications, polygons as well as envelops, also to alter the user's look of a precise map area. The core class in Google Maps Android API is MapView which shows map information acquired from Google Maps exercise. When the MapView acquires essence, it will apprehend key presses additionally and experience gestures to pan along with zooming of map automatically, along with experiencing crossway demands for increased maps tiles. It additionally assigns all of the UI constituents essential for users to administer the map. Your application can additionally utilize MapView class approaches in order to command the map systematically and furthermore will illustrate number of envelops on top of the map. The Google Maps Android APIs are not comprised in the Android platform, furthermore are obtainable on several device with the Google Play Store going Android 2.2 or higher using Google Play activities.

Check your progress 3

1. Which class is used to show map in android API?
   a. Activity class
   b. MapView class
   c. Intent
   d. All of these
2. Location manager is used to ______________.
   a. access location providers
   b. register location
   c. update listeners
   d. All of these

1.5 Sensors, NFC, Speech, Gestures

Near Field Communication

Near Field Communication (NFC) exists a collection of short-range wireless technologies, unconditionally stipulating a distance of 4cm or lower than to commence a correlation. NFC apportions you to share little payloads of information between an NFC tag as well as an Android-powered appliance, or between dual Android-powered apparatuses. Tags can range in difficulty. Simple tags advance just read as well as formulate semantics, sometimes with one-time-programmable domains to create the card read-only. More complicated tags offer math operations, along with acquire cryptographic hardware to authorize approach to a sector. The foremost refined tags cover functioning environments, assigning complex interactions with code accomplishing on the tag. The data accumulated in the tag can additionally be composed in a variation of formats, but numerous of the Android arrangement APIs are based around a NFC Forum standard labeled NDEF (NFC Data Exchange Format). Android-powered devices with NFC simultaneously support three main modes of operation:

- Reader/writer mode, allowing the NFC device to read and/or write passive NFC tags and stickers.
- P2P mode, allowing the NFC device to exchange data with other NFC peers; this operation mode is used by Android Beam.
- Card emulation mode, allowing the NFC device itself to act as an NFC card. The emulated NFC card can then be accessed by an external NFC reader, such as an NFC point-of-sale terminal.

Before you commence writing your NFC applications, it continues consequential to comprehend the contrasting types of NFC tags, how the tag express mechanism parses NFC tags, along with the definite work that the tag deliver system conducts when it determines an NDEF message. NFC tags advance in a broad array of technologies as well as can also have data transcribed to them in numerous different approaches. Android acquires the better uphold for the NDEF standard, which is described nearby the NFC Forum.

NDEF deposition is encapsulated inside an acknowledgment (NdefMessage) that holds one or additional records (NdefRecord). Each NDEF indicate demand be well-formed according to the specification of the category of record that you expect to develop. Android additionally upholds external categories of tags that do not enclose NDEF data, which you can act with immediate utilizing the classes in the android.nfc.tech package. Working with these external types of tags encompasses formulating your own protocol stack to broadcast with the tags, so we advise utilizing NDEF when possible for alleviate of development as well as maximum benefit for Android-powered appliances.

When the tag post system is complete developing an intent that encapsulates the NFC tag additionally its disclosing information, it conveys the intent to an absorbed application that filters for the intent. If additional than one application can experience the intent, the Activity Chooser is advanced because the user can choose the Activity. The tag dispatch system defines three intents, which are listed in order of highest to lowest priority:

- ACTION_NDEF_DISCOVERED: It is applied for Activity having tag with NDEF payload that gets scanned and recognized. It is the top priority intent where tag delivers system that helps in starting an Activity before other intent.
- ACTION_TECH_DISCOVERED: It is applied when there is no activity related to handling of ACTION_NDEF_DISCOVERED intent, it will deliver in order to begin with an application. This intent can be applied straight without starting ACTION_NDEF_DISCOVERED first and on scanning will have NDEF data which cannot be mapped to MIME type or URI.
- ACTION_TAG_DISCOVERED: This intent is started if no activities handle the ACTION_NDEF_DISCOVERED or ACTION_TECH_DISCOVERED intents.

Sensor:

We see that sensors will able to control and monitor 3D device movement else also to alter in particular environment of device. Android provides sensor api to work with different types of sensors such as:

1) Motion Sensors: These will measure acceleration forces as well as rotational forces using all axes.

2) Position Sensors: It is applied to calculate physical position of device.

3) Environmental Sensors: It will find environmental changes which can be temperature, humidity etc.

Android Sensor API

It is noted that Android sensor API will show different classes and interface which can be:

1) Sensor Manager class: This will handle following methods:

- to get sensor instance,
- to access and list sensors,
- to register and unregister sensor listeners etc.

It is achieved by calling the method getSystemService() and passing the SENSOR_SERVICE constant in it.

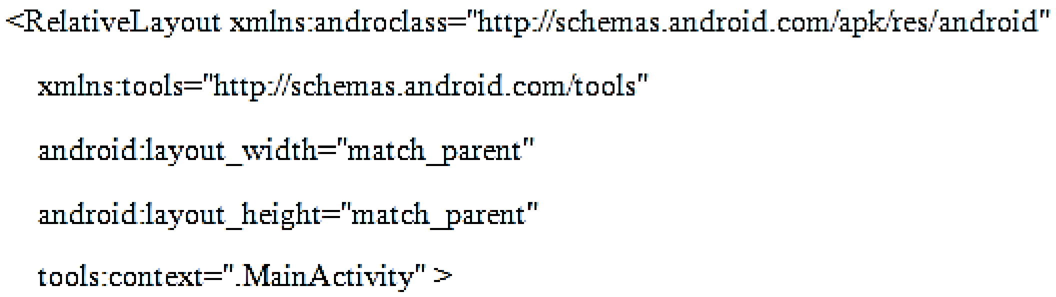
SensorManager sm = (SensorManager)getSystemService(SENSOR_SERVICE);

2) Sensor class: The android.hardware.Sensor class will show methods to receive information about sensors which could be sensor name, sensor type, sensor resolution, sensor type etc.

3) SensorEvent class: Its instance is created by the system. It provides information about the sensor.

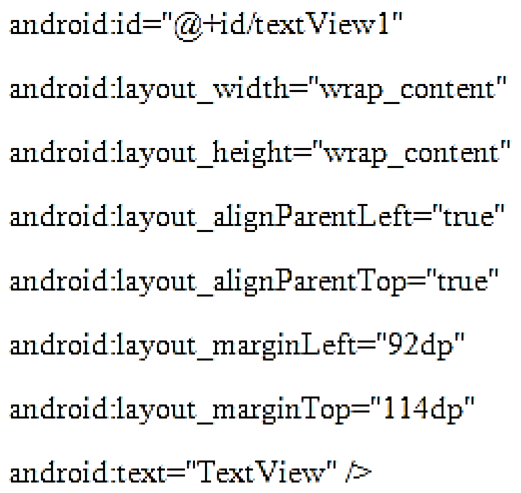
4) SensorEventListener interface: It delivers dual call back methods in order to have information about sensor regarding sensor values (x,y and z) change or sensor accuracy changes.

File: activity_main.xml



```
<RelativeLayout xmlns:androclass="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity" >
```

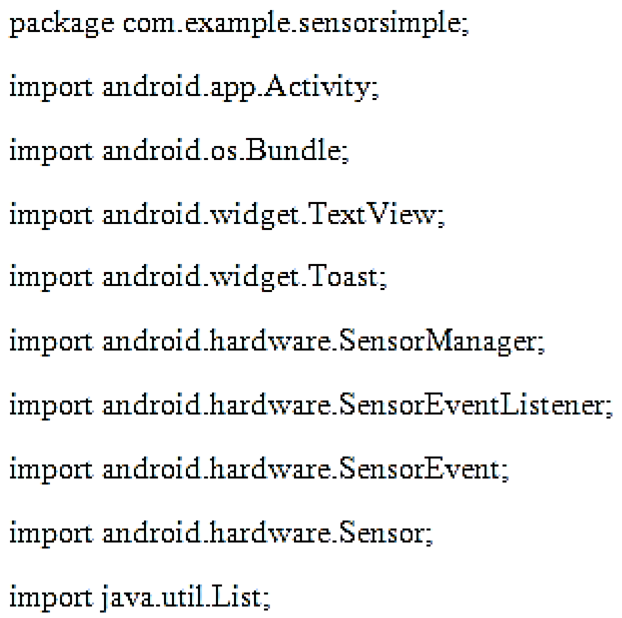
```
    <TextView
```



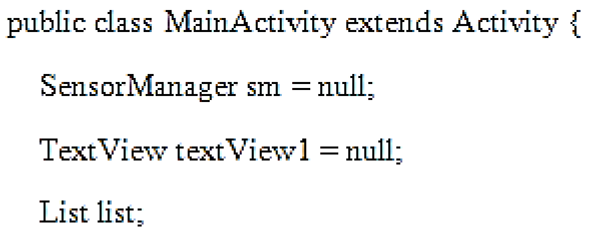
```
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginLeft="92dp"
    android:layout_marginTop="114dp"
    android:text="TextView" />
```

</RelativeLayout>

Activity class: Consider a code which gives values of x axis, y axis and z axis.



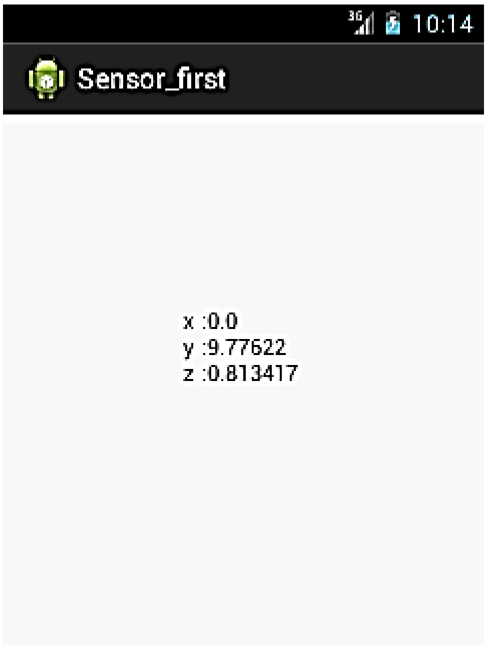
```
package com.example.sensorsimple;
import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;
import android.widget.Toast;
import android.hardware.SensorManager;
import android.hardware.SensorEventListener;
import android.hardware.SensorEvent;
import android.hardware.Sensor;
import java.util.List;
```



```
public class MainActivity extends Activity {
    SensorManager sm = null;
    TextView textView1 = null;
    List list;
```

```
  SensorEventListener sel = new SensorEventListener(){
   public void onAccuracyChanged(Sensor sensor, int accuracy) {}
   public void onSensorChanged(SensorEvent event) {
      float[] values = event.values;
      textView1.setText("x: "+values[0]+"\ny: "+values[1]+"\nz: "+values[2]
   }
 };
   @Override
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      /* Get a SensorManager instance */
      sm = (SensorManager)getSystemService(SENSOR_SERVICE);
      textView1 = (TextView)findViewById(R.id.textView1);
      list = sm.getSensorList(Sensor.TYPE_ACCELEROMETER);
      if(list.size()>0){
     sm.registerListener(sel, (Sensor) list.get(0),
 SensorManager.SENSOR_DELAY_NORMAL);
      }else {
         Toast.makeText(getBaseContext(), "Error: No Accelerometer.",
 Toast.LENGTH_LONG).show();
      }
   }
@Override
protected void onStop() {
        if(list.size()>0){
          sm.unregisterListener(sel);
        }
        super.onStop();
     }
  }
```

Output:



```
x :0.0
y :9.77622
z :0.813417
```

Gesture:

In Android phones, gesture serves as hand drawn shape on touch screen having many strokes. In this, every stroke is arrangement of timed points. It is seen that a user explained gesture will locate using Gesture Library. We see that Android will provide particular types of touch screen events which can be pinch, double tap, scrolls , long presses and flinch.

It is seen that Android shows GestureDetector class which will invite motion events and will explain about events with details regarding relevance to particular gestures. In order to apply, you need to form an object of GestureDetector which extends other class using GestureDetector.SimpleOnGestureListener which plays the role of listener by overriding certain applications.

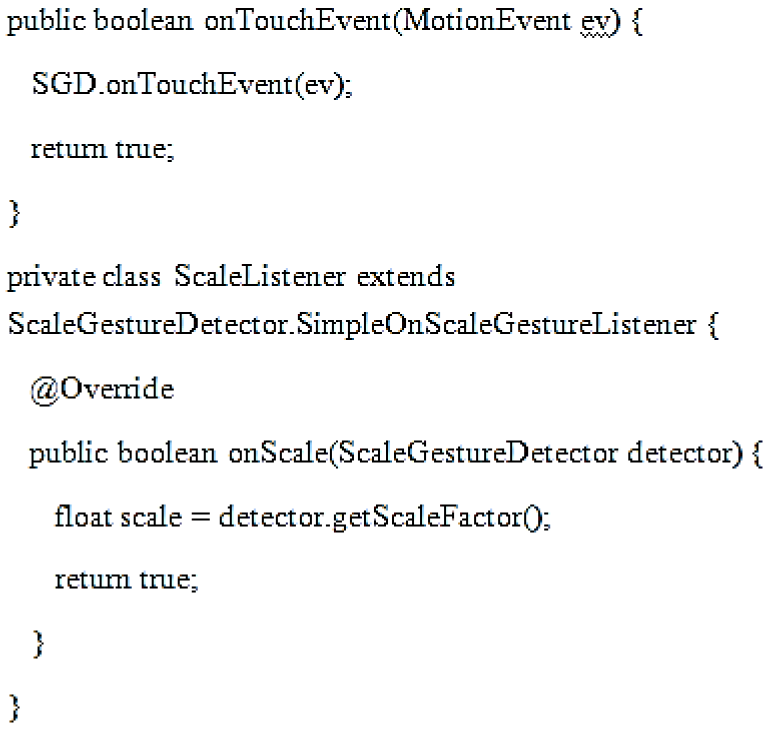
Handling Pinch Gesture

In Android, ScaleGestureDetector class will take care about gestures which is pinch e.t.c. To apply, you have to initiate object of this class. Its syntax is as follow −

ScaleGestureDetector SGD;

SGD = new ScaleGestureDetector(this,new ScaleListener());

We see that initial parameter here is context along with event listener. In this, an event listener will override function OnTouchEvent in order to make it work. Its syntax is given below −



```
public boolean onTouchEvent(MotionEvent ev) {
  SGD.onTouchEvent(ev);
  return true;
}
private class ScaleListener extends
ScaleGestureDetector.SimpleOnScaleGestureListener {
  @Override
  public boolean onScale(ScaleGestureDetector detector) {
    float scale = detector.getScaleFactor();
    return true;
  }
}
```

TalkBack continues the screen reader constructed by Google for Android. Methodically, it’s an accessibility operation that assigns oral feedback. The fraction of the accessibility operation that authorizes blind users to experience the screen without carelessly recharging controls is identified Explore by Touch.

These clear gestures are exercised to guide your Android appliance. Use a touch that is weightless as well as even-handedly rapid. If you’re abstracting, scrolling, or utilizing right-angle or two-part gestures, the touch as well as action are unceasing lighter and faster, as if you were brushing dust or fine particles from a tabletop. Keep in mind that these gestures are available only when TalkBack and

Explore by touch are on:

Explore: Slip one finger over and above the screen in some direction and finds the TalkBack broadcasts text as well as administers as your finger goes over them.

Swipe with one finger: rapidly slip a finger over and above the screen to proceed to the prior item. This is contrasting to pressing shift+tab on a Windows or Apple computer.

Double-tap: Tap the screen double with one finger to revive the conclusive item you listened. By default, this endures how you flap controls, assure as well as uncheck boxes, along with expose dropdown lists. If you're fresh to touch-screens, the double-tap continues a fast arrangement. Analyze of a two-syllable term, like "double" or "Android;" each tap acknowledges to a syllable.

Single-tap: examine to the element further tap it once to revive. This continues an elective method for knocking controls, checking as well as unchecking boxes, additionally releasing dropdown lists. To assign this mode, move into Settings—Accessibility—TalkBack--Settings and furthermore check Single-tap arrangement. You can additionally grab as well as double-tap anywhere when in this arrangement.

Long-press: slip a finger smoothly to an item, tap twice, in addition following the second tap, keep your finger on the screen instead of elevating. This gesture is facilitated to carry up extended options. It additionally opens as well as closes the on-screen keyboard along with it permits you to pull items on the home screens. You can for the time being long-press anywhere, although long-pressing smoothly on an item develops the foremost predictable results.

Scroll vertically: slip double fingers up or down to scroll through a detail. To scroll fruitfully, pass intersection to the list by declining a finger to an element in the list before starting the two-finger fleece. To pass to the screen underling, swipe up, as well as to move to the screen over and above, swipe down. To recall, observe you're experiencing thing on a roll. Short swipes scroll the record by an item or two; many elongated, fast swipes flow through fasten of screens. Be safe to scroll in the work area only. commencing the two-finger swipe together with close to the lowest of the screen may unheedfully launch Google present, as well as introducing the two-finger swipe also close to the peak of the screen may haphazardly open Notifications.

Scroll horizontally: slip two fingers right or left to alter pages as well as screens. Scroll in the function area of the screen, or descend a finger to an element in the menu before scrolling evenly. Scrolling right to left goes to the screen on the right, as well as scrolling left to right flows to the screen on the left. To recall, think you're rotating the pages of a book. In furthermost cases, you can fleece anywhere in the function area, but if your approach is to absolve a notification or discharge an element from recent apps, you expect to examine to the item along with two-finger swipe evenly accurately above it.

Check your progress 4

1. What operations are supported by Android NFC technology?
   a. Reader/writer mode
   b. P2P mode
   c. Card emulation mode
   d. All of these

2. To control gestures in android which class is used?
   a. GestureDetector
   b. Activity class
   c. MapView class
   d. None of these

1.6 Accessibility

Once you've authorized accessibility, you can begin applying your phone. This domain explains many of the foremost common approaches to collaborate with your device furthermore describes several vocabulary. On appliances running Android 4.0 along with higher, you can feel your appliance's screen to hear the content below your finger spoken vocal. To recharge an element, for exemplary to click a button, tap the screen once ensuing examining the item. Or, to charge an feature without examining it first, clearly double-tap the item.

When you examine content that enlarges far the screen furthermore can be scrolled, your device schedule beep with a elevating tone. To scroll content, set two fingers on the screen as well as pass your fingers up or down. You will steer elevating or declining tones as you scroll the detail to allow you recognize your related position within the list. If you pause later scrolling a list, you will steer a recited description of your complete position within the list.

On appliances with a hardware d-pad or arrow keys, you can certainly guide without wishing to utilize the touch screen. This guiding is feasible to entire users furthermore does not stipulate Accessibility to be permitted. Authorizing Accessibility will assign you with oral communication feedback to help in guiding if you cannot glance the screen.

The brand of directional controller differs by phone, however numerous phones possess one of the following: a clickable trackball, arrow keys, or a directional pad. On phones without some hardware directional controls, you can download the Eyes-Free Keyboard from Android Market to obtain an on-screen directional pad. All of these controllers allow you to move in 4 directions, and click to select items on the screen. It is noted that there are keys that are used frequently such as:

Home button key:

This key is applied to take you to parent position, where you can start with an applications, check notifications and can do many things. With home key in Android, you will able to browse certain applications and if required can download apps to use.

Back button key:

This key will allow you to move to previous screen which can be in same app or in another app. If you use this key many times, then you will reach to home screen.

Menu button key:

It is an important key as it will help in operating particular menu present in current screen. They are mostly 2D where arrows are there to move up, down, left and right and can also used to click if required. You have to press the Back key to close a menu.

Check your progress 5

1. Which of the following key is used to get back to the position from where we had started?

   a. Back button key

   b. Home button key

   c. Menu button key

   d. None of these

1.7 Let Us Sum Up

MS-DOS and OS/2 use another variation on linked list called FAT

Index allocation addresses many of the problems of contiguous and chained allocation.

C-Scan Scheduling is a type of scheduling, where the processes get arranged by using particular circular order list.

Round Robin is a type of scheduling where the time of CPU is shared into equal numbers which is called as Quantum Time.

1.8 Answers for Check Your Progress

Check your progress 1

Answers: (1 –b), (2-c)

Check your progress 2

Answers: (1 -a), (2 -b)

Check your progress 3

Answers: (1 –b), (2 –d)

Check your progress 4

Answers: (1 -d), (2 –a)

Check your progress 5

Answers: (1 -b)

1.9 Glossary

1. **Android -** it s a Google's open-source mobile operating system used in smartphones and tablets.
2. **Apps -** It is an application program which can be download and run on smartphone.
3. **Resolution -** It shows different individual pixels in a display.
4. **Application -** It serves as component having one or more activities, services, listeners, and intent receivers.
5. **Drawable -** A compiled visual resource used as a background, title, or other part of the screen.
6. **Intent -** A message object to launch or communicate with other applications/activities asynchronously.

1.10 Assignment

Explain the web API in detail.

1.11 Activities

Create an API which involves all gestures technique.

1.12 Case Study

Study the NFC technology.

1.13 Further Readings

1. Reto Meier, “Professional Android 2 Application Development”, Wiley India Pvt. Ltd. (2011).

UNIT 2: ADVANCED TOPICS - II

Unit Structure

2.0 Learning Objectives

After learning this unit, you will be able to understand:

- About Communication and Identity
- About Social Media
- About Android Native Development Kit

2.1 Introduction

To use Fragment UI components frequently, you need to create each as a completely self-contained, modular component which describes own layout and feature. On defining reusable Fragments, you will linked with them with the help of certain Activities and can connect them using certain application logic in order to realize complete composite UI.

Further it is seen that if you want single Fragment to communicate among other which can be changing content as per user event, then every Fragment-to-Fragment communication will be carried out by certain linked Activities. It is noted that both the Fragments can never communicate directly.

2.2 Communication

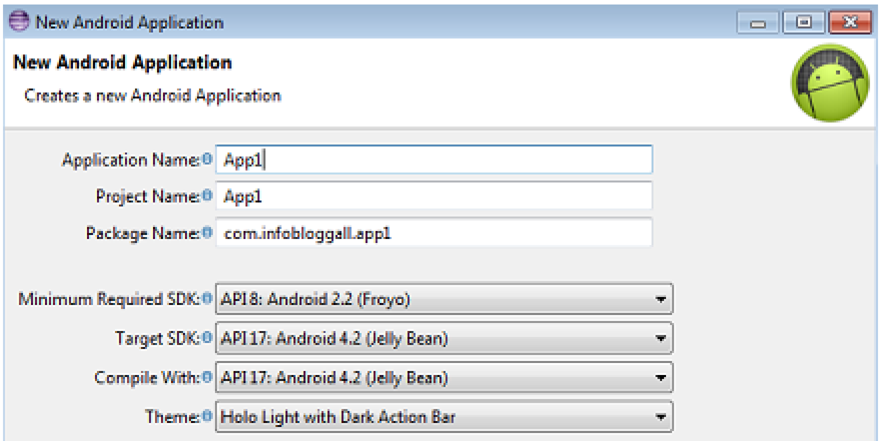
In Android, we see that we can communicate among dual applications with the help of dual process mechanism. It is seen that Messenger class will open a way in order to send messages so that it can be runnable from one process to other process.

Communication among two Applications

Consider two Applications: Application 1 and Application 2

- Creating Application 1 having services that is open to other application.
- Frame a messenger and allow it to work in onBind() method of service.
- Create handler so that it will allow message to be there on Application 2.
- Frame Application 2 by creating ServiceConnection object in order to open messenger.
- Fix exposed service to Application 2 and apply messenger in order to get messages to be send to Application 1.

Step 1: Create Application 1 with name App1.



New Android Application

New Android Application

Creates a new Android Application

Application Name: App1

Project Name: App1

Package Name: com.infobloggall.app1

Minimum Required SDK: API 8: Android 2.2 (Froyo)

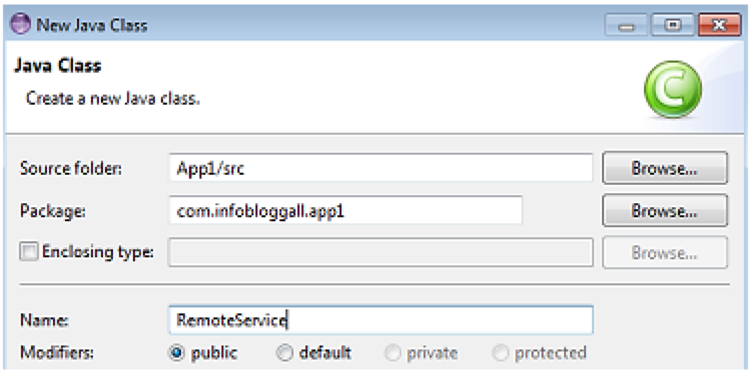
Target SDK: API 17: Android 4.2 (Jelly Bean)

Compile With: API 17: Android 4.2 (Jelly Bean)

Theme: Holo Light with Dark Action Bar

Fig 2.1 New application 1

Step 2: Create class as RemoteService.java which acts as remote service that will open other applications.



Java Class

Create a new Java class.

Source folder: App1/src

Package: com.infobloggall.app1

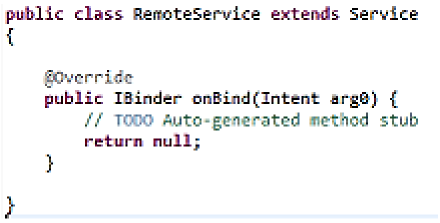
☐ Enclosing type:

Name: RemoteService

Modifiers: ● public ○ default ○ private ○ protected

Fig 2.2 New Java Class

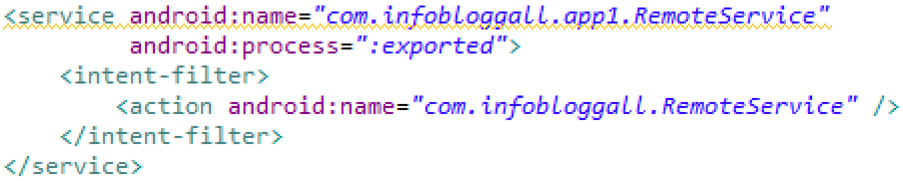
Step 3: Now we have to extend service so that our class will have features of service and also implement the onBind() method.



```
public class RemoteService extends Service
{

    @Override
    public IBinder onBind(Intent arg0) {
        // TODO Auto-generated method stub
        return null;
    }

}
```

Fig 2.3 Features

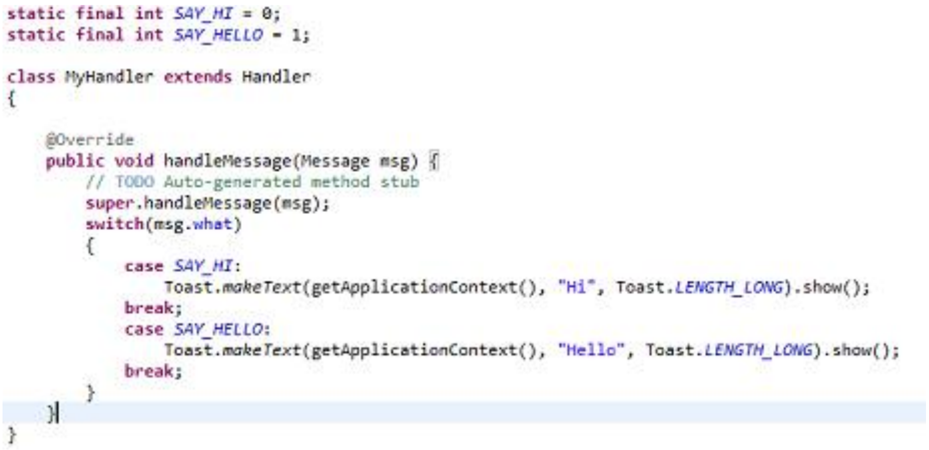
Step 4: After now, to make such service remote, the RemteService to other Applications or Process will exists. In Manifest file, name the service and make it remote so that application can use it. For this, export field that will allow many process to utilize service and explain intent filter along with action name in order to call the service from various applications.



```
<service android:name="com.infobloggall.app1.RemoteService"
         android:process=":exported">
    <intent-filter>
        <action android:name="com.infobloggall.RemoteService" />
    </intent-filter>
</service>
```

Fig 2.4 Code for remote service

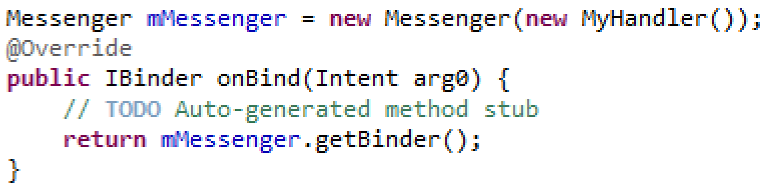
Step 5: Move to RemoteService.java class and create handler to take care of messages sent across from different other application that will be displayed in a toast. Also frame constants with regards to message in order to find the sent messages.



```
static final int SAY_HI = 0;
static final int SAY_HELLO = 1;

class MyHandler extends Handler
{

    @Override
    public void handleMessage(Message msg) {
        // TODO Auto-generated method stub
        super.handleMessage(msg);
        switch(msg.what)
        {
            case SAY_HI:
                Toast.makeText(getApplicationContext(), "Hi", Toast.LENGTH_LONG).show();
            break;
            case SAY_HELLO:
                Toast.makeText(getApplicationContext(), "Hello", Toast.LENGTH_LONG).show();
            break;
        }
    }
}
```

Fig 2.5 RemoteService.java class

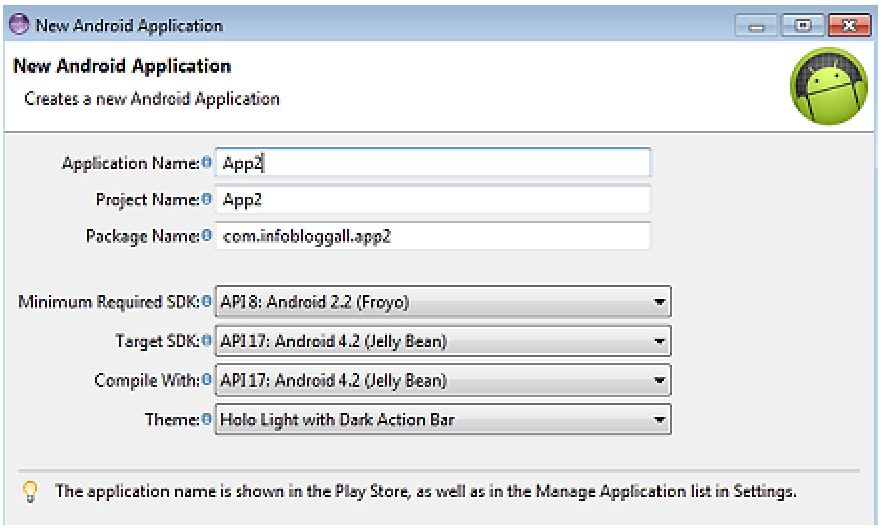
Step 6: Create the Messenger by passing the handler to messenger’s constructor and attach the handler with messenger. Now in onBind() method, open the messenger so that service can send messages using open messenger.



```
Messenger mMessenger = new Messenger(new MyHandler());
@Override
public IBinder onBind(Intent arg0) {
    // TODO Auto-generated method stub
    return mMessenger.getBinder();
}
```

Fig 2.6 Code for messenger

Step 7: From this, we have completed with Application 1 and now start creation of Application 2.



New Android Application

New Android Application

Creates a new Android Application

Application Name: App2

Project Name: App2

Package Name: com.infobloggall.app2

Minimum Required SDK: API 8: Android 2.2 (Froyo)

Target SDK: API 17: Android 4.2 (Jelly Bean)

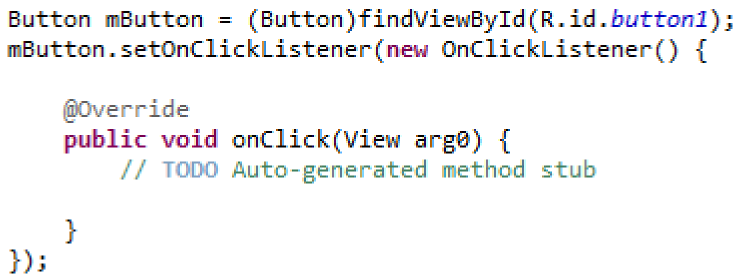
Compile With: API 17: Android 4.2 (Jelly Bean)

Theme: Holo Light with Dark Action Bar

The application name is shown in the Play Store, as well as in the Manage Application list in Settings.

Fig 2.7 New Android application

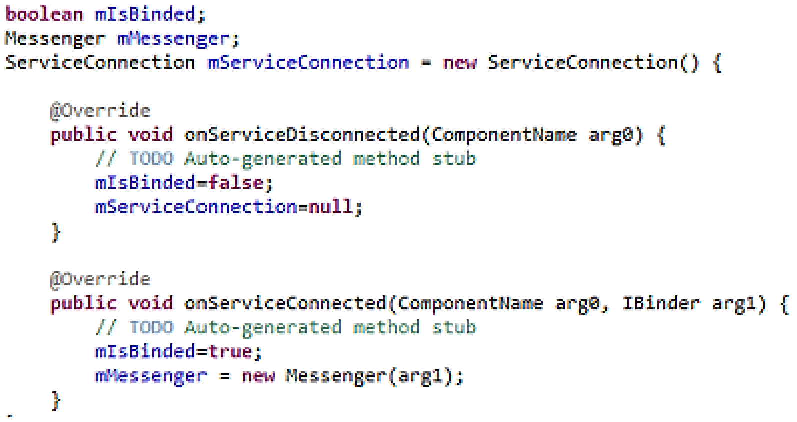
Step 8: Here we will add a button to activity which on clicking, will send the message to Application 1.



```
Button mButton = (Button)findViewById(R.id.button1);
mButton.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View arg0) {
        // TODO Auto-generated method stub

    }
});
```

Fig 2.8 Code for adding button

Step 9: No frame ServiceConnection object that fetch Messenger exposed by RemoteService. The arg1 variable is used to retrieve Messanger.



```
boolean mIsBinded;
Messenger mMessenger;
ServiceConnection mServiceConnection = new ServiceConnection() {

    @Override
    public void onServiceDisconnected(ComponentName arg0) {
        // TODO Auto-generated method stub
        mIsBinded=false;
        mServiceConnection=null;
    }

    @Override
    public void onServiceConnected(ComponentName arg0, IBinder arg1) {
        // TODO Auto-generated method stub
        mIsBinded=true;
        mMessenger = new Messenger(arg1);
    }
```

Fig 2.9 Creating service connection

We have taken the messenger out and apply messenger to send message to App1 by clicking the button. Also, we make similar constants as what we did in App 1. On click of button, send() method will be called and send message to messenger of App 1.

Check your progress 1

1. Which class is used to communicate among dual applications?

   a. Gesture Detector

   b. Messenger class

   c. Map View class

   d. None of these

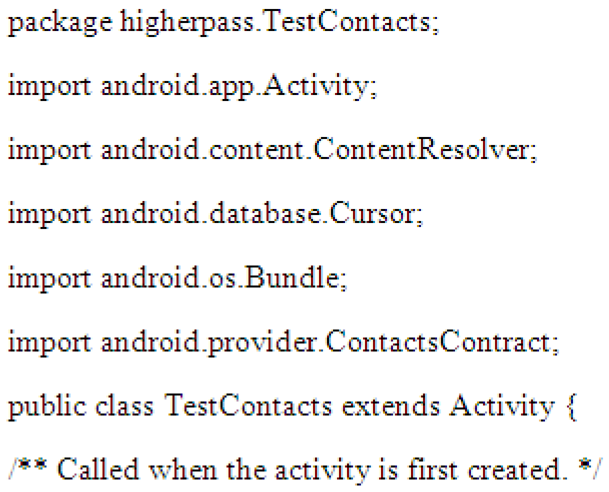
2.3 Identity, Sync and Social Media

Before an application can query the contact records access must be granted through the AndroidManifest.xml file stored in the root of the project. Add the following uses-permission bellows the uses-sdk statement.

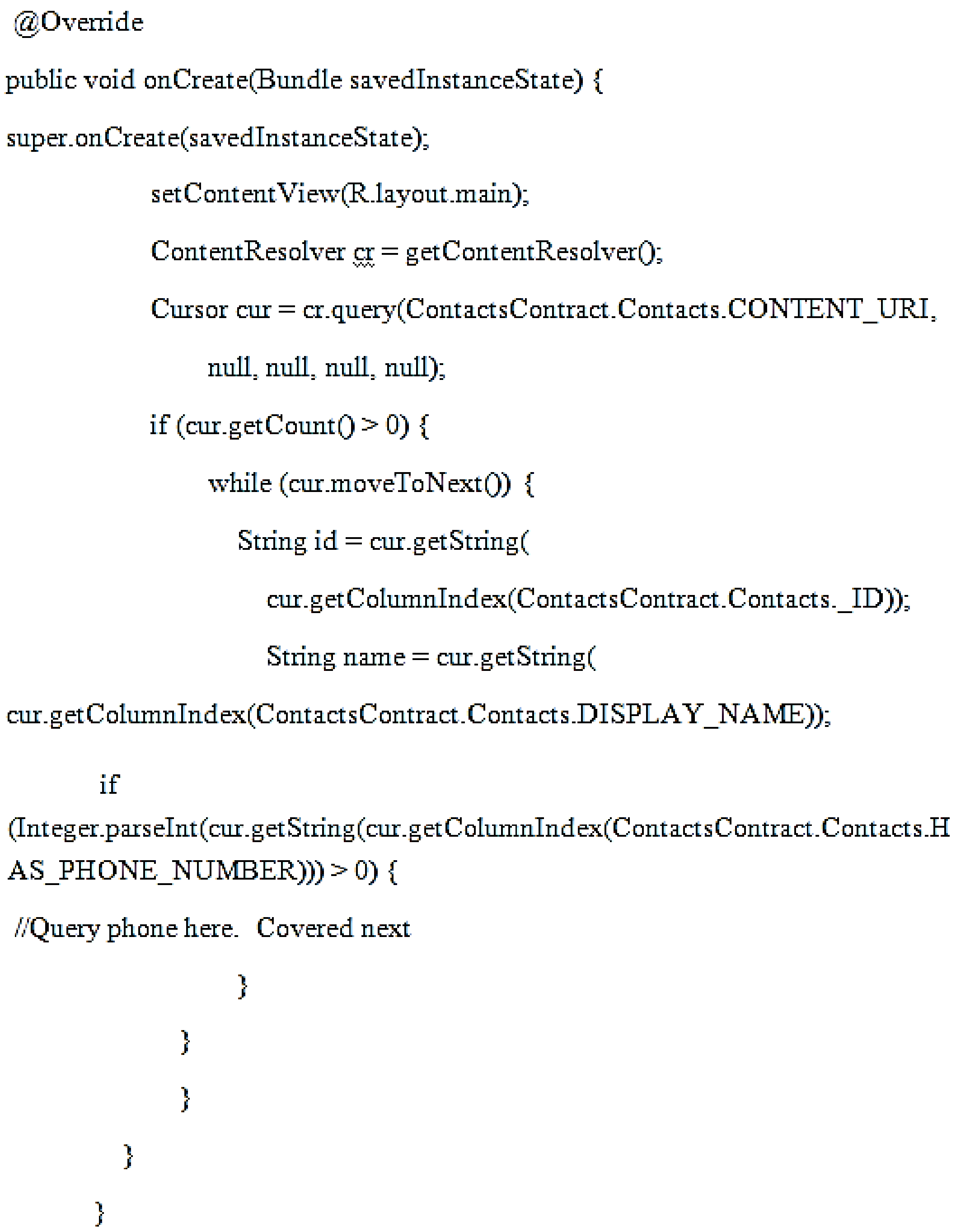
Querying The Android Contact Database

Retrieving Contact Details

Basic contact information stored in Contacts table with detailed information stored in individual tables for normalization. In Android 2.0 to query the base contact records the URI to query is stored in ContactsContract.Contacts.CONTENT_URI.



```
package higherpass.TestContacts;

import android.app.Activity;

import android.content.ContentResolver;

import android.database.Cursor;

import android.os.Bundle;

import android.provider.ContactsContract;

public class TestContacts extends Activity {

/** Called when the activity is first created. */
```



```
 @Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
                setContentView(R.layout.main);
                ContentResolver cr = getContentResolver();
                Cursor cur = cr.query(ContactsContract.Contacts.CONTENT_URI,
                      null, null, null, null);
                if (cur.getCount() > 0) {
                      while (cur.moveToNext()) {
                         String id = cur.getString(
                            cur.getColumnIndex(ContactsContract.Contacts._ID));
                            String name = cur.getString(
cur.getColumnIndex(ContactsContract.Contacts.DISPLAY_NAME));
          if
(Integer.parseInt(cur.getString(cur.getColumnIndex(ContactsContract.Contacts.H
AS_PHONE_NUMBER))) > 0) {
 //Query phone here.  Covered next
                         }
                   }
                   }
             }
          }
```

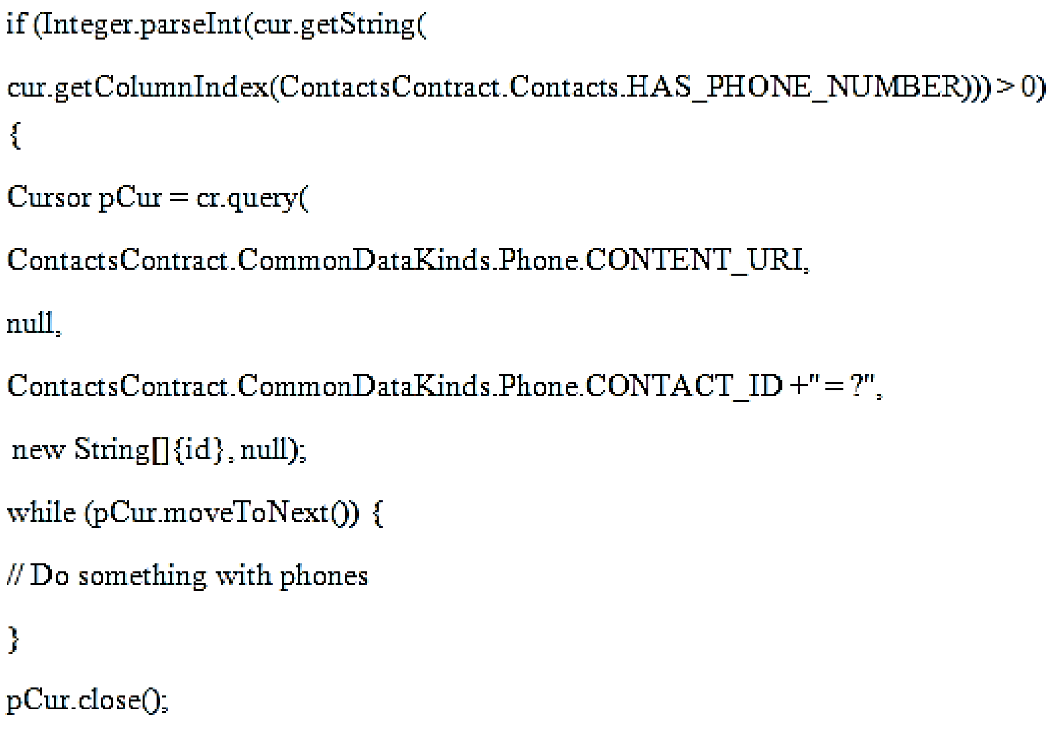
We see that this application will work as any other Android application where initially we have to create ContentResolver instance in cr and further ContentResolver instance so as to query database which return Cursor with contacts list. Here we see that query has done against URI which is inside ContactsContract.Contacts.CONTENT_URI. Further it will see whether the cursor has records and if so, then it will loop them.

Phone Numbers

We see that phone numbers are kept inside table and queried separately using URI stored in SDK variable

ContactsContract.CommonDataKinds.Phone.CONTENT_URI.

Apply WHERE conditional to get phone numbers for particular contact.



```
if (Integer.parseInt(cur.getString(
cur.getColumnIndex(ContactsContract.Contacts.HAS_PHONE_NUMBER))) > 0)
{
Cursor pCur = cr.query(
ContactsContract.CommonDataKinds.Phone.CONTENT_URI,
null,
ContactsContract.CommonDataKinds.Phone.CONTACT_ID +" = ?",
 new String[]{id}, null);
while (pCur.moveToNext()) {
// Do something with phones
}
pCur.close();
```

}

No we will do second query against Android contacts SQLite database where phone numbers are queried against URI which is kept in

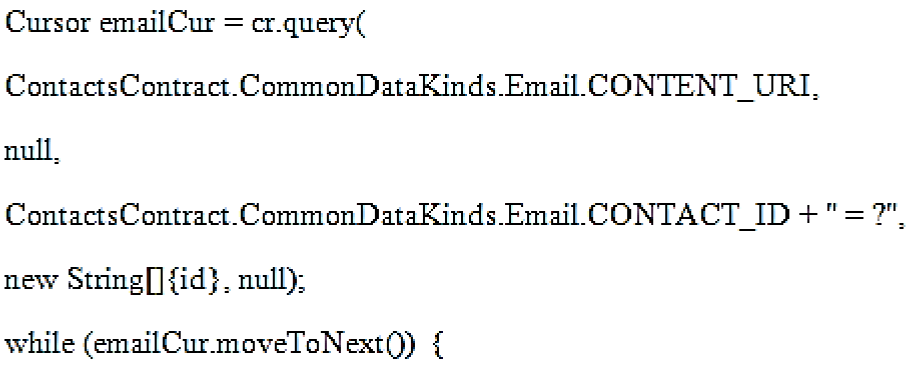
ContactsContract.CommonDataKinds.Phone.CONTENT_URI.

The contact ID is stored in the phone table as

ContactsContract.CommonDataKinds.Phone.CONTACT_ID and the WHERE clause is used to limit the data returned.

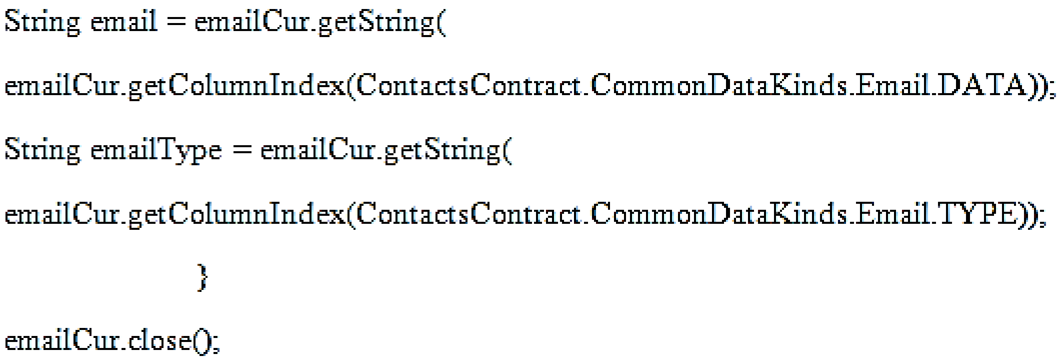
Email Addresses

It is noted that querying email address is same as phone numbers which can be done in order to have email addresses from database. Query URI stored in ContactsContract.CommonDataKinds.Email.CONTENT_URI will query email address table as:



```
Cursor emailCur = cr.query(
ContactsContract.CommonDataKinds.Email.CONTENT_URI,
null,
ContactsContract.CommonDataKinds.Email.CONTACT_ID + " = ?",
new String[]{id}, null);
while (emailCur.moveToNext()) {
```

```
// This would allow you get several email addresses
// if the email addresses were stored in an array
```



```
String email = emailCur.getString(
emailCur.getColumnIndex(ContactsContract.CommonDataKinds.Email.DATA));
String emailType = emailCur.getString(
emailCur.getColumnIndex(ContactsContract.CommonDataKinds.Email.TYPE));
                    }
emailCur.close();
```

Along with phone query, field names related to email table gets stored under ContactsContract.CommonDataKinds. It is seen that email query is done on URI in ContactsContract.CommonDataKinds.Email.CONTENT_URI and WHERE clause will match ContactsContract.CommonDataKinds.Email.CONTACT_ID field. As numerous email addresses can be stored, it is found that loop through records will return in Cursor.

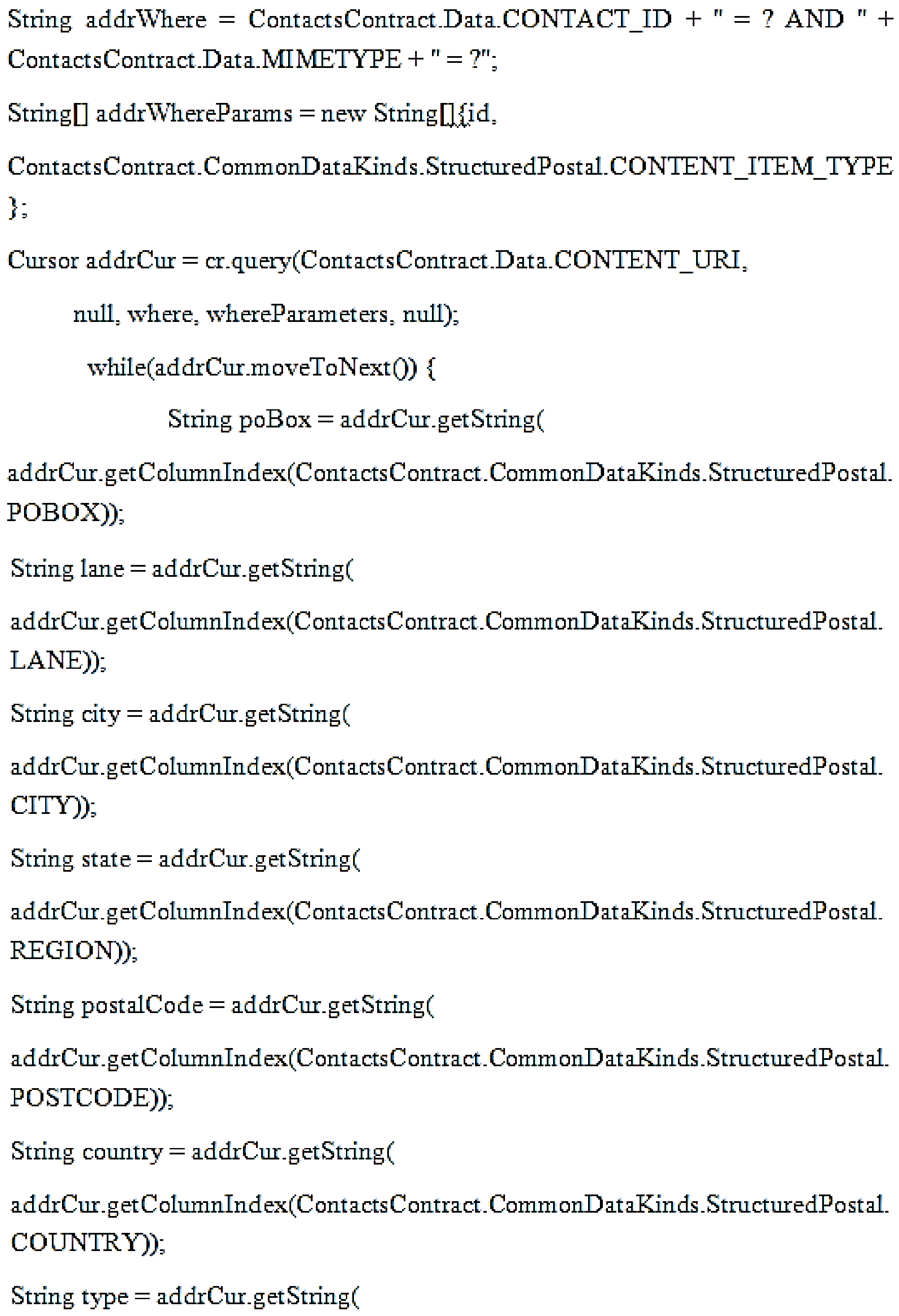
Postal Addresses

It is found that Android stores many postal addresses including their contact details that are kept in data table which can be notes and gets queried through URI stored in ContactsContract.Data.CONTENT_URI. It is noted that same as notes query, MIMETYPE gets added to WHERE conditional. It is found that in Android 2.0, Address record gets splitted into various fields having many portions of address which can be:

- PO-Box
- Lane
- City
- Region

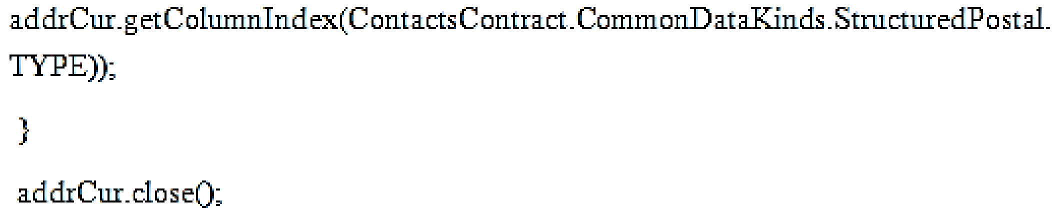
- Postal code

In earlier versions of the Android SDK this was free form string storage.



```
String addrWhere = ContactsContract.Data.CONTACT_ID + " = ? AND " +
ContactsContract.Data.MIMETYPE + " = ?";

String[] addrWhereParams = new String[]{id,

ContactsContract.CommonDataKinds.StructuredPostal.CONTENT_ITEM_TYPE
};

Cursor addrCur = cr.query(ContactsContract.Data.CONTENT_URI,

        null, where, whereParameters, null);

          while(addrCur.moveToNext()) {

                      String poBox = addrCur.getString(

addrCur.getColumnIndex(ContactsContract.CommonDataKinds.StructuredPostal.
POBOX));

 String lane = addrCur.getString(

 addrCur.getColumnIndex(ContactsContract.CommonDataKinds.StructuredPostal.
 LANE));

 String city = addrCur.getString(

 addrCur.getColumnIndex(ContactsContract.CommonDataKinds.StructuredPostal.
 CITY));

 String state = addrCur.getString(

 addrCur.getColumnIndex(ContactsContract.CommonDataKinds.StructuredPostal.
 REGION));

 String postalCode = addrCur.getString(

 addrCur.getColumnIndex(ContactsContract.CommonDataKinds.StructuredPostal.
 POSTCODE));

 String country = addrCur.getString(

 addrCur.getColumnIndex(ContactsContract.CommonDataKinds.StructuredPostal.
 COUNTRY));

 String type = addrCur.getString(
```

(

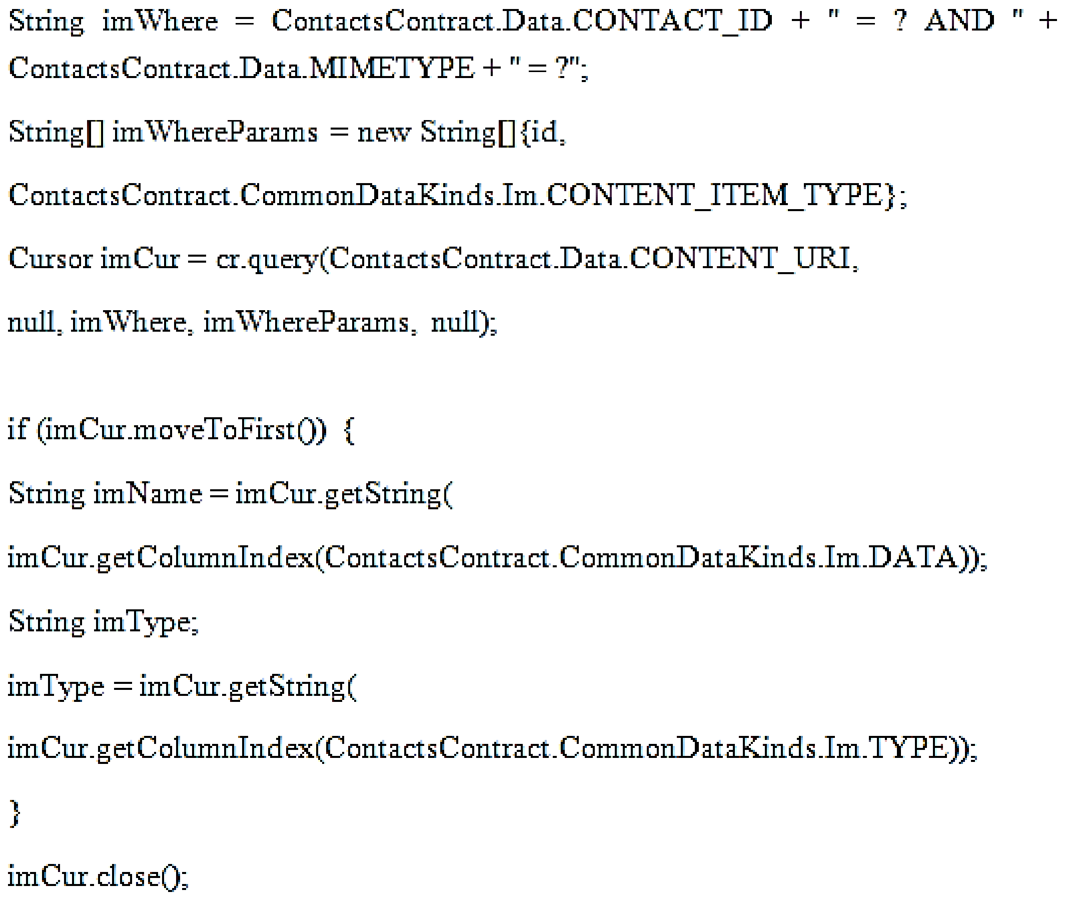


```
addrCur.getColumnIndex(ContactsContract.CommonDataKinds.StructuredPostal.
TYPE));
 }
 addrCur.close();
```

We see that the above program code is same as previous shown where field names for address are kept in ContactsContract.CommonDataKinds.StructuredPostal

Instant Messenger (IM)

The instant messenger query performs just as the notes and address queries. Important field names for IM related data are stored in ContactsContract.CommonDataKinds.Im.

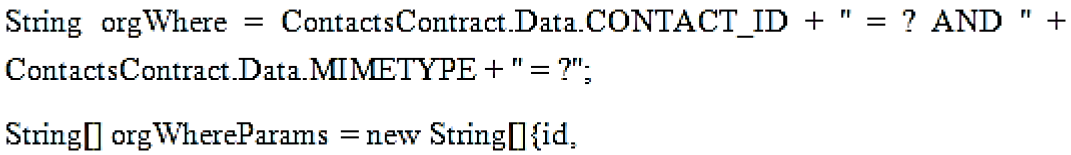


```
String imWhere = ContactsContract.Data.CONTACT_ID + " = ? AND " +
ContactsContract.Data.MIMETYPE + " = ?";
String[] imWhereParams = new String[]{id,
ContactsContract.CommonDataKinds.Im.CONTENT_ITEM_TYPE};
Cursor imCur = cr.query(ContactsContract.Data.CONTENT_URI,
null, imWhere, imWhereParams, null);

if (imCur.moveToFirst()) {
String imName = imCur.getString(
imCur.getColumnIndex(ContactsContract.CommonDataKinds.Im.DATA));
String imType;
imType = imCur.getString(
imCur.getColumnIndex(ContactsContract.CommonDataKinds.Im.TYPE));
}
imCur.close();
```

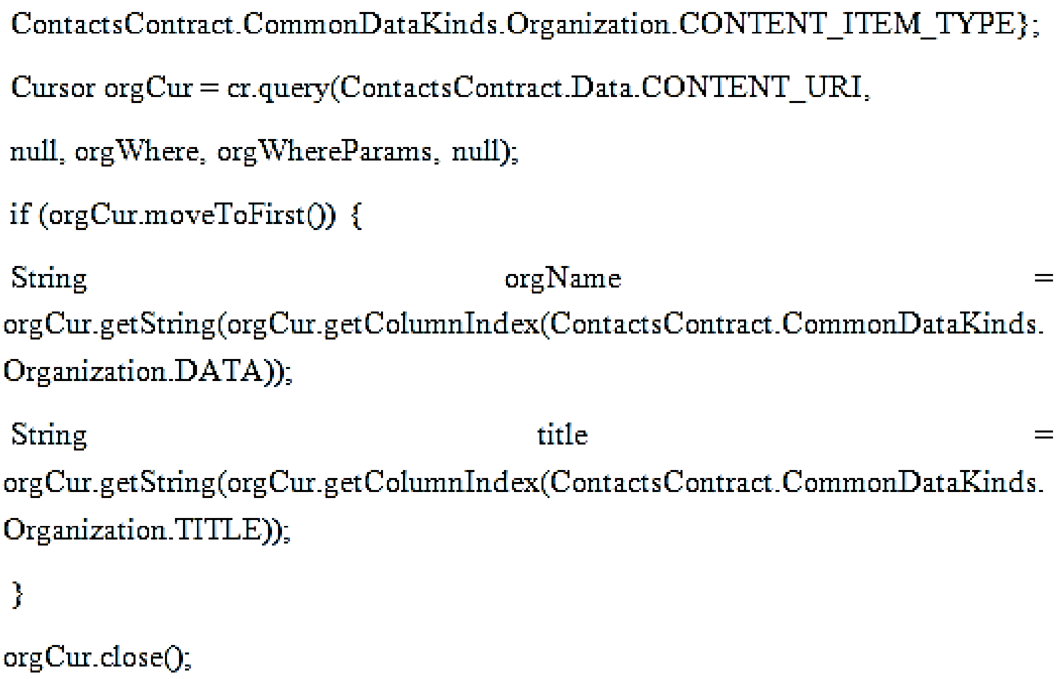
Organizations

In Android, we see that organization data is also part of contact record which carries detail about Employment, professional and memberships. Such information gets queried from URI stored in ContactsContract.Data.CONTENT_URI.

In this an important field names for organization data gets stored in ContactsContract.CommonDataKinds.Organization.



```
String orgWhere = ContactsContract.Data.CONTACT_ID + " = ? AND " + ContactsContract.Data.MIMETYPE + " = ?";
String[] orgWhereParams = new String[]{id,
```



```
ContactsContract.CommonDataKinds.Organization.CONTENT_ITEM_TYPE};
Cursor orgCur = cr.query(ContactsContract.Data.CONTENT_URI,
null, orgWhere, orgWhereParams, null);
if (orgCur.moveToFirst()) {
String orgName = orgCur.getString(orgCur.getColumnIndex(ContactsContract.CommonDataKinds.Organization.DATA));
String title = orgCur.getString(orgCur.getColumnIndex(ContactsContract.CommonDataKinds.Organization.TITLE));
}
orgCur.close();
```

Check your progress 2

1. Instant Messenger data can be queried using_______query.

   a. ContactsContract.Data.CONTENT_URI.

   b. ContactsContract.CommonDataKinds.Email.CONTENT_URI

   c. ContactsContract.CommonDataKinds.Im.

   d. None of these

2. Email addresses can be retrieved by using query __________.

   a. ContactsContract.Contacts.CONTENT_URI

   b. ContactsContract.CommonDataKinds.Email.CONTENT_URI

   c. ContactsContract.CommonDataKinds.Im.

   d. None of these

2.4 The Android Native Development Kit (NDK)

There are some ways to make apps for android devices, ranging from very high-level cross-platform engines to writing code native to specific processors. The Native Development Toolkit (NDK) allows using C and C++ in android apps, however why you'd need to and how to do so could be a mystery to several developers. Android apps run within the Dalvik virtual machine, which interprets device-agnostic, cross-platform commands into instructions for the specific device that it's running on. Code written in Java (or in other high-level languages that are converted into Java) doesn't need to handle machine-specific details; the virtual machine manages details for the processor, graphics, and even memory. You do not have to be compelled to handle any communication to processes or modules outside of the virtual machine.

In most cases, the speed and memory overhead could be a worthwhile tradeoff. In some cases, developers want absolutely the quickest performance possible. The NDK permits embedding C and C++ elements among android apps, permitting the foremost performance-intensive items to be as near the hardware as attainable. This comes at a value, although — using native code complicates development. There are additional tools to use and infrastructure to set up. Also, some details that were handled by the Dalvik virtual machine should now be handled by the developer. For these reasons, native code should be used only when necessary.

When native code is needed

Many times the selection of high-level language and tools depends on the developer's skill set and therefore the existing assets; but, the android team admonishes that familiarity with C or C++ isn't a sufficient reason to use the NDK.

There are times that using native code may be advantageous, like processing data or computing physics and graphics for games. Access to existing native libraries, further as high-performance code, may also be smart reasons.

The magnified quality from using native code ought to be balanced with a suitable performance increase. Native code is not inherently faster — the developer is responsible for optimizing the native code and understanding that items of every app should be native.

Uses for the native code

The android team makes it clear that most developers should not need to use the NDK. In other words, if you do not understand what native code is, and if you do not have a clear reason to use it, stay away from it.

Game engine developers usually dive right in to native code. The limited speed and memory of mobile devices means that native code is also necessary to squeeze as of potential out for them. There are variety of established high-performance game engines based on C++ that are used to develop cross-platform games; this does not mean that game development requires working in C++.

Another notable use of the NDK is once using a platform that cross-compiles into native code. Whereas the platform builders would want to understand the NDK very well, the developer wouldn't. As an example, some game engines enable developers to use Lua or similar high-level scripting languages to create the game, and handle the interaction with the NDK internally.

While you would possibly assume that interfacing with external hardware would need using native code, the accessory Development Kit handles that. One final example of once to use native code is computationally expensive math, like audio, video, or image process. This is often not going to be a locality for developers to be casually concerned in and is most frequently utilized in specialised applications.

Installing Native Development Kit (NDK)

The NDK provides all the tools (compilers, libraries, and header files) to make apps that access the device natively. Native code (in C/C++) is important for high performance to overcome the limitations in Java's memory management and performance.

To install Android NDK:

Set necessary tools for Android programming like JDK, Eclipse, Android SDK, Eclipse ADT and Cygwin.

Download Android NDK from http://developer.android.com/tools/sdk/ndk/index.html

Unzip downloaded file in directory. The NDK will be unzipped as d:\myproject\android-ndk-r8 which is a home directory.

Include the NDK installed directory in the PATH environment variable.

Writing an application:

The NDK documentation is kept in the "docs" sub-directory. The steps in building an android NDK app are:

- Create a sub-directory referred to as "jni" and place all the native sources here.
- Create an "Android.mk" to explain your native sources to the NDK build system.
- Build your native code by running the "ndk-build" (in NDK put in directory) script from your project's directory. The build tools copy the stripped, shared libraries required by your application to the right location within the application's project directory.
- Finally, compile and run your application using the SDK tools within the usual way. The SDK build tools can package the shared libraries within the application's deployable ".apk" file.

Check your progress 3

1. The main advantage of native code is_______________.
   a. computing physics and
   b. Graphics for games.
   c. high-performance of code in data processing
   d. All of these

2.5 Let Us Sum Up

While studying this unit, we have learnt that for using Fragment UI components frequently, you need to create each as a completely self-contained, modular component which describes own layout and feature. In Android, we see that we can communicate among dual applications with the help of dual process mechanism. The Messenger class will open way in order to send messages so that it can be runnable from one process to other process.

It is noted that prior to an application, a query contains contact records whose access must be granted by AndroidManifest.xml file stored in the root of the project. In Android, we see that organization data is also part of contact record which carries detail about Employment, professional and memberships. Native Development Toolkit allows use of C and C++ in android apps for running in Dalvik virtual machine that interprets device-agnostic, cross-platform commands into instructions for the specific device that it's running on.

2.6 Answers for Check Your Progress

Check your progress 1

Answers: (1 –b)

Check your progress 2

Answers: (1 -c), (2-b)

Check your progress 3

Answers: (1 –d)

2.7 Glossary

1. **Resolution -** It shows different individual pixels in a display.
2. **Application -** It serves as component having one or more activities, services, listeners, and intent receivers.
3. **Resources -** Nonprogrammatic application components apart from compiled application code that loaded from application code by well-known reference format.

2.8 Assignment

Write short note on Native development kit.

2.9 Activities

Create an API that will retrieve user’s phone number, email id and postal address.

2.10 Case Study

Study communication between Android applications.

2.11 Further Readings

1. Reto Meier, “Professional Android 2 Application Development”, Wiley India Pvt. Ltd. (2011).
2. Mark L Murphy, “Beginning Android”, Wiley India Pvt. Ltd. (2009).
3. Sayed Y Hashimi and Satya Komatineni, “Pro Android”, Wiley India Pvt. Ltd. (2009).
4. Lauren Darcey and Shane Conder, “Android Wireless Application Development”, Pearson Education, 2nd ed. (2011

Block Summary

In this block, you have learnt and understand about the basic working features of android application related to location, maps and contacts with knowledge on its running capabilities. The block gives an idea on the study and concept of Native Development Toolkit using C and C++ in android apps. You have been well explained on the concepts of usability and application of working with Gestures in Gesture Library.

The block detailed about the basic of Fragment UI component techniques. The concept related to input output supervisors and input output drivers will also be explained to you. You will be demonstrated practically about Web Telephony Integration technique.

Block Assignment

Short Answer Questions

1. What is Android networking?
2. How Google maps can be embedded in Android API?
3. Explain the role of sensors in android
4. What is NFC? How it can be used in android?

Long Answer Questions

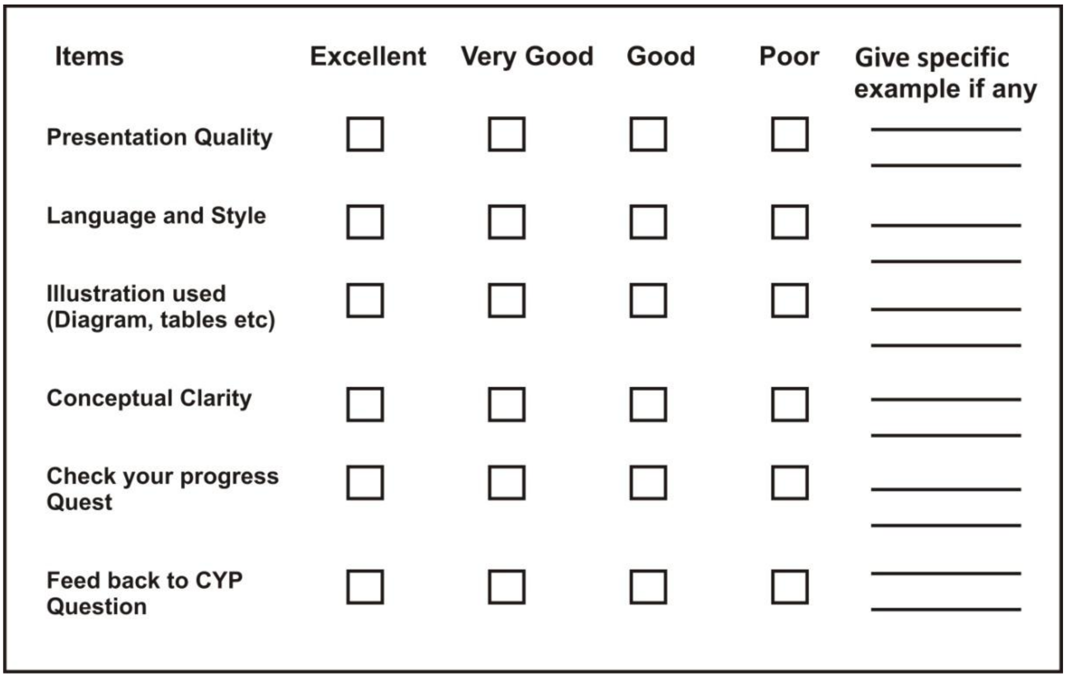
1. Explain Android native development kit (NDK).
2. Explain how two applications can communicate with each other.
3. What are the difference between web and telephony API?
4. Explain various gestures in android.

Enrolment No. ________

1. How many hours did you need for studying the units?

Unit No	**1**	**2**	**3**	**4**
Nos of Hrs				

2. Please give your reactions to the following items based on your reading of the block:



Items	Excellent	Very Good	Good	Poor	Give specific example if any
Presentation Quality	□	□	□	□	________ ________
Language and Style	□	□	□	□	________ ________
Illustration used (Diagram, tables etc)	□	□	□	□	________ ________
Conceptual Clarity	□	□	□	□	________ ________
Check your progress Quest	□	□	□	□	________ ________
Feed back to CYP Question	□	□	□	□	________ ________

3. Any Other Comments

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

> *Education is something which ought to be brought within the reach of every one.*

- Dr. B. R. Ambedkar

Dr. Babasaheb Ambedkar Open University
Jyotirmay' Parisar, Opp. Shri Balaji Temple, Sarkhej-Gandhinagar Highway, Chharodi,
Ahmedabad-382 481.

MOBILE APPLICATION DEVELOPMENT

PGDCA 203

BLOCK 3:
ANDROID APPLICATION PUBLISHING AND CONTENT PROVIDERS

स्वाध्यायः परमं तपः

Dr. Babasaheb Ambedkar Open University
Ahmedabad

MOBILE APPLICATION DEVELOPMENT

KNOWLEDGE MANAGEMENT & RESEARCH ORGANIZATION

Knowledge Management and
Research Organization
Pune

Editorial Panel

Author
Er. Nishit Mathur

Language Editor
Prof. Jaipal Gaikwad

Graphic and Creative Panel
Ms. K. Jamdal
Ms. Lata Dawange
Ms. Pinaz Driver
Ms. Ashwini Wankhede
Mr. Kiran Shinde
Mr. Prashant Tikone
Mr. Akshay Mirajkar

Copyright © 2015 Knowledge Management and Research Organization.
All rights reserved. No part of this book may be reproduced, transmitted or utilized in any form or by a means, electronic or mechanical, including photocopying, recording or by any information storage or retrieval system without written permission from us.

Acknowledgment
Every attempt has been made to trace the copyright holders of material reproduced in this book. Should an infringement have occurred, we apologize for the same and will be pleased to make necessary correction/amendment in future edition of this book.
The content is developed by taking reference of online and print publications that are mentioned in Bibliography. The content developed represents the breadth of research excellence in this multidisciplinary academic field. Some of the information, illustrations and examples are taken "as is" and as available in the references mentioned in Bibliography for academic purpose and better understanding by learner.'

ROLE OF SELF INSTRUCTIONAL MATERIAL IN DISTANCE LEARNING

The need to plan effective instruction is imperative for a successful distance teaching repertoire. This is due to the fact that the instructional designer, the tutor, the author (s) and the student are often separated by distance and may never meet in person. This is an increasingly common scenario in distance education instruction. As much as possible, teaching by distance should stimulate the student's intellectual involvement and contain all the necessary learning instructional activities that are capable of guiding the student through the course objectives. Therefore, the course / self-instructional material are completely equipped with everything that the syllabus prescribes.

To ensure effective instruction, a number of instructional design ideas are used and these help students to acquire knowledge, intellectual skills, motor skills and necessary attitudinal changes. In this respect, students' assessment and course evaluation are incorporated in the text.

The nature of instructional activities used in distance education self-instructional materials depends on the domain of learning that they reinforce in the text, that is, the cognitive, psychomotor and affective. These are further interpreted in the acquisition of knowledge, intellectual skills and motor skills. Students may be encouraged to gain, apply and communicate (orally or in writing) the knowledge acquired. Intellectual-skills objectives may be met by designing instructions that make use of students' prior knowledge and experiences in the discourse as the foundation on which newly acquired knowledge is built.

The provision of exercises in the form of assignments, projects and tutorial feedback is necessary. Instructional activities that teach motor skills need to be graphically demonstrated and the correct practices provided during tutorials. Instructional activities for inculcating change in attitude and behavior should create interest and demonstrate need and benefits gained by adopting the required change. Information on the adoption and procedures for practice of new attitudes may then be introduced.

Teaching and learning at a distance eliminates interactive communication cues, such as pauses, intonation and gestures, associated with the face-to-face method of teaching. This is particularly so with the exclusive use of print media. Instructional activities built into the instructional repertoire provide this missing interaction between the student and the teacher. Therefore, the use of instructional activities to affect better distance teaching is not optional, but mandatory.

Our team of successful writers and authors has tried to reduce this.

Divide and to bring this Self Instructional Material as the best teaching and communication tool. Instructional activities are varied in order to assess the different facets of the domains of learning.

Distance education teaching repertoire involves extensive use of self-instructional materials, be they print or otherwise. These materials are designed to achieve certain pre-determined learning outcomes, namely goals and objectives that are contained in an instructional plan. Since the teaching process is affected over a distance, there is need to ensure that students actively participate in their learning by performing specific tasks that help them to understand the relevant concepts. Therefore, a set of exercises is built into the teaching repertoire in order to link what students and tutors do in the framework of the course outline. These could be in the form of students' assignments, a research project or a science practical exercise. Examples of instructional activities in distance education are too numerous to list. Instructional activities, when used in this context, help to motivate students, guide and measure students' performance (continuous assessment)

PREFACE

We have put in lots of hard work to make this book as user-friendly as possible, but we have not sacrificed quality. Experts were involved in preparing the materials. However, concepts are explained in easy language for you. We have included may tables and examples for easy understanding.

We sincerely hope this book will help you in every way you expect.

All the best for your studies from our team!

Mobile Application Development

Contents

BLOCK 3: ANDROID NETWORKING AND DEVELOPMENT

UNIT 1 **ADVANCED TOPICS - I**

Android Networking, Web and Telephony API, Search, Location and Mapping, Sensors, NFC, Speech, Gestures, and Accessibility,

UNIT 2 **ADVANCED TOPICS - II**

Communication, Identity, Sync, and Social Media, The Android Native Development Kit (NDK)

BLOCK 4: ANDROID APPLICATION PUBLISHING AND CONTENT PROVIDERS

UNIT 1 **MORE ON ANDROID**

Handling and Persisting Data, A Content Provider as a Facade for a RESTful Web Service, Using Content Providers

UNIT 2 **PUBLISHING ANDROID APPLICATION**

Deploying Android Application to the World, Selling your Android application

Dr. Babasaheb
Ambedkar
Open University

PGDCA-203

MOBILE APPLICATION DEVELOPMENT

BLOCK 4: ANDROID APPLICATION PUBLISHING AND CONTENT PROVIDERS

BLOCK 4: ANDROID APPLICATION PUBLISHING AND CONTENT PROVIDERS

Block Introduction

Android application publishing is a process that makes your Android applications available to users. In order to make your application to work across world, you need to release it through application marketplace such as Google Play. Android provides many options to save persistent application data which is as per needs. It depends on type of data which can be private for an application or visible or usable to other applications along with storage space. It is found that many Android applications work with wide variety of data types. It can be done using handling and persisting of data inside an application.

In this block, we will detail about options of saving persistent application data in Android as required. The block will focus on the study and concept of release of Android application across world after performing release test and process. You will get an idea on various platforms required in order to market Android application.

In this block, you will make to learn and understand about the basic of Content Provider function with its data description features. The concept related to content provider working with role in Android application development will be explained to you. You will be demonstrated practically about marketing models applied in case of Android application.

Block Objective

After learning this block, you will be able to understand:

- About Handling of Data
- The Features of Facade
- Idea about Content Providers
- Characteristics of deploying Android Application
- Selling of Android application

Block Structure

Unit 1: **More on Android**

Unit 2: **Publishing Android Application**

UNIT 1: MORE ON ANDROID

Unit Structure

1.0 Learning Objectives

After learning this unit, you will be able to understand:

- About Persisting Data
- About Content Provider
- About Content Providers

1.1 Introduction

Android provides many options to save persistent application data which is as per needs. It depends on type of data which can be private for an application or visible or usable to other applications along with storage space.

Android provides a way for you to open private data to other applications with the help of content provider. It is seen that a content provider serves as optional component which gives read/write access to application data depends on the type of restrictions that to be imposed.

1.2 Handling and Persisting Data

It is found that many Android applications work with wide variety of data types. It can be done using handling and persisting of data inside an application. To have all features offered by modern mobile devices such as tracking contacts, handling events and tasks, mobile operating system and its applications in such case be capable of storing large volume of data with monitoring of track with large data volume. The data is usually structured in rows and columns, similar to a spreadsheet or a standard database.

To do numerous activities given by modern mobile phones like tracking contacts, events and tasks mobile operating system along with its applications should follow process in order to store and keep track of heavy volumes of data which are arranged in rows and columns as seen in spreadsheet program or in database. Beyond a traditional application's requirements for storing data, the Android application life cycle demands rapid and consistent persistence of data for it to survive the volatility of the mobile environment, where devices can suddenly lose power or the Android operating system can arbitrarily decide to remove your application from memory.

Ahead of a traditional application's requirements for storing data, the Android application life cycle demands quick and consistent persistence of data for it to survive the uncertain mobile environment, where the devices can suddenly lose power or the Android operating system can arbitrarily decide to remove your application from memory.

Database tables are a natural adjustment for data having various instances of same kind of thing that occurs commonly in software development. It is noted a contact list carry many contacts that carries similar type of information such as address, mobile, etc. where each row of data keeps information regarding person with its attributes stored in every column.

The Android framework offers several options and strategies for persistence:

- Shared Preferences: It will save basic data as key-value pairs in private persisted dictionary.
- Local Files – It saves arbitrary files to internal or external device storage.

- SQLite Database: It persist data in tables inside application specific database.
- ORM: it shows and persist model objects with higher level query/update syntax.

Each storage option has typical associated use cases as follows:

- Shared Preferences: it is applied for app preferences, keys and session information.
- Local Files: It is used for data file caches
- SQLite Database: It is applied for complex data execution or for raw speed
- ORM: It stores relational data locally to lower SQL boilerplate

Android can read/write files to internal as well as external storage. Applications have access to an application-specific directory where preferences and sqlite databases are also stored. Every Activity has helpers to get the writeable directory. File I/O API is a subset of the normal Java File API. Most Android apps will need to persist user data many times. There are many ways to store user data, but SQLite databases are a very convenient and speedy method of saving user data and information. It is an open source SQL database which keeps database in form of text file on mobile device. It is a type of relational database management system similar to Oracle, MySQL and PostgreSQL.

Check your progress 1

1. Shared Preferences applies to:

    a. application preferences

    b. application keys

    c. session information

    d. all of above

2. SQLite is not like:

    a. PHP

    b. Oracle

    c. MySQL

    d. PostgreSQL

1.3 A Content Provider as a Facade for a RESTful Web Service

Content Provider receives data from central repository. It is found that Android application has content provider that will deliver data to other applications. They will form an abstraction layer among its repository of data and external application which uses data. In case of external application, this will call Content Provider methods using Content Resolver. The Content Resolver will work as Content Provider client object along with Content Resolver object that are used to receive data from Content Provider. It is seen that both Content Provider and Content Resolver both will form an interface for data so as to handles inter-process communication and also to work with data in secret manner.

The Content Provider is not commonly usable main Android components. It is noted that some Android developers heard about it and further applies one or two which is present in the system in order to get back user's calendar data or contact information. They are particularly designed in order to facilitate applications to get open as well as can share data across application processes. It is not an Object Relational Mapper nor needs any particular storage mechanism which can be sqlite, but serves as part which opens simple REST-like API for a data store.

Once Content Provider is set at its position, it establishes REST client and further will make request to server. It can be applied in an app simply by inserting some dummy data which can be utilized for coding on certain parts of an application without any problem of external dependency. While creating a software you have to look on several aspects. The work involving creating an application and feeling about its features will open everything that relates to certain changes in internal model before running an application at the start. With this, lots of redundant work is saved which can lead to fetching, parsing and adapting of external data used for altering internal data model.

Representational State Transfer is a professional style which distributes hypermedia systems which can be World Wide Web. Central to RESTful layout is the feature of resources that is located by universal resource identifiers (URIs). Such resources gets calculated using standard interface which can be HTTP protocol or information is exchanged using representations of these resources.

RESTful web services are services built using the RESTful architectural style. Building web services using the RESTful approach is emerging as a popular alternative to using SOAP-based technologies for deploying services on the internet, due to its lightweight nature and the ability to transmit data directly over HTTP.

RESTful web services in Java depend on Java Persistence API in order to interact with database. Moreover, particularly RESTful web services depends on entity classes and on persistence unit as explained in Persistence API. Here, entity classes maps the objects in relational database which shows a table in relational database with each entity instance corresponds to row in particular table. The persistence unit comprises of set of entity classes, data source, persistence provider, and persistence units own name as specified in persistence.xml file.

Check your progress 2

1. The Content Provider gets recharged by:

   a. Intent

   b. SQLite

   c. Content Resolver

   d. None of above

1.4 Using Content Providers

A content provider manages access to a central repository of data. A provider is part of an Android application, which often provides its own UI for working with the data. However, content providers are mainly applied for other applications that access provider by provider client object. Both, providers and provider client's results consistent, standard interface to data in handling inter-process communication and safe data execution.

A content provider presents data to external applications with one or many tables which are identical to tables available in relational database. A row represents an instance of some type of data the provider collects, and each column in the row represents an individual piece of data collected for an instance.

A content provider component gives data from single application to others based on request. Such requests are handled by methods of ContentResolver class.

A content provider can use different ways to store its data and the data can be stored in a database, in files, or even over a network.

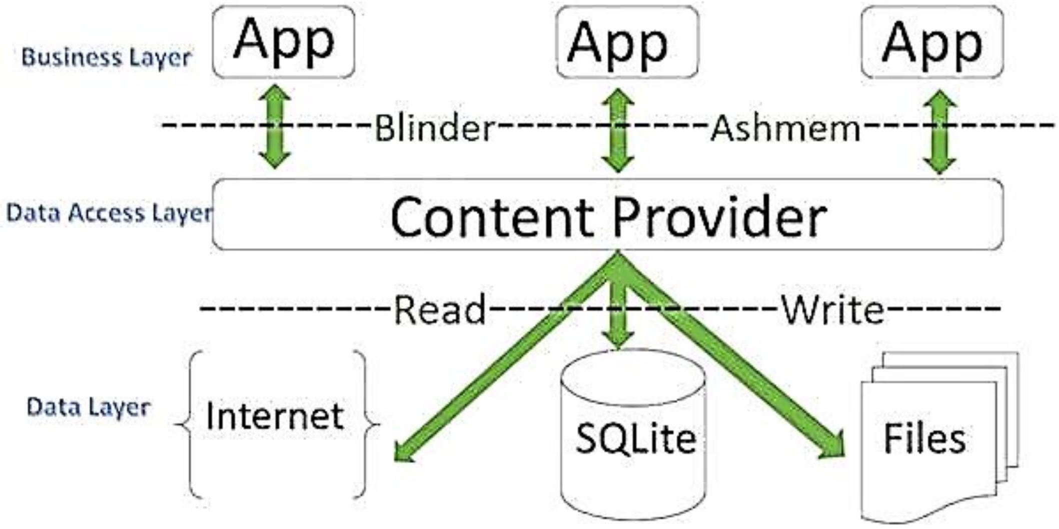


Fig 1.1 Content Provider

Content providers allows to keep content in central position in single place where it can be accessed by various applications. A content provider poses similar to database where it can be queried and edited the content along with adding or deleting of content using insert(), update(), delete(), and query() methods. In most cases this data is stored in a SQlite database. A content provider is implemented as a subclass of ContentProvider class and must implement a standard set of APIs that enable other applications to perform transactions.

```
public class My Application extends  ContentProvider {
}
```

Create Content Provider

There are simple process to create content provider:

- Initially, create Content Provider class which extends ContentProviderbaseclass.
- Define content provider URI address so as to work on content.
- Create database to keep content as in Android, SQLite database and framework gets override by onCreate() method applying SQLite Open Helper method for provider's database.
- Execute Content Provider queries for doing various database operations.
- Register the Content Provider in activity file with <provider> tag.

Content Provider class can be overridden by following standards:

- onCreate(): This method is called when the provider is started.
- query(): This method receives a request from a client. The result is returned as a Cursor object.
- insert(): This method inserts a new record into the content provider.
- delete(): This method deletes an existing record from the content provider.
- update(): This method updates an existing record from the content provider.
- getType(): This method returns the MIME type of the data at the given URI.

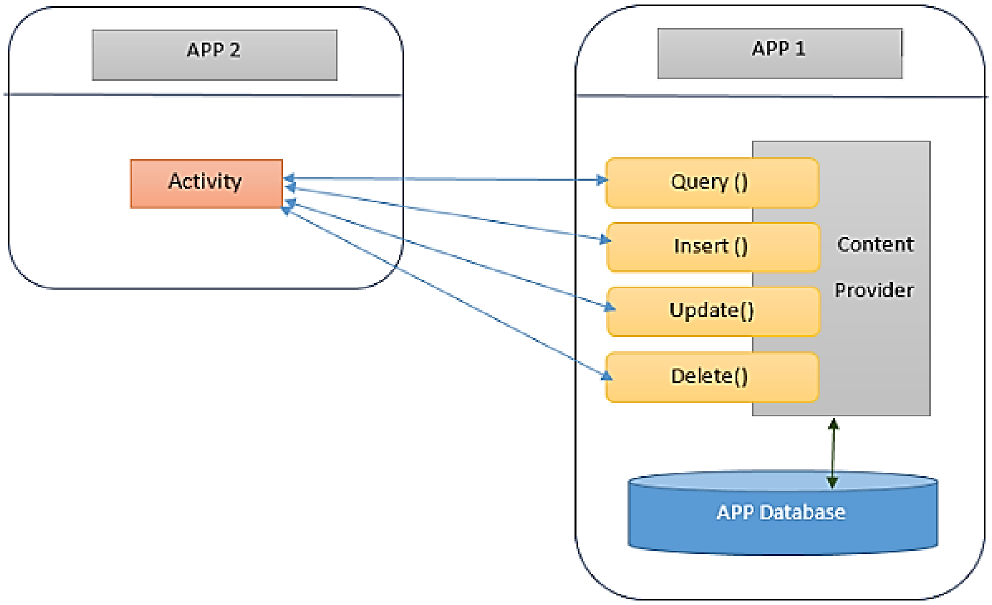


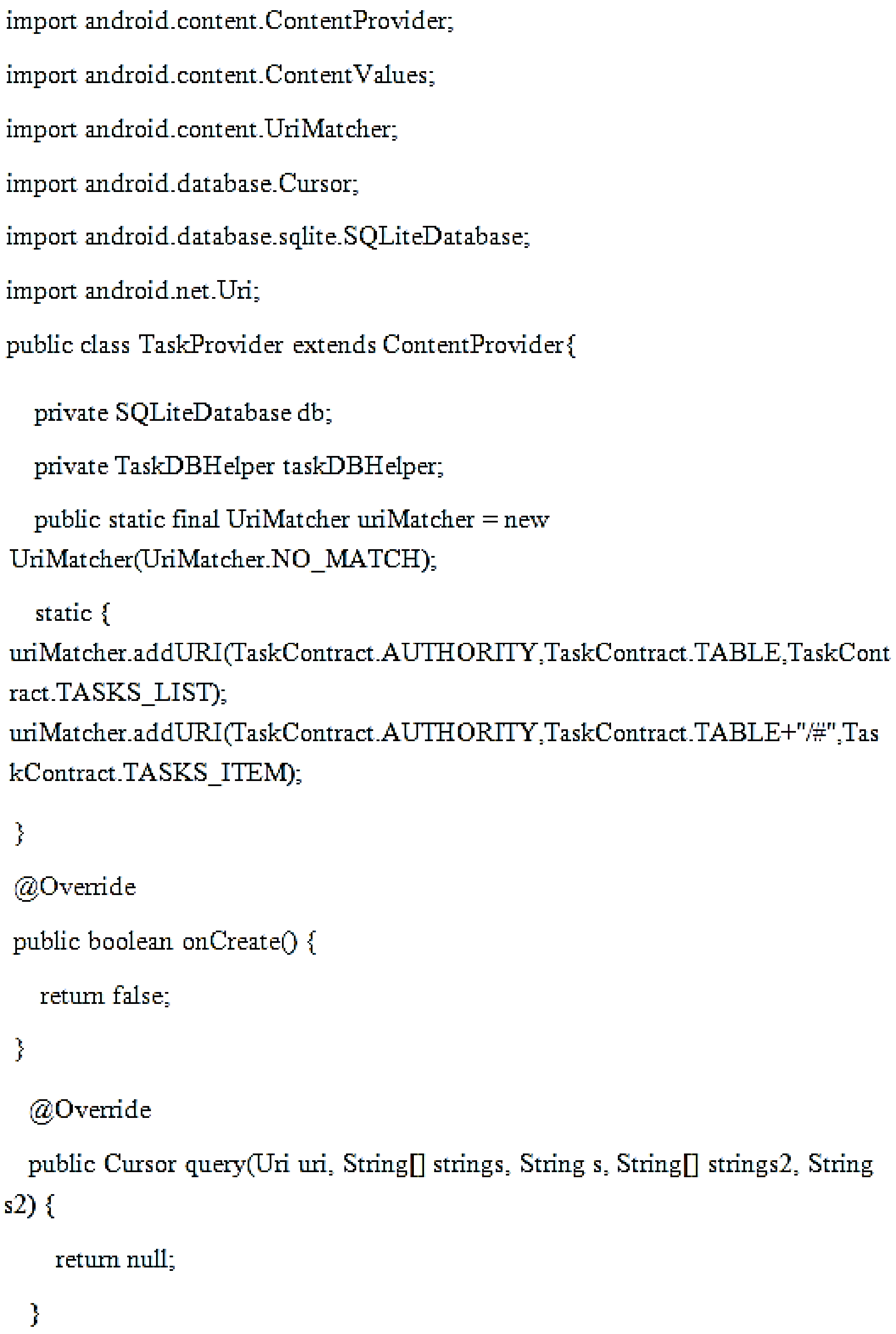
Fig 1.2 Standards for content provider

In fig 1.2, it is described how content provider works with two applications. App 1 will keep data in its database and give it to the provider, while App 2 will interact with provider to work on App 1's data. It is found that Content providers serves as simple interface that uses standard such as insert(), query(), update(), delete() methods to access application data. So it is easy to implement a content provider.

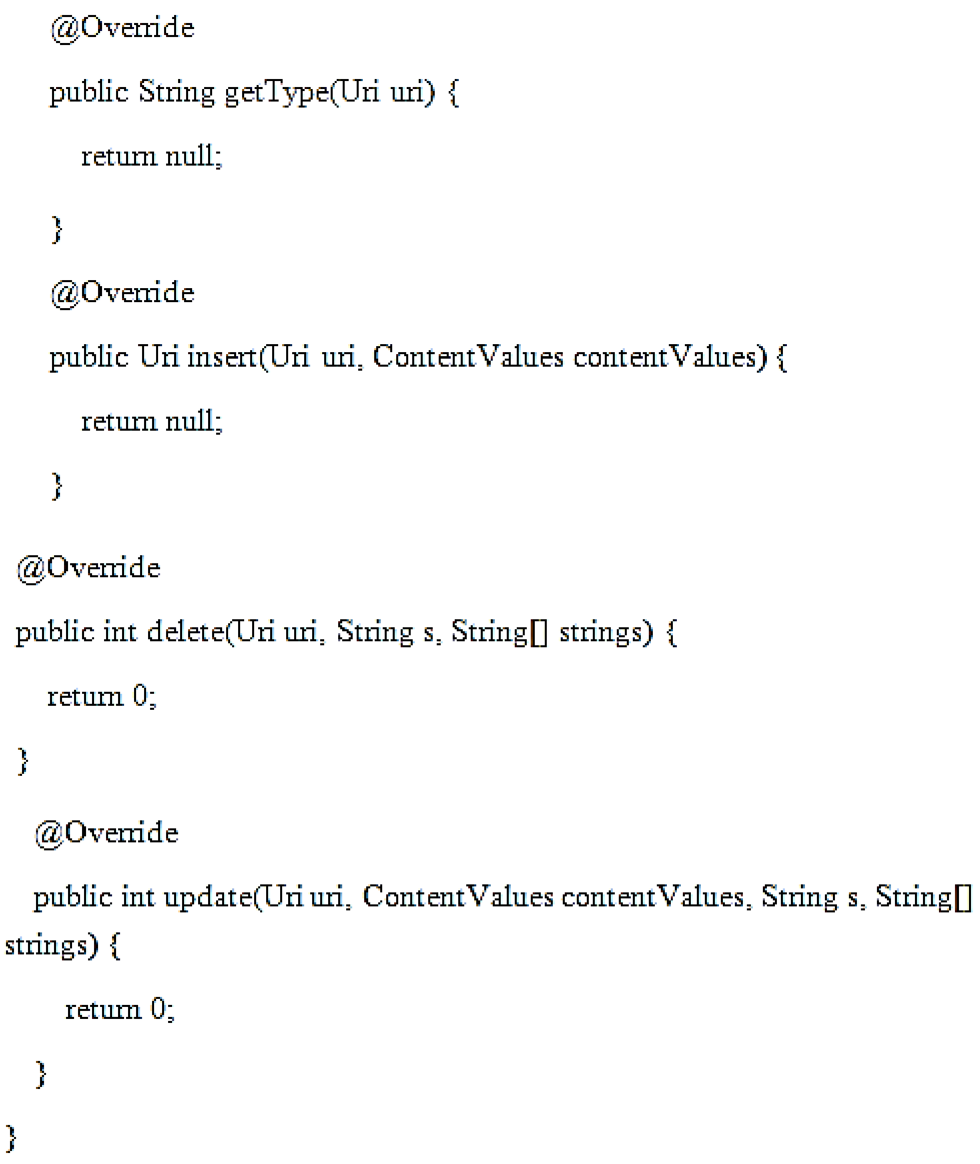
Creating new class

It is noted that content provider serves as Java class that widen ContentProvider class and implements its methods. It is called as TaskProvider, which places in db package. For creating new class file in TaskProvider.java you have to use the following code:

```
package com.example.TodoList.db;
```



```
import android.content.ContentProvider;
import android.content.ContentValues;
import android.content.UriMatcher;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.net.Uri;
public class TaskProvider extends ContentProvider{
    private SQLiteDatabase db;
    private TaskDBHelper taskDBHelper;
    public static final UriMatcher uriMatcher = new
UriMatcher(UriMatcher.NO_MATCH);
    static {
uriMatcher.addURI(TaskContract.AUTHORITY,TaskContract.TABLE,TaskCont
ract.TASKS_LIST);
uriMatcher.addURI(TaskContract.AUTHORITY,TaskContract.TABLE+"/#",Tas
kContract.TASKS_ITEM);
 }
 @Override
 public boolean onCreate() {
    return false;
 }
   @Override
   public Cursor query(Uri uri, String[] strings, String s, String[] strings2, String
s2) {
       return null;
   }
```



```
    @Override
    public String getType(Uri uri) {
        return null;
    }
    @Override
    public Uri insert(Uri uri, ContentValues contentValues) {
        return null;
    }
  @Override
  public int delete(Uri uri, String s, String[] strings) {
      return 0;
  }
   @Override
   public int update(Uri uri, ContentValues contentValues, String s, String[]
strings) {
        return 0;
   }
}
```

In this, we see that such class will not do much. Since it is a content provider, but it doesn't provide any content because of non-execution of methods. Further it will form UriMatcher instance that is applied to check whether URI accessed is valid or not.

Check your progress 3

1. Which among the following is not charged by an Intent?

   a. broadcast receiver

   b. services

   c. contentProvider

   d. activity

2. ContentProvider uses _______methods for altering the content.

   a. insert()

   b. update()

   c. query()

   d. all of above

1.5 Let Us Sum Up

In this unit we have learnt that Android provides many options to save persistent application data which is as per needs that rely on data-type. It is found that many Android applications work with variety of data types that uses handling and persisting of data in an application.

The ContentProvider gets data from central repository as in case of Android application, the contentprovider delivers data to several applications. It is found that contentprovider manages access to central repository of data and provides Android application having own UI for working with certain data.

1.6 Answers for Check Your Progress

Check your progress 1

Answers: (1 -d), (2 –a)

Check your progress 2

Answers: (1 -c)

Check your progress 3

Answers: (1 –c), (2 –d)

1.7 Glossary

1. **Application -** In Android application, there are activities, services, listeners and intent receivers.

2. **Content Provider -** It is a data abstraction layer which opens an application data with other applications.
3. **Broadcast Receiver -** It is an application class which listens for Intents which is broadcast instead of sending to single target application/activity.

1.8 Assignment

Explain the working of ContentProvider in altering of data.

1.9 Activities

Study about various standards for content provider.

1.10 Case Study

Study about Handling and Persisting Data in Android in detail.

1.11 Further Readings

1. Learn Java for Android Development (2nd edition), Jeff Friesen, 2013.
2. Android Application Development for Java Programmers, James C. Sheusi, 2012.
3. Android Application Development for Dummies (2nd edition), Micheal Burton, Donn Felker, 2012.
4. Android Programming: The Big Nerd Ranch Guide, Bill Phillips, Brian Hardy, 2013.

UNIT 2: ANDROID APPLICATION DESIGN ESSENTIALS - I

Unit Structure

2.0 Learning Objectives

After learning this unit, you will be able to understand:

- about features of Android Application
- about marketing of Android application

2.1 Introduction

Android application publishing is a process that makes your Android applications available to users. Infect, publishing is the last phase of the Android application development process.

For designing great Android applications you should follow the following themes.

- Design for touch
- Show only what is needed

- Don't ask for permission for a certain action but allow the user to undo his action
- Only interrupt, if it is important
- Keep messages brief, use pictures
- Never lose your users data
- Allow users to make important things fast
- If it looks the same, it should act the same.

2.2 Deploying Android Application to the World

In order to make your application to work across world, you need to release it through application marketplace such as Google Play. Also, the application after testing can be sent directly to end-users or can be download from website. For deployment of application, you have to do complete publishing process which is shown in fig 2.1 publishing process.

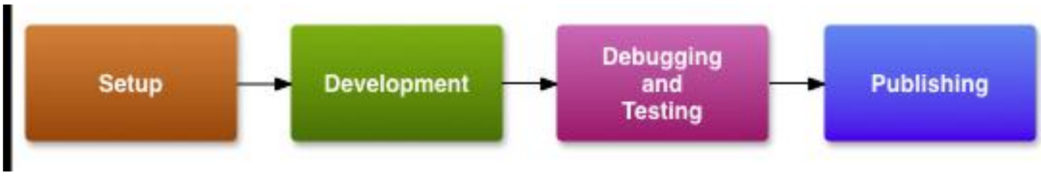


Fig 2.1 Publishing Process

Figure 2.1 describes publishing process which can be performed after testing of an application in debug environment. Further you need to pass the application before deploying in marketplace through release criteria which checks for functionality, performance and stability of application. The release process will involve multi-steps with following tasks:

- Configuring an application for release: Initially you have to omit Log calls and remove android:debuggable attribute available in manifest file. Here you need to give values for android:versionCode and android:versionName attributes. Apart from this, you have to configure settings in order to meet Google Play requirements for release an application.
- Creating and Signing of release version for application: In this, apply Gradle build files with release build type so as to create and sign release version of an application.

- Testing release version of application: Before launching in market, check thoroughly its release version on handset device and on tablet device.
- Updating of application for release: Make sure that every application resources such as multimedia files, graphics are updated and are covered in an application.
- Preparing remote servers and services for an application: If your application works with external servers or services, in such case, make sure that they are secure and ready for production.

After performing all checks on your application, you can release Android applications in many ways. Normally, it is seen that such type of applications release is mostly applied in application marketplace such as Google Play apart from releasing it on website or directly to consumer.

Releasing through App Marketplace

If you wish to sale your application to broadest audience, in such case, releasing application by Google Play is the best. It is found that Google Play appears to be premier marketplace for Android applications which is mainly for distribution of designed application for use to large audience across the globe. Apart from this, the designed applications can be distributed through any application marketplace or multiple marketplaces.

Releasing Your Applications on Google Play

Google Play is strong publishing platform which helps in publicizing, selling and distribution of Android applications for consumers across the globe. On releasing of an application via Google Play, you need to work upon suitable developer tools which help in finding sales, market trend and look for recipient of particular applications. With rich array of tools and features with many end-user features allows Google Play as good marketplace for selling and buying of Android applications.

Check your progress 1

1. Google Play is an:

   a. application

   b. platform

   c. file

   d. all

2. Publishing process does not involves:

   a. selling

   b. debugging

   c. setting

   d. development

2.3 Selling your Android application

Once an application is developed and tested on mobile devices by checking all releases, it is now ready for sell and distribution. The application can be marketed by various means. It can be done either by Google App or can be directly sent to the user by email or can be allowed for download by consumer from website. Selling an application is basically allowing the rights of an android application to be followed by consumers on basis of generating profit. The application can be sold through permissions from Android. Below are two of the top app business selling websites that you may consider using:

Appbusinessbrokers – It is a brokering firm that will help in selling of app with clearing certain formalities regarding the rights and ownership of app.

Apptopia – It is a website that will list app and sells it. It is a free web portal that allows customer to take product with listed price. On this, the app needs to be applied with all price.

Check your progress 2

1. Selling of Android Application can be done through:

   a. website

   b. Google play store

   c. email

   d. all

2.4 Let Us Sum Up

While studying this unit, Android application publishing is a process that makes your Android applications available to users. It is noted that to make an application to work across world, you need to release it through application marketplace such as Google Play.

It is known that in order to sale and application to broadest audience, in such case, releasing application by Google Play is the best. Google Play is strong publishing platform which helps in publicizing, selling and distribution of Android applications for consumers across the globe.

2.5 Answers for Check Your Progress

Check your progress 1

Answers: (1 –b), (2 –a)

Check your progress 2

Answers: (1 -d)

2.6 Glossary

1. **Activity -** An application screen that supports Java code from Activity class.
2. **Application -** In Android application, there are activities, services, listeners and intent receivers.

3. **Content Provider -** It is a data abstraction layer which opens an application data with other applications.

2.7 Assignment

Write short note on publishing process of an application in Android.

2.8 Activities

How will you market your own designed application in Android?

2.9 Case Study

How Google Play will work?

2.10 Further Readings

1. Learn Java for Android Development (2nd edition), Jeff Friesen, 2013.
2. Android Application Development for Java Programmers, James C. Sheusi, 2012.
3. Android Application Development For Dummies (2nd edition), Micheal Burton, Donn Felker, 2012.
4. Android Programming: The Big Nerd Ranch Guide, Bill Phillips, Brian Hardy, 2013.

Block Summary

In this block, you have learnt and understand about the basic of selling concept of designed Android application among audience with certain releasing features. The block gives an idea on the study and concept of SQLite method applied in android application. You have been well explained on the concepts of activities and applications in Android platform.

The block detailed about the role of contentprovider in marketing and designing of applications. The concept related to publishing platform and knowledge on Google Play as application marketplace are explained to you. You will be demonstrated about ways of designing an application.

Block Assignment

Short Answer Questions

1. What is the role of contentprovider?
2. Explain the function of Google play?
3. Write note on methods applied to alter data in Android?
4. Write short note on SQLite method?

Long Answer Questions

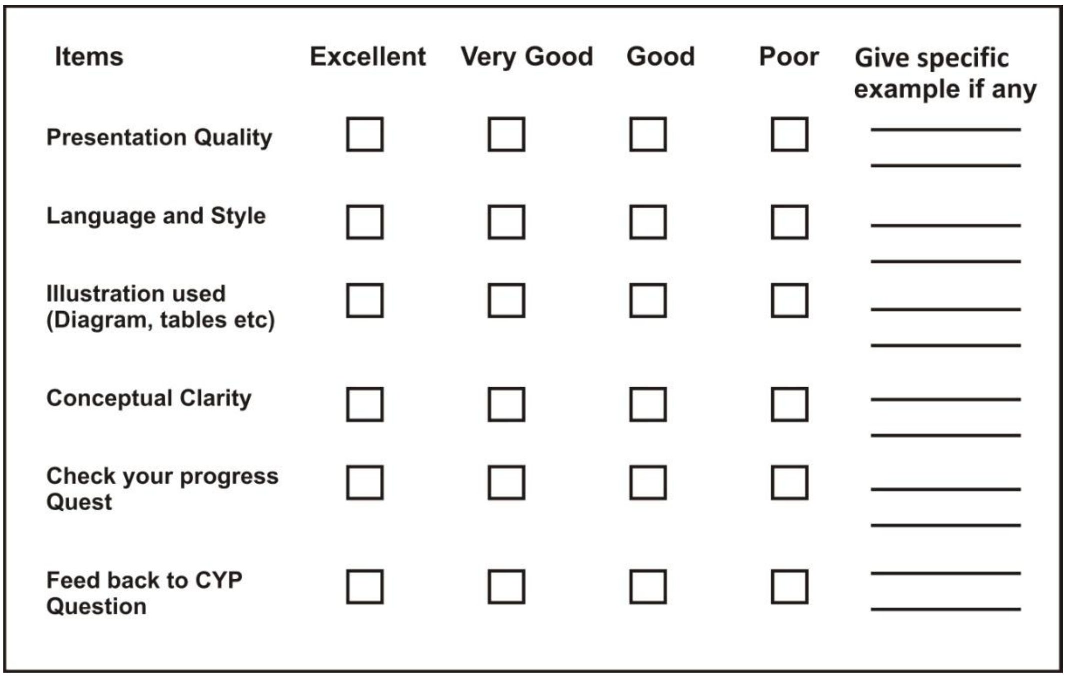
1. Write short notes on marketing of Android application?
2. Write short note on various marketing websites used in marketing of designed Android application?
3. Write note on Manifest File in Android?

Enrolment No. □

1. How many hours did you need for studying the units?

Unit No	**1**	**2**	**3**	**4**
Nos of Hrs				

2. Please give your reactions to the following items based on your reading of the block:



Items	Excellent	Very Good	Good	Poor	Give specific example if any
Presentation Quality	□	□	□	□	________ ________
Language and Style	□	□	□	□	________ ________
Illustration used (Diagram, tables etc)	□	□	□	□	________ ________
Conceptual Clarity	□	□	□	□	________ ________
Check your progress Quest	□	□	□	□	________ ________
Feed back to CYP Question	□	□	□	□	________ ________

3. Any Other Comments

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

> *Education is something which ought to be brought within the reach of every one.*

- Dr. B. R. Ambedkar

Dr. Babasaheb Ambedkar Open University
Jyotirmay' Parisar, Opp. Shri Balaji Temple, Sarkhej-Gandhinagar Highway, Chharodi,
Ahmedabad-382 481.

Post Graduate Diploma in Computer Application

Lab Manual

For

Mobile Application Development

Course Code: PGDCA – 203

Dr. Babasaheb Ambedkar Open University

Gujarat

Contents

Introduction of Mobile Application Development

Android continues an open source as well as Linux-based operating system for mobile instruments comparable as smartphones along with tablet computers. Android had been constructed immediate the Open Handset collaboration, commanded by Google, along with external companies. Android endures an Operating System for mobile appliances constructed proximate Google, which continues developed upon Linux kernel. Android applies with Apple's iOS (for iPhone/iPad), RIM's Blackberry, Microsoft's Windows Phone, Symbian OS, along with abundant alien proprietary mobile OSes.

Android platform exists disinterred close-at-hand both numerals: a version identity (x.y) along with an API level integer (a moving character commences from 1, which exists applied in the Android Market/Google Play to identify new version). Android contributes you everything you expect to develop best-in-class app experiences.

Android provides many options to save persistent application data which is as per needs. It depends on type of data which can be private for an application or visible or usable to other applications along with storage space. It is found that many Android applications work with wide variety of data types. It can be done using handling and persisting of data inside an application.

Importance

- The Android SDK appliances constitute your code—along with numerous data along with resource files—into an APK: an Android package, which continues an archive file with an .apk suffix.
- The Android platform endures a platform for mobile appliances that facilitates a changed Linux kernel.
- The Android Platform had been commenced near the Open Handset coalition in November relevantly 2007.
- Android application publishing is a process that makes your Android applications available to users.
- Android provides a way for you to open private data to other applications with the help of content provider.
- Android provides many options to save persistent application data which is as per needs. It depends on type of data which can be private for an application or visible or usable to other applications along with storage space.

Objective of Lab Manual

After completion of this lab, learners will:

- Features of installation of Android Studio.
- Framework for Well-Behaved Application
- Concept of Intent Filter
- Learn characteristics of deploying Android Application
- Learn idea about Content Providers

Introduction of LAB

There are 50 systems installed in computer Lab. Their configurations are as follows:

Processor	:	Pentium 4 or faster
RAM	:	2 GB
Hard Disk	:	40 GB
Mouse	:	Optical Mouse
Operating System	:	Windows XP (or latest version)
Software	:	Android SDK, JDK 1.7 or above, Android Studio
Network Interface card	:	Present

Guidelines Related To Lab

Guidelines to learners:

- Equipment in the lab for the use of learner's community. Learners need to maintain a proper decorum in the computer lab.
- Learners must use the equipment with care. Any damage is caused is punishable.
- Learners are required to carry their observation / programs book with completed exercises while entering the lab.
- Learners are supposed to occupy the machines allotted to them and are not supposed to talk or make noise in the lab. The allocation is put up on the lab notice board.
- Lab can be used in lab time decided by lab-in charge.
- Lab records need to be submitted on or before date of submission.
- Learners are not supposed to use any USB or other devices.
- Use of computer network is encouraged.

Practicals

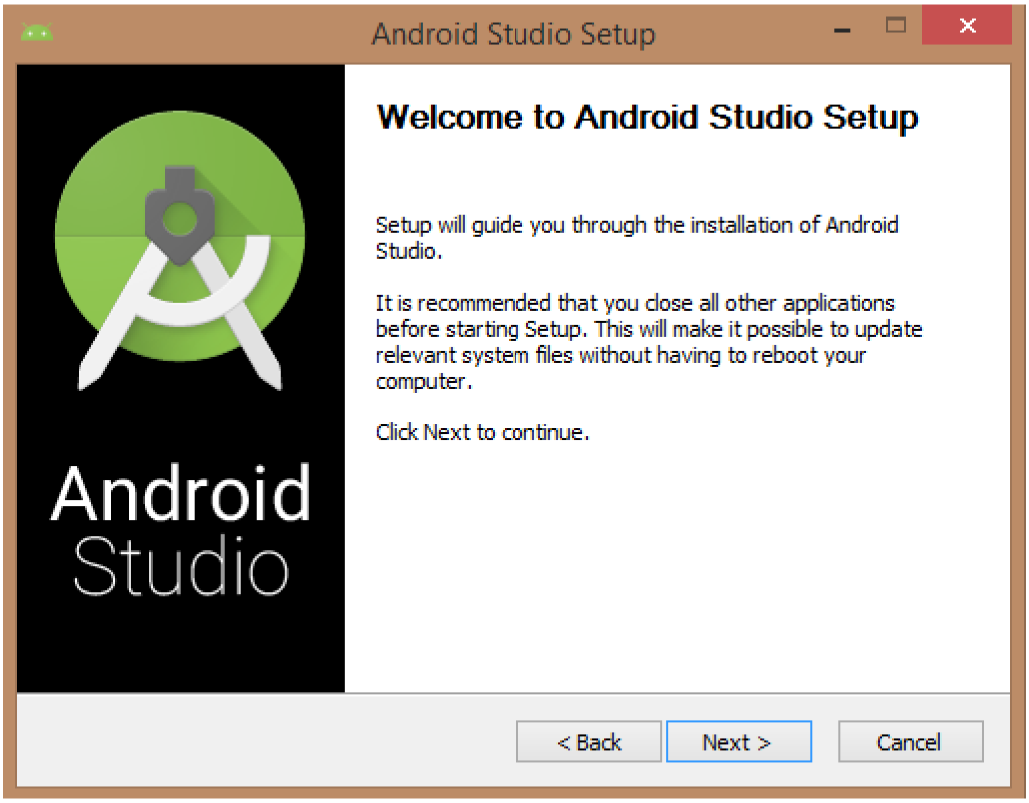
&

Their solutions

Lab 1: To install android studio.

Solution:

Start with Android Studio.exe. Before Android Studio, the computer should have Java JDK which if not can be installed taking references of Android environment setup



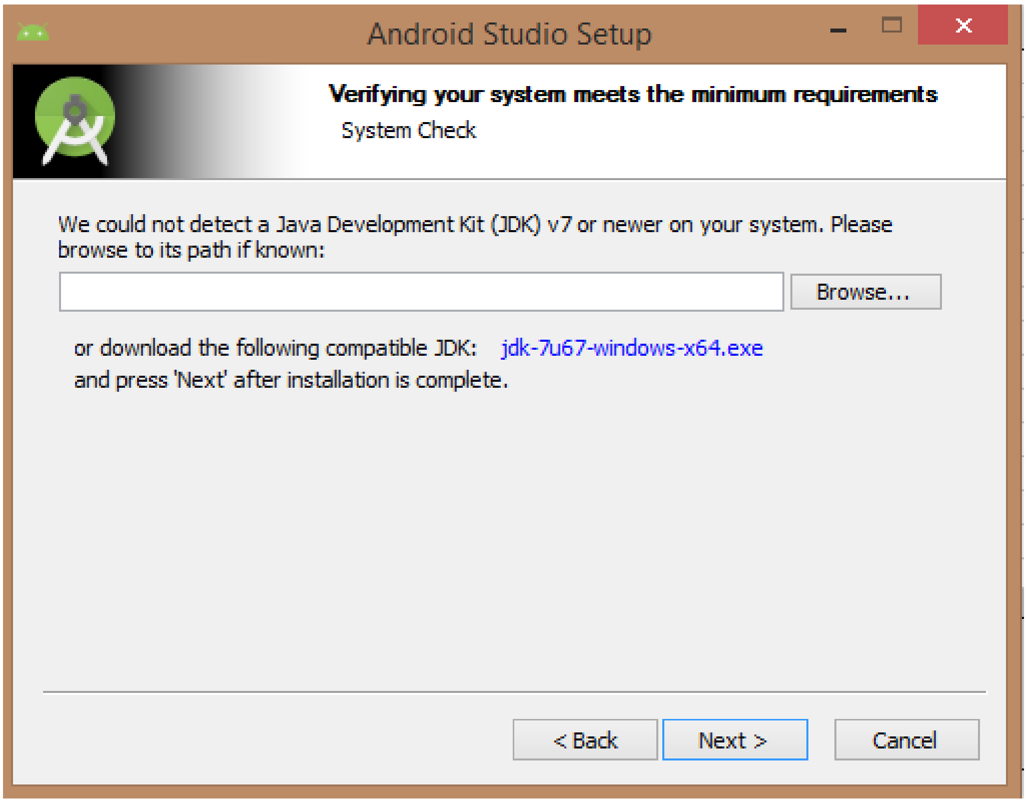
Welcome to Android Studio Setup

Setup will guide you through the installation of Android Studio.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

After launching Android Studio, show JDK5 path in android studio installer.



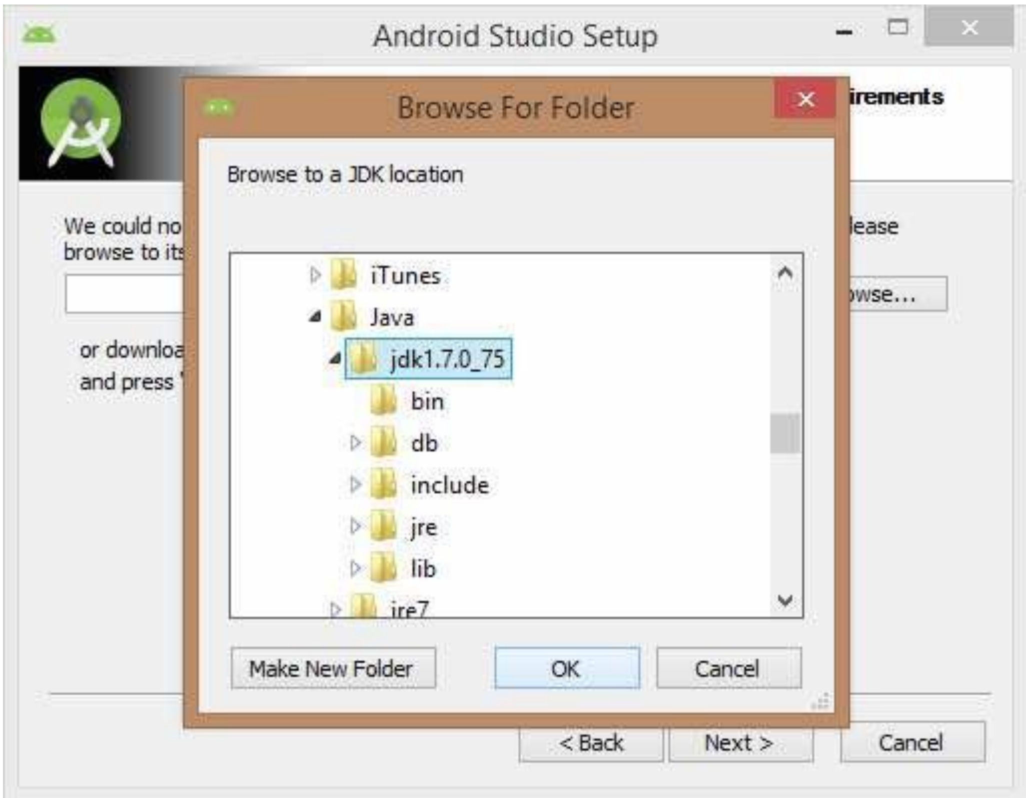
Verifying your system meets the minimum requirements

System Check

We could not detect a Java Development Kit (JDK) v7 or newer on your system. Please browse to its path if known:

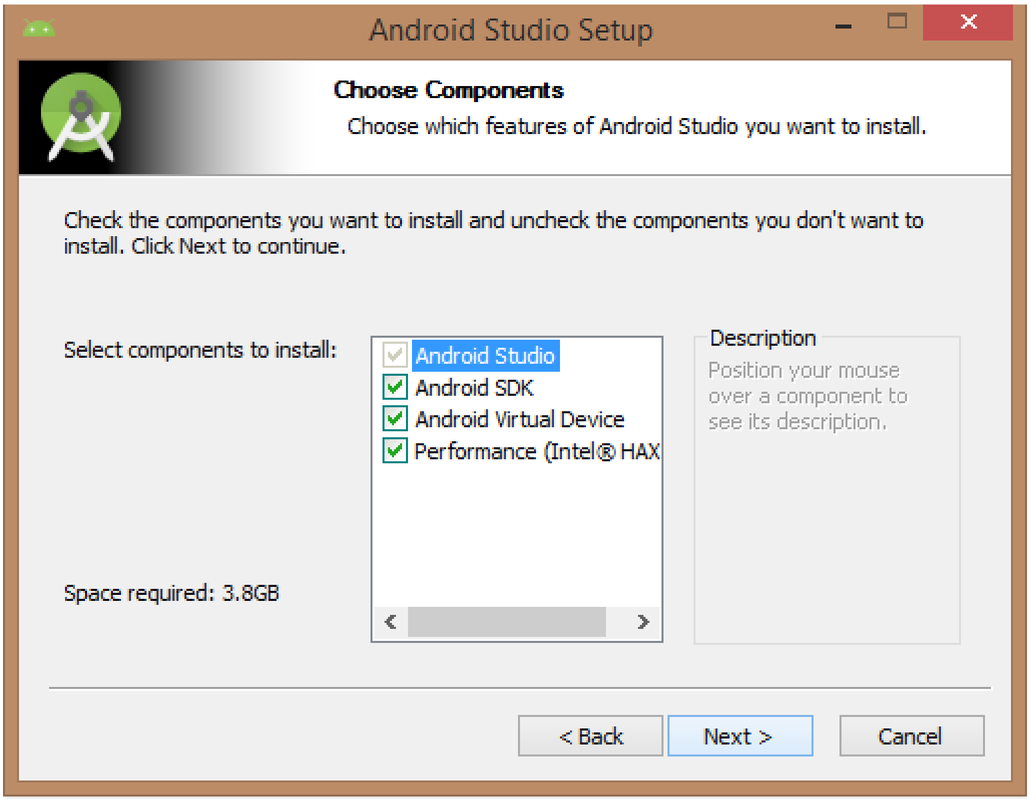
or download the following compatible JDK: jdk-7u67-windows-x64.exe
and press 'Next' after installation is complete.

Now initiate JDK to android SDK



Browse to a JDK location

After this, verify the components that frame applications a Android Studio, Android SDK, Android Virtual Machine and performance.



Choose Components

Choose which features of Android Studio you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

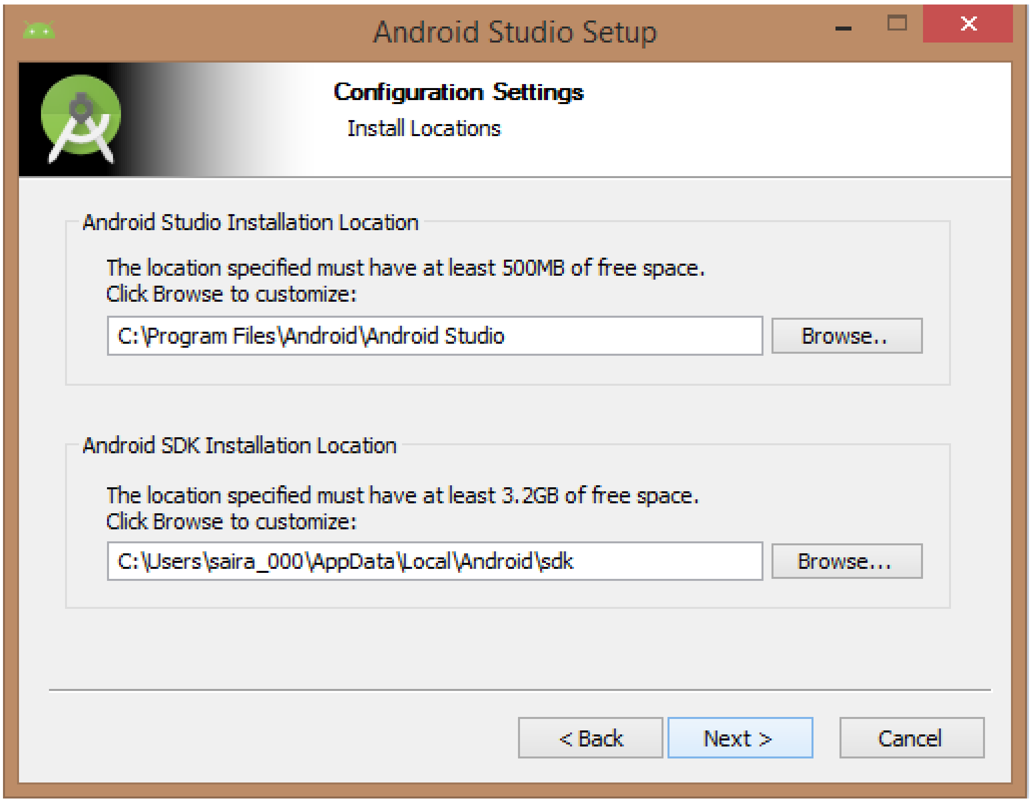
- ☑ Android Studio
- ☑ Android SDK
- ☑ Android Virtual Device
- ☑ Performance (Intel® HAX

Description

Position your mouse over a component to see its description.

Space required: 3.8GB

Now you have to select the location of local machine path for Android studio and also for Android SDK as shown:



Configuration Settings

Install Locations

Android Studio Installation Location

The location specified must have at least 500MB of free space.
Click Browse to customize:

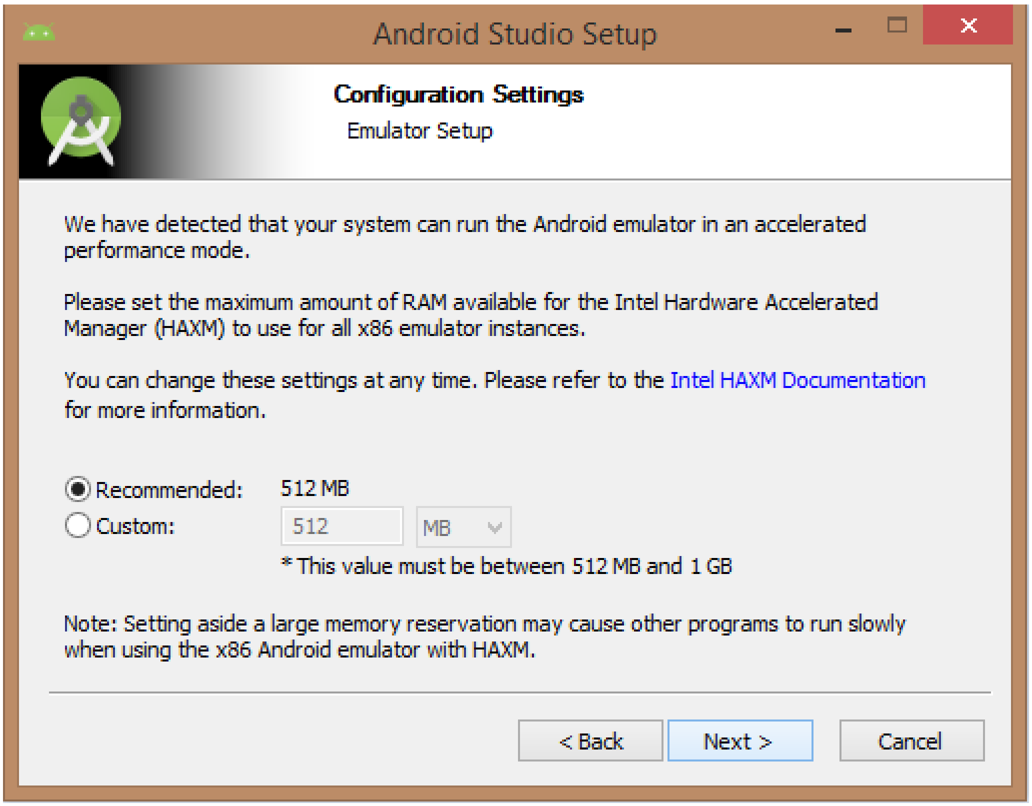
C:\Program Files\Android\Android Studio

Android SDK Installation Location

The location specified must have at least 3.2GB of free space.
Click Browse to customize:

C:\Users\saira_000\AppData\Local\Android\sdk

Now check for RAM. As the download needs minimum 512MB of RAM in the local machine.



Configuration Settings

Emulator Setup

We have detected that your system can run the Android emulator in an accelerated performance mode.

Please set the maximum amount of RAM available for the Intel Hardware Accelerated Manager (HAXM) to use for all x86 emulator instances.

You can change these settings at any time. Please refer to the Intel HAXM Documentation for more information.

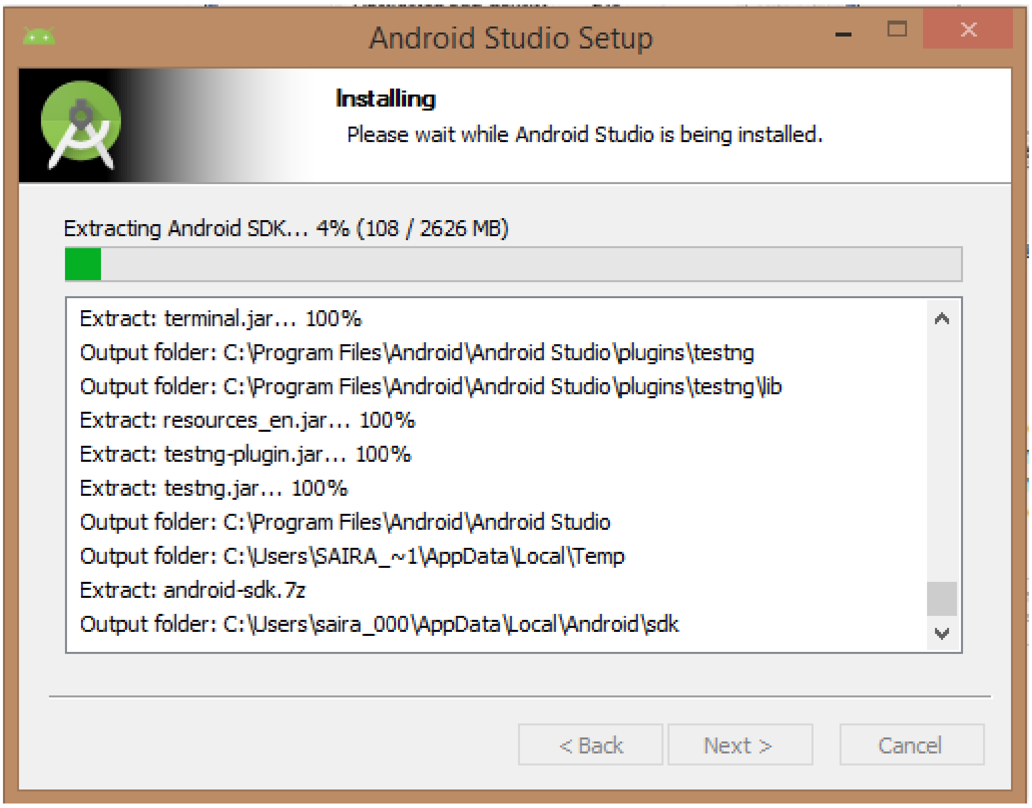
● Recommended: 512 MB

○ Custom: 512 MB

* This value must be between 512 MB and 1 GB

Note: Setting aside a large memory reservation may cause other programs to run slowly when using the x86 Android emulator with HAXM.

Now finally extract SDK packages into local machine which requires about 2626MB of Hard disk space.



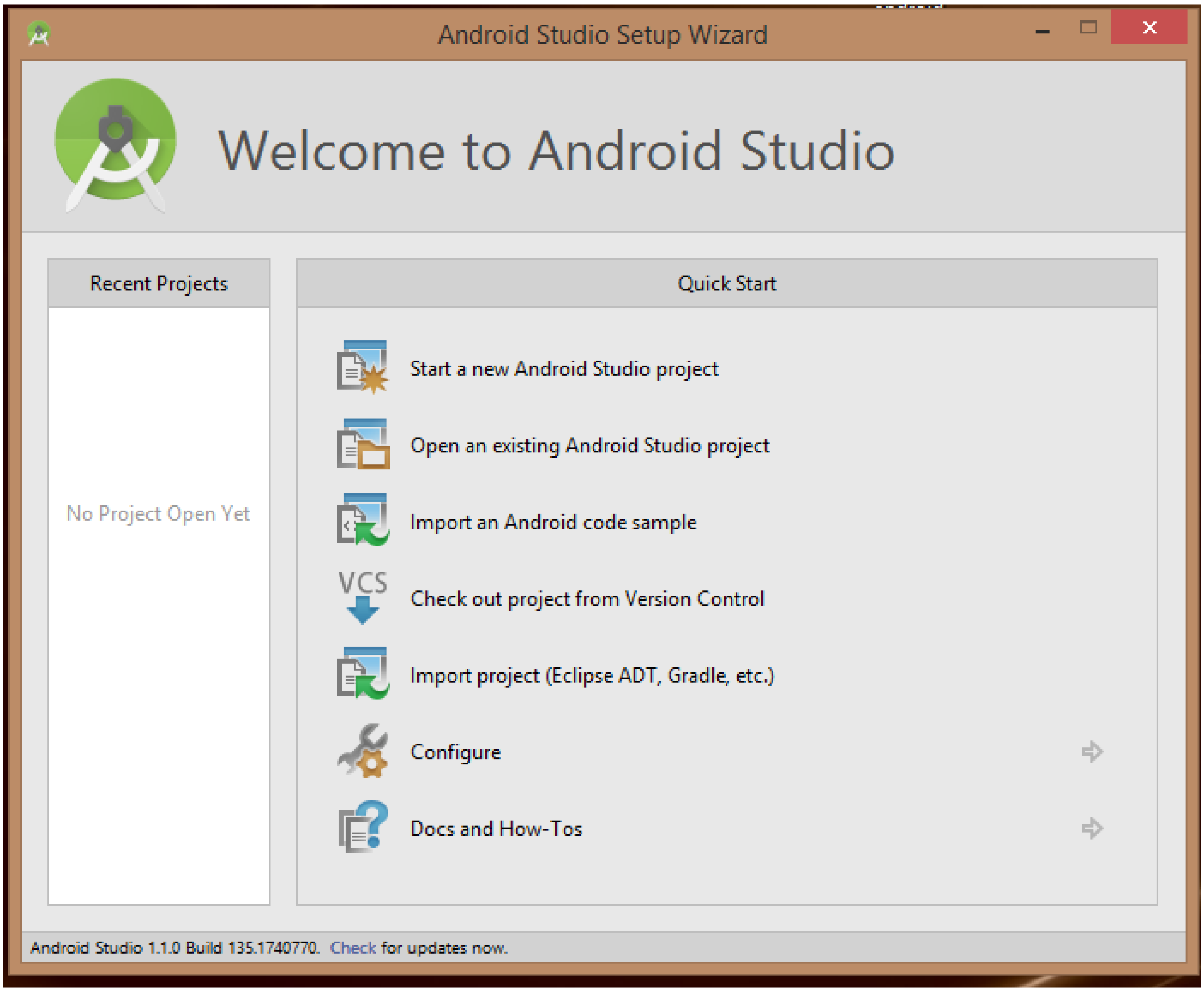
Installing

Please wait while Android Studio is being installed.

Extracting Android SDK... 4% (108 / 2626 MB)

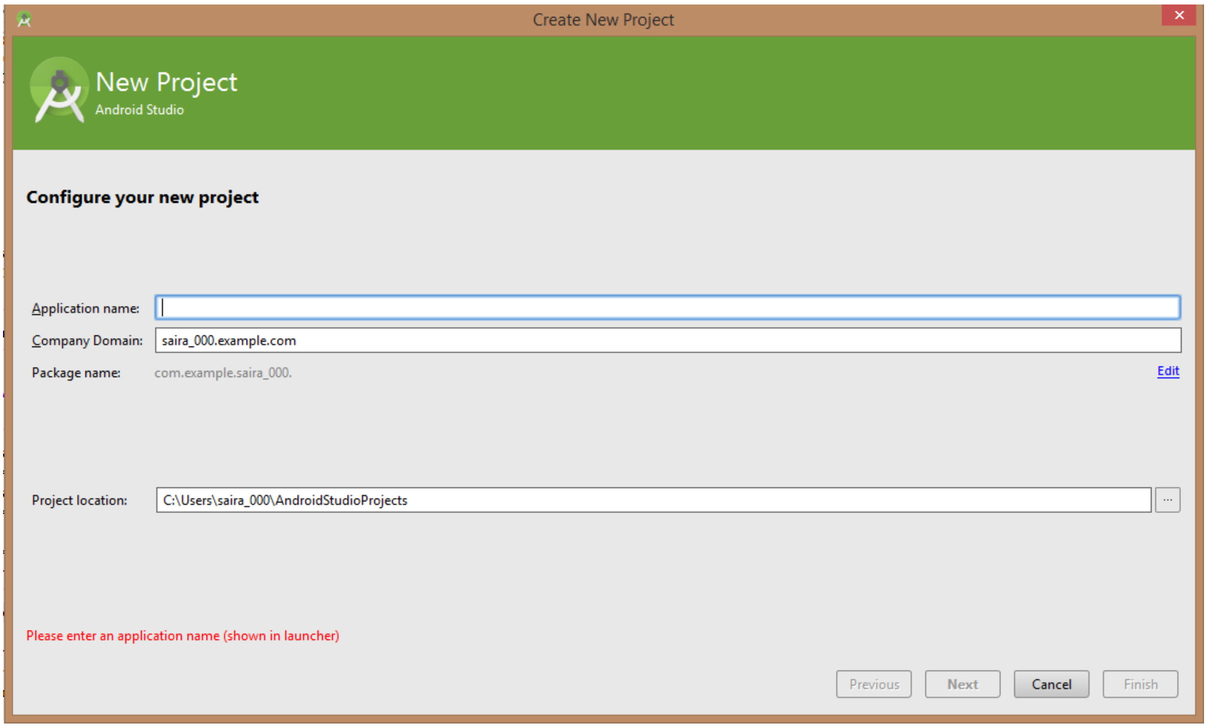
```
Extract: terminal.jar... 100%
Output folder: C:\Program Files\Android\Android Studio\plugins\testng
Output folder: C:\Program Files\Android\Android Studio\plugins\testng\lib
Extract: resources_en.jar... 100%
Extract: testng-plugin.jar... 100%
Extract: testng.jar... 100%
Output folder: C:\Program Files\Android\Android Studio
Output folder: C:\Users\SAIRA_~1\AppData\Local\Temp
Extract: android-sdk.7z
Output folder: C:\Users\saira_000\AppData\Local\Android\sdk
```

Now after it, click on finish button so that the android studio project gets started with Welcome message as shown below



Android Studio Setup Wizard
Welcome to Android Studio
Recent Projects
No Project Open Yet
Quick Start
Start a new Android Studio project
Open an existing Android Studio project
Import an Android code sample
VCS
Check out project from Version Control
Import project (Eclipse ADT, Gradle, etc.)
Configure
Docs and How-Tos
Android Studio 1.1.0 Build 135.1740770. Check for updates now.

Start your application development by calling start new android studio project. Here you need to specify Application name, package information and location of the project.



Configure your new project

Application name:

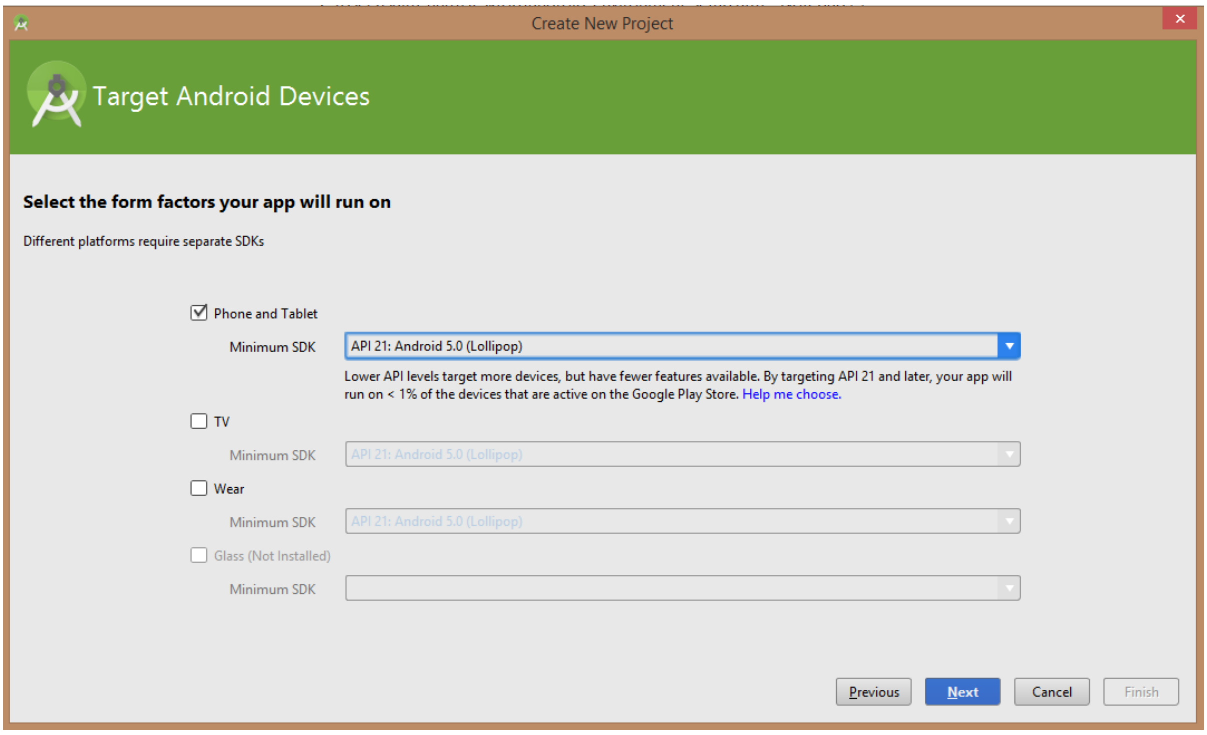
Company Domain: saira_000.example.com

Package name: com.example.saira_000. Edit

Project location: C:\Users\saira_000\AndroidStudioProjects

Please enter an application name (shown in launcher)

Once all that is entered, select form factors for an application to run.



Select the form factors your app will run on

Different platforms require separate SDKs

☑ Phone and Tablet

Minimum SDK: API 21: Android 5.0 (Lollipop)

Lower API levels target more devices, but have fewer features available. By targeting API 21 and later, your app will run on < 1% of the devices that are active on the Google Play Store. Help me choose.

☐ TV

Minimum SDK: API 21: Android 5.0 (Lollipop)

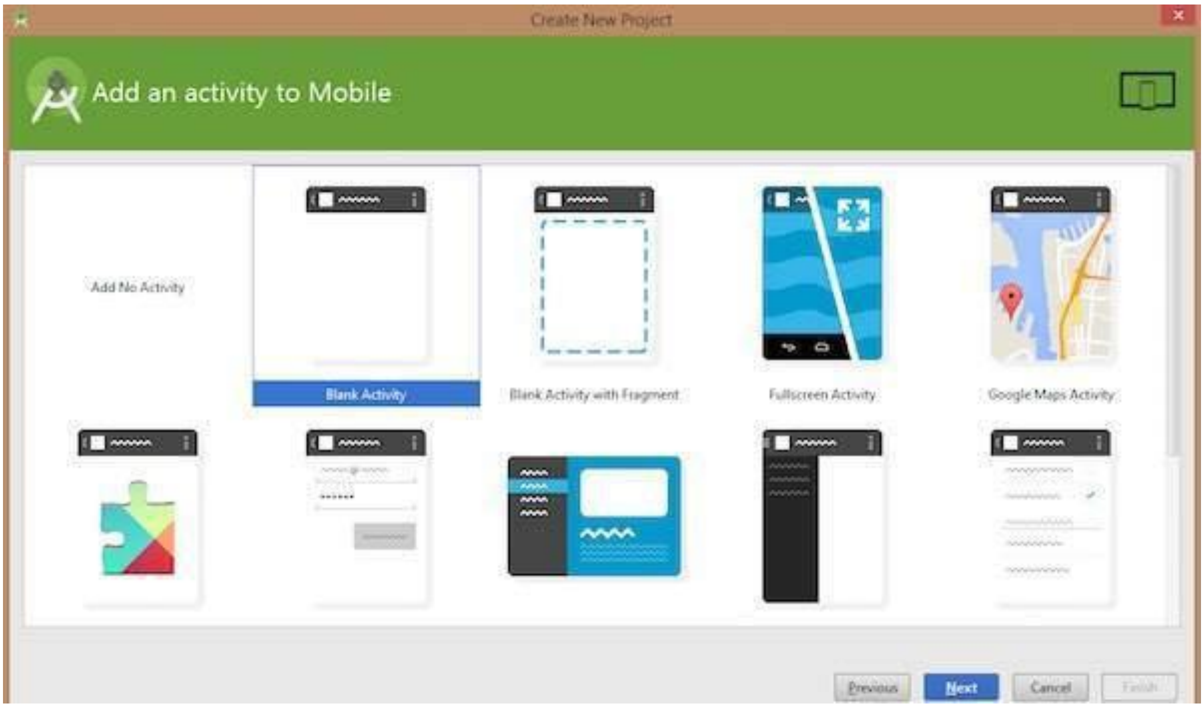
☐ Wear

Minimum SDK: API 21: Android 5.0 (Lollipop)

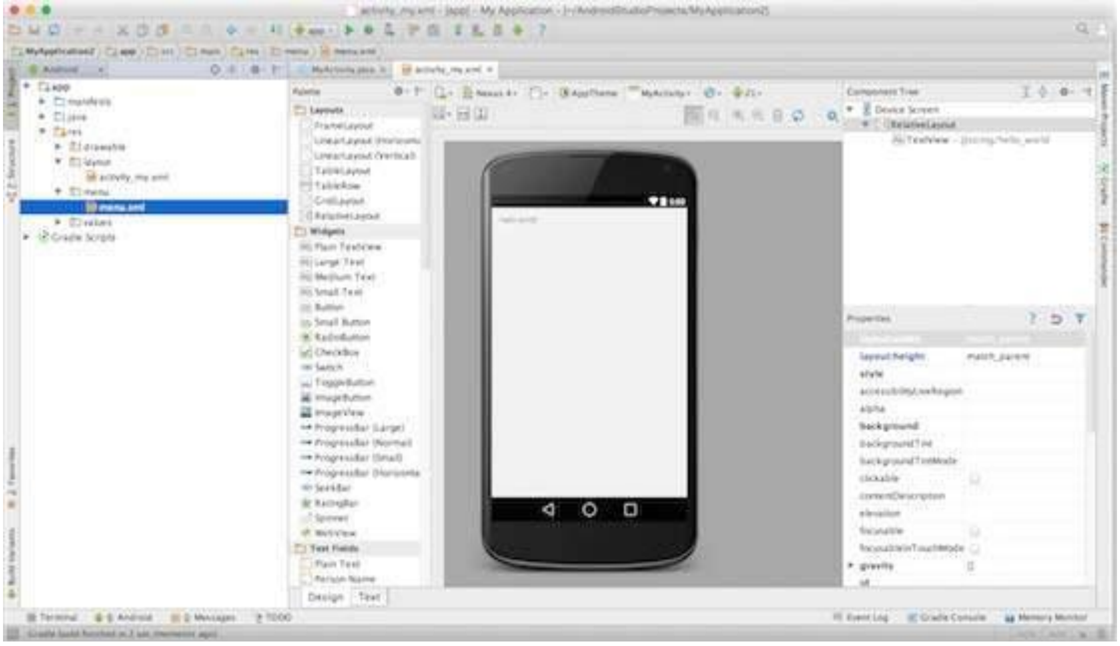
☐ Glass (Not Installed)

Minimum SDK:

Further, select activity to mobile showing default layout for Applications



Finally, open development tool to write application code.



Lab 2: Creating a project with Android Studio.

Solution:

Step 1: In Android Studio, create a new project. If you have no project shown in Welcome screen, then you have to click on New Project option. In case you have project opened, then in such case in File menu, you have to choose New Project which allows creating New Project from the screen which appears on desktop.

Step 2: Once the screen is there, you need to fill all fields and click on the Next.

- You have to give the Application Name in application name column.
- In Company domain qualifier, you have to write Company Name so that Android Studio will remember for each new project you create.
- In Package name, you have to fill name for the project which should be unique. You can Edit this independently from application name or company domain.
- In Project location, you have to specify the directory path of your system where your project files gets stored.

Step 3: Under Select the form factors your app will run on, check the box for Phone and Tablet.

Step 4: For Minimum SDK, you have to choose API 8: Android 2.2 (Froyo).

For Minimum Required SDK, you have to refer to earliest version of Android which is supported by app with API level.

Step 5: Leave options as TV, Wear, and Glass unchecked and click on Next.

Step 6: In Add an activity to <template>, select Blank Activity and click Next.

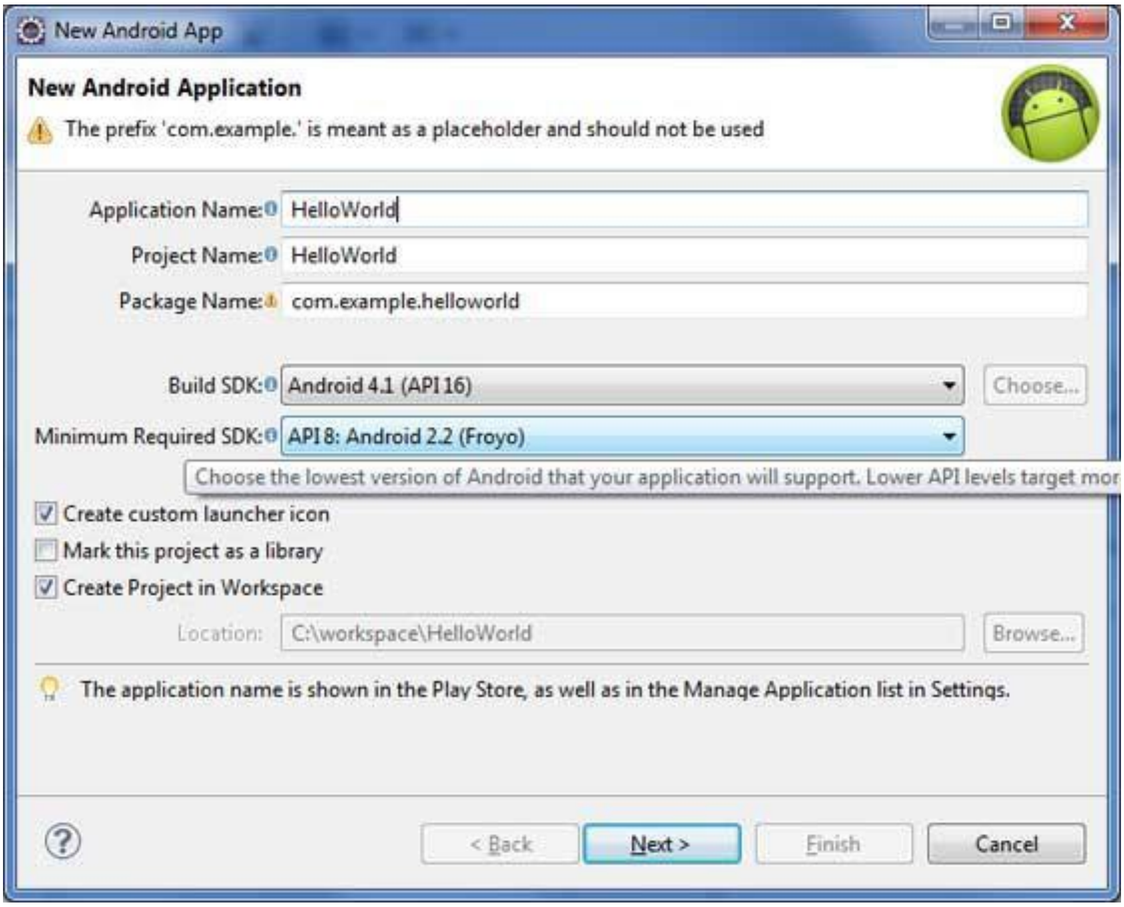
Step 7: In Customize the Activity, you can change Activity Name to MyActivity. In this, the Layout Name will change to activity_my and Title to MyActivity. The Menu Resource Name is menu_my.

Step 8: Now finally, choose the Finish button to create the project.

Lab 3: Creating a small Android application.

Solution:

The first step is to create a simple Android Application using Eclipse IDE. Follow the option File -> New -> Project and finally select Android New Application wizard from the wizard list. Now name your application as ‘Hello World’ using the wizard window as follows:



New Android App

New Android Application

The prefix 'com.example.' is meant as a placeholder and should not be used

Application Name: HelloWorld

Project Name: HelloWorld

Package Name: com.example.helloworld

Build SDK: Android 4.1 (API 16)

Minimum Required SDK: API 8: Android 2.2 (Froyo)

Choose the lowest version of Android that your application will support. Lower API levels target mor

☑ Create custom launcher icon

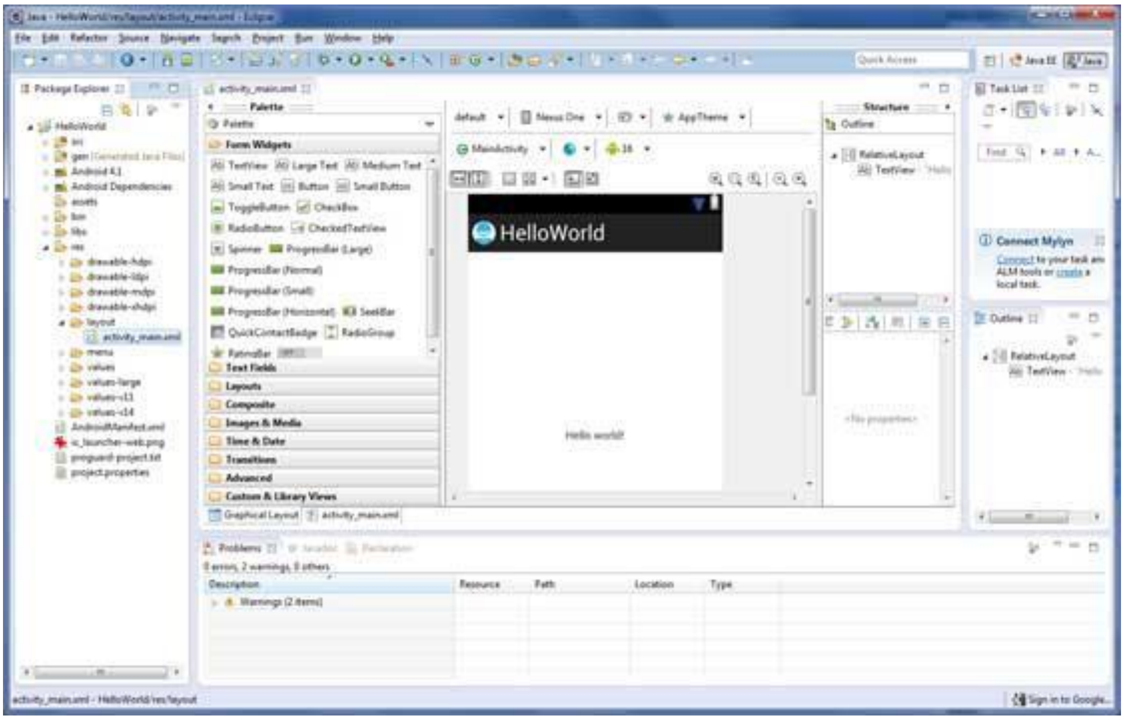
□ Mark this project as a library

☑ Create Project in Workspace

Location: C:\workspace\HelloWorld

The application name is shown in the Play Store, as well as in the Manage Application list in Settings.

Next, follow the instructions provided and keep all other entries as default till the final step. Once your project is created successfully, you will have following project screen:



The Main Activity File

The main activity code is a Java file MainActivity.java. This is the actual application file which ultimately gets converted to a Dalvik executable and runs your application. Following is the default code generated by the application wizard for Hello World! application:

```
package com.example.helloworld;
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.MenuItem;
import android.support.v4.app.NavUtils;
public class MainActivity extends Activity {
   @Override
   public void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_main);
   }
            @Override
            public boolean onCreateOptionsMenu(Menu menu) {
            getMenuInflater().inflate(R.menu.activity_main, menu);
             return true;
 }
 }
```

Here, R.layout.activity_main refers to the activity_main.xml file located in the res/layout folder. TheonCreate() method is one of many methods that are fired when an activity is loaded.

The Manifest File

Whatever component you develop as a part of your application, you must declare all its components in a manifest file called AndroidManifest.xml which ressides at the root of the application project directory. This file works as an interface between Android OS and your application, so if you do not declare your component in this file, then it will not be considered by the OS. For example, a default manifest file will look like as following file.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.example.helloworld"
  android:versionCode="1"
  android:versionName="1.0" >
  <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="15" />
<application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme">
        <activity
                android:name=".MainActivity"
                android:label="@string/title_activity_main">
                <intent-filter>
                        <action android:name="android.intent.action.MAIN"/>
                        <category android:name="android.intent.category.LAUNCHER"/>
                </intent-filter>
        </activity>
</application>
 </manifest>
```

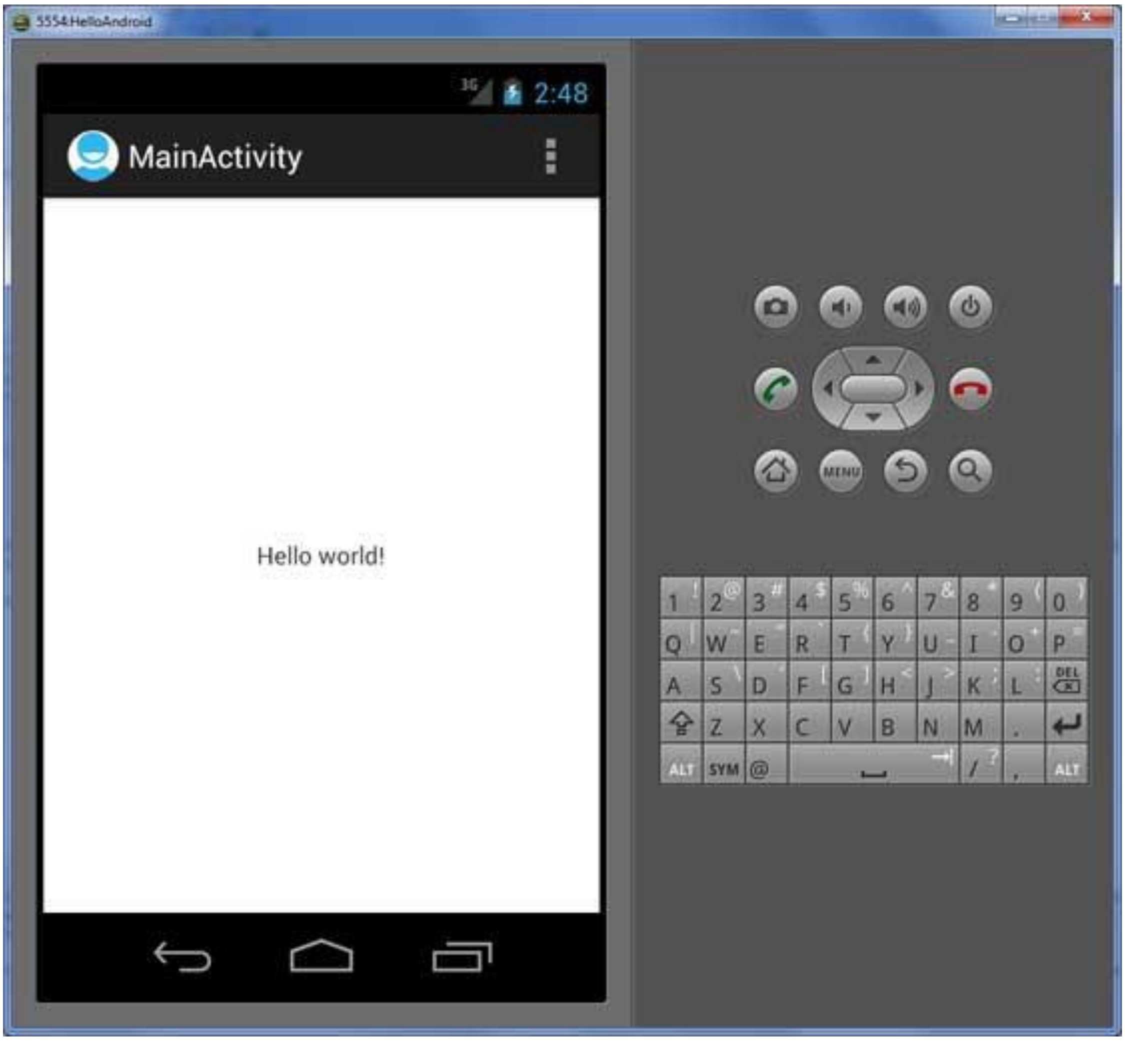
The Layout File

The activity_main.xml is a layout file available in res/layout directory, that is referenced by your application when building its interface. You will modify this file very frequently to change the layout of your application. For your "Hello World!" application, this file will have following content related to default layout:

```
<RelativeLayout  xmlns:android=http://schemas.android.com/apk/res/android
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent" >
        <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
         android:padding="@dimen/padding_medium"
        android:text="@string/hello_world" ]
        tools:context=".MainActivity" />
 </RelativeLayout>
```

Running the Application

Let's try to run our Hello World! application we just created. I assume you had created your AVD while doing environment setup. To run the app from Eclipse, open one of your project's activity files and click Run icon from the toolbar. Eclipse installs the app on your AVD and starts it and if everything is fine with your setup and application, it will display following Emulator window:



Lab 4: How to communicate among two applications.

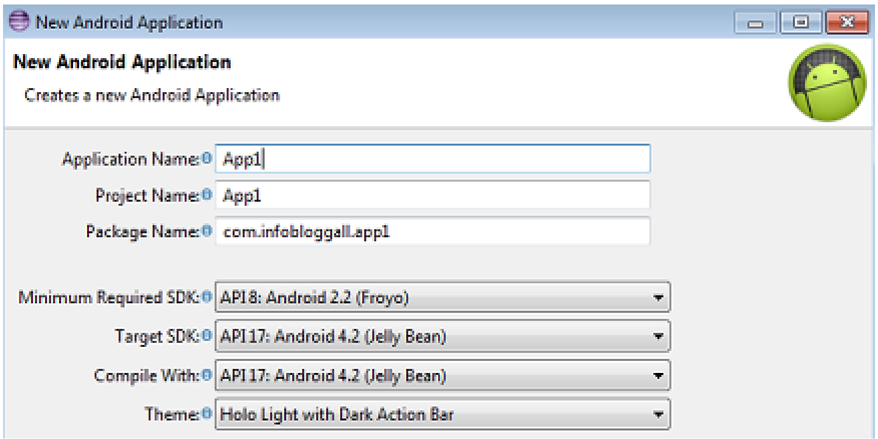
Solution:

Communication among two Applications

Consider two Applications: Application 1 and Application 2

- Creating Application 1 having services that is open to other application.
- Frame a messenger and allow it to work in onBind() method of service.
- Create handler so that it will allow message to be there on Application 2.
- Frame Application 2 by creating ServiceConnection object in order to open messenger.
- Fix exposed service to Application 2 and apply messenger in order to get messages to be send to Application 1.

Step 1: Create Application 1 with name App1.



New Android Application

New Android Application

Creates a new Android Application

Application Name: App1

Project Name: App1

Package Name: com.infobloggall.app1

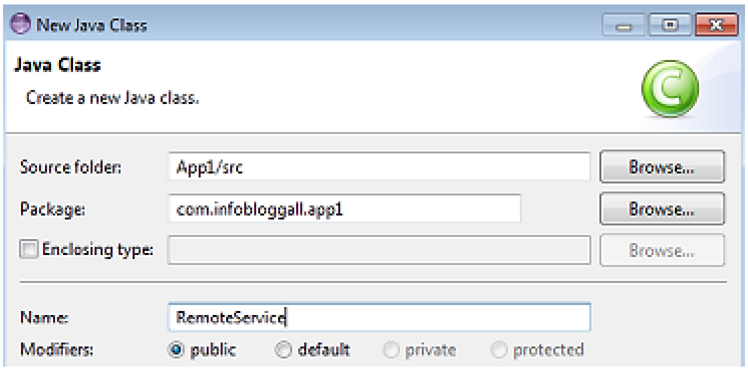
Minimum Required SDK: API 8: Android 2.2 (Froyo)

Target SDK: API 17: Android 4.2 (Jelly Bean)

Compile With: API 17: Android 4.2 (Jelly Bean)

Theme: Holo Light with Dark Action Bar

Step 2: Create class as RemoteService.java which acts as remote service that will open other applications.



New Java Class

Java Class

Create a new Java class.

Source folder: App1/src

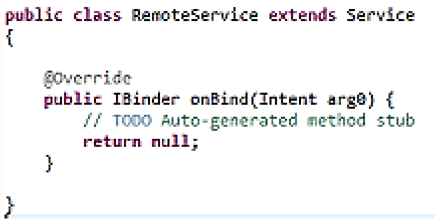
Package: com.infobloggall.app1

☐ Enclosing type:

Name: RemoteService

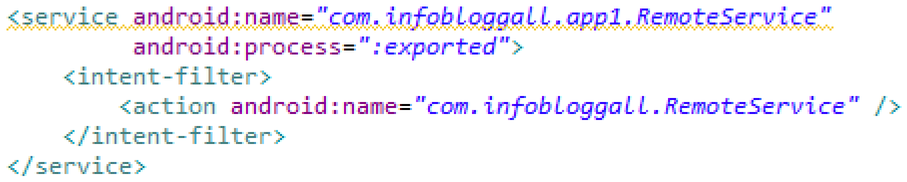
Modifiers: ● public ○ default ○ private ○ protected

Step 3: Now we have to extend service so that our class will have features of service and also implement the onBind () method.



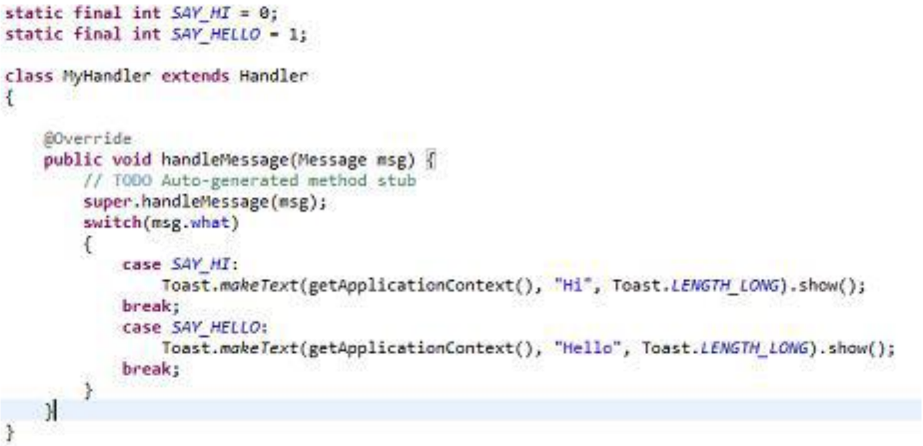
```
public class RemoteService extends Service
{

    @Override
    public IBinder onBind(Intent arg0) {
        // TODO Auto-generated method stub
        return null;
    }

}
```

Step 4: After now, to make such service remote, the RemoteService to other Applications or Process will exists. In Manifest file, name the service and make it remote so that application can use it. For this, export field that will allow many process to utilize service and explain intent filter along with action name in order to call the service from various applications.



```
<service android:name="com.infobloggall.app1.RemoteService"
         android:process=":exported">
    <intent-filter>
        <action android:name="com.infobloggall.RemoteService" />
    </intent-filter>
</service>
```

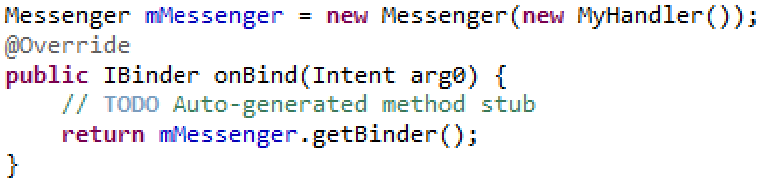
Step 5: Move to RemoteService.java class and create handler to take care of messages sent across from different other application that will be displayed in a toast. Also frame constants with regards to message in order to find the sent messages.



```
static final int SAY_HI = 0;
static final int SAY_HELLO = 1;

class MyHandler extends Handler
{

    @Override
    public void handleMessage(Message msg) {
        // TODO Auto-generated method stub
        super.handleMessage(msg);
        switch(msg.what)
        {
            case SAY_HI:
                Toast.makeText(getApplicationContext(), "Hi", Toast.LENGTH_LONG).show();
            break;
            case SAY_HELLO:
                Toast.makeText(getApplicationContext(), "Hello", Toast.LENGTH_LONG).show();
            break;
        }
    }
}
```

Fig 2.5 RemoteService.java class

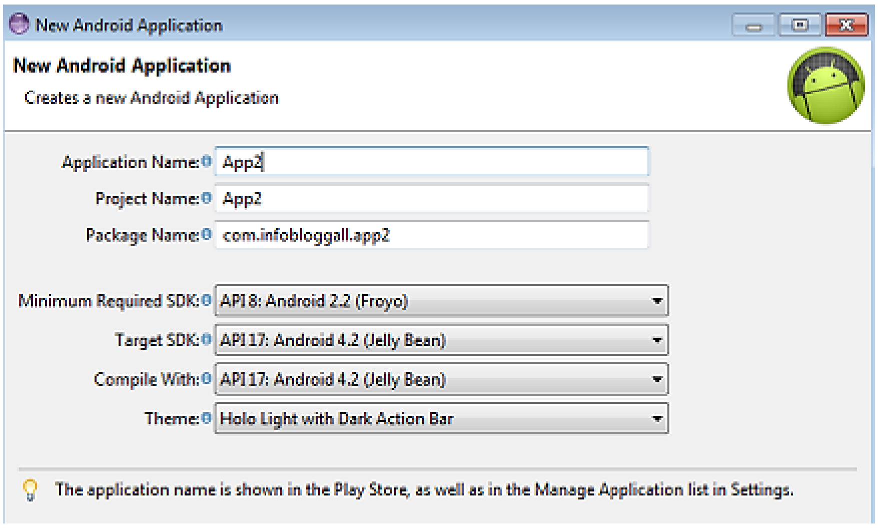
Step 6: Create the Messenger by passing the handler to messenger's constructor and attach the handler with messenger. Now in onBind() method, open the messenger so that service can send messages using open messenger.



```
Messenger mMessenger = new Messenger(new MyHandler());
@Override
public IBinder onBind(Intent arg0) {
    // TODO Auto-generated method stub
    return mMessenger.getBinder();
}
```

Fig 2.6 Code for messenger

Step 7: From this, we have completed with Application 1 and now start creation of Application 2.



New Android Application

New Android Application

Creates a new Android Application

Application Name: App2

Project Name: App2

Package Name: com.infobloggall.app2

Minimum Required SDK: API 8: Android 2.2 (Froyo)

Target SDK: API 17: Android 4.2 (Jelly Bean)

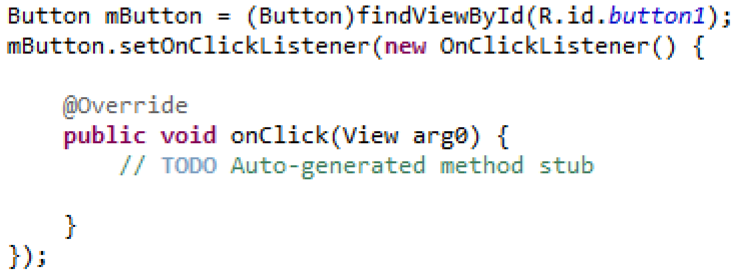
Compile With: API 17: Android 4.2 (Jelly Bean)

Theme: Holo Light with Dark Action Bar

The application name is shown in the Play Store, as well as in the Manage Application list in Settings.

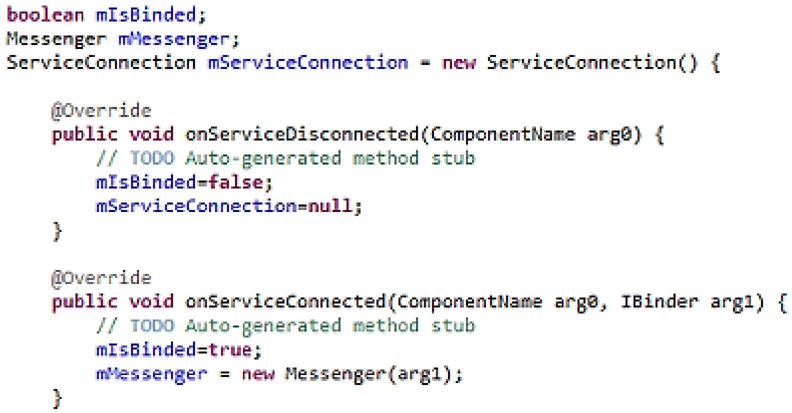
Fig 2.7 New Android application

Step 8: Here we will add a button to activity which on clicking, will send the message to Application 1.



```
Button mButton = (Button)findViewById(R.id.button1);
mButton.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View arg0) {
        // TODO Auto-generated method stub

    }
});
```

Step 9: No frame ServiceConnection object that fetch Messenger exposed by RemoteService. The arg1 variable is used to retrieve Messanger.



```
boolean mIsBinded;
Messenger mMessenger;
ServiceConnection mServiceConnection = new ServiceConnection() {

    @Override
    public void onServiceDisconnected(ComponentName arg0) {
        // TODO Auto-generated method stub
        mIsBinded=false;
        mServiceConnection=null;
    }

    @Override
    public void onServiceConnected(ComponentName arg0, IBinder arg1) {
        // TODO Auto-generated method stub
        mIsBinded=true;
        mMessenger = new Messenger(arg1);
    }
```

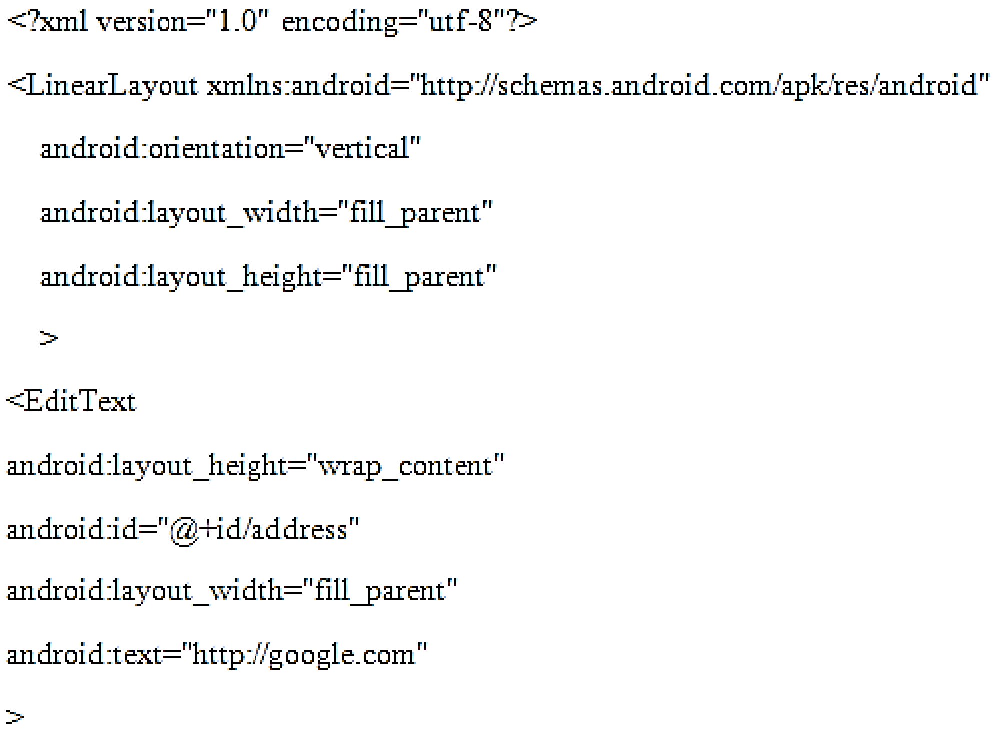
We have taken the messenger out and apply messenger to send message to App1 by clicking the button. Also, we make similar constants as what we did in App 1. On click of button, send() method will be called and send message to messenger of App 1.

Lab 5: To use networking tools in android.

Solution:

Android allows your application adheres to the internet or several other local network furthermore authorizes you to practice network actions. A device can hold different categories of network attachments. Before you practice some network actions, you need initial examine that are you joined to that network or internet e.t.c. For this android assigns ConnectivityManager class. You expect to instantiate an object of this class by designating getSystemService() method.

Once you instantiate the object of ConnectivityManager class, you can utilize getAllNetworkInfo approach to apprehend the details of complete the networks. After checking that you are connected to the internet, you can conduct some network activity. Android allows HttpURLConnection as well as URL class to handle these actions. You hope to instantiate an object of URL class by assigning the sequence of website. After that you expect to call openConnection mechanism of url class as well as accept it in a HttpURLConnection object. After that you need to call the connect approach of HttpURLConnection class.



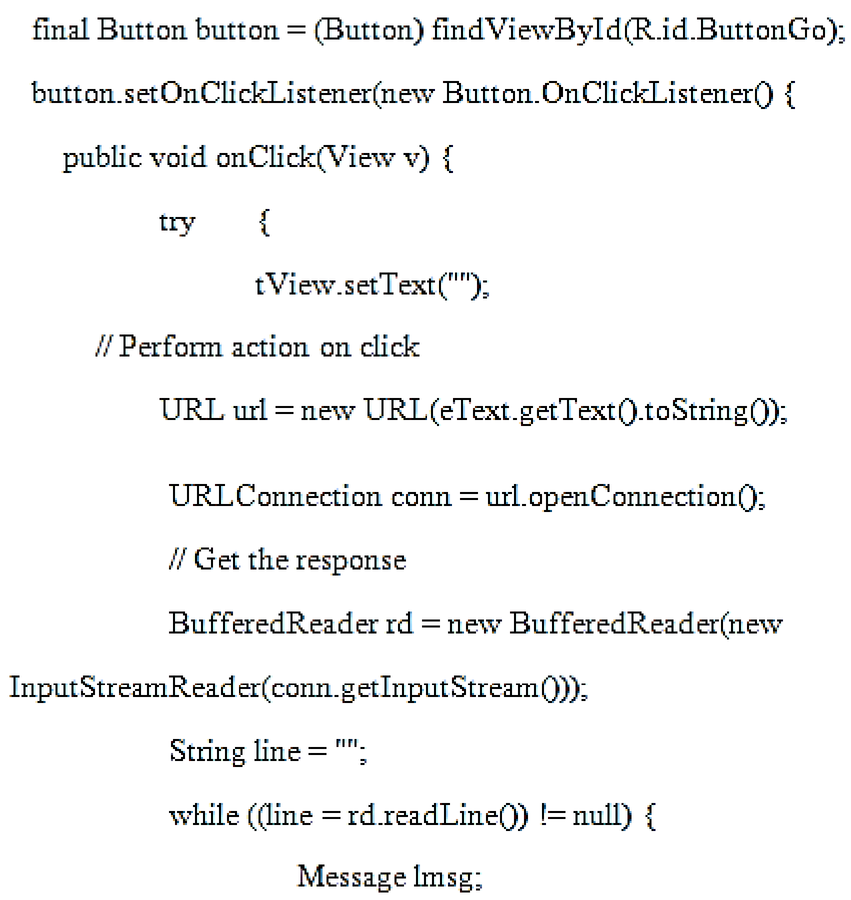
```
<?xml version="1.0"  encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:orientation="vertical"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   >
<EditText
android:layout_height="wrap_content"
android:id="@+id/address"
android:layout_width="fill_parent"
android:text="http://google.com"
>
```

```
</EditText>
<Button
 android:id="@+id/ButtonGo"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:text="go!"
 >
</Button>
<TextView
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   android:background="#ffffff"
   android:textColor="#000000"
         android:id="@+id/pagetext"
   />
</LinearLayout>
```

Listing 2 the program below describes Java code applied in above example.

```
package com.msi.getwebpage;
import android.app.Activity;
import android.os.Bundle;
// used for interacting with user interface
import android.widget.Button;
import android.widget.TextView;
import android.widget.EditText;
import android.view.View;
```

```
// used for passing data
import android.os.Handler;
import android.os.Message;
// used for connectivity
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.URL;
import java.net.URLConnection;
public class GetWebPage extends Activity {
 /** Called when the activity is first created. */
 Handler h;
     @Override
 public void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
setContentView(R.layout.main);
final EditText eText = (EditText) findViewById(R.id.address);
final TextView tView = (TextView) findViewById(R.id.pagetext);
this.h = new Handler() {
  @Override
  public void handleMessage(Message msg) {
    // process incoming messages here
    switch (msg.what) {
      case 0:
        tView.append((String) msg.obj);
        break;
    }
    super.handleMessage(msg);
  }
};
```



```
    final Button button = (Button) findViewById(R.id.ButtonGo);
    button.setOnClickListener(new Button.OnClickListener() {
        public void onClick(View v) {
                try       {
                        tView.setText("");
            // Perform action on click
                URL url = new URL(eText.getText().toString());
                 URLConnection conn = url.openConnection();
                 // Get the response
                 BufferedReader rd = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
                 String line = "";
                 while ((line = rd.readLine()) != null) {
                              Message lmsg;
                          Message lmsg;
                        lmsg = new Message();
                        lmsg.obj = line;
                        lmsg.what = 0;
                        GetWebPage.this.h.sendMessage(lmsg);
               }
               }
               catch (Exception e) {
               }
               }
          });
     }
}
```

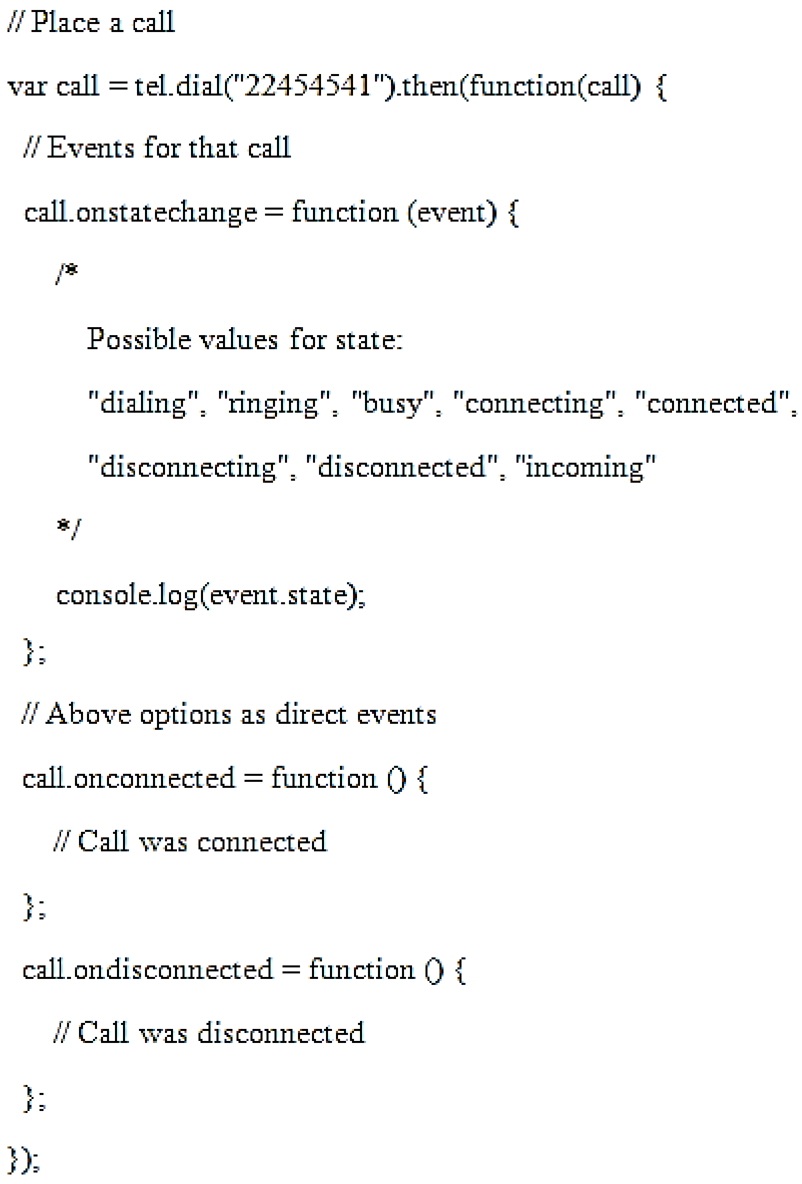
We see that code is distributed in general areas where many import statements will reference UI, data passing and networking classes. It is seen that every code with network is available in OnClick method of OnClickListener.

In this, URL and URLConnection classes team will show real connectivity to Web site of user's. Since every line is readable, text gets appended to TextView where data is assigned directly to TextView.

We see that Android application is working with HTTP Web server like Apache or Internet Information Server where an application is directly involved to TCP socket rather than HTTP.

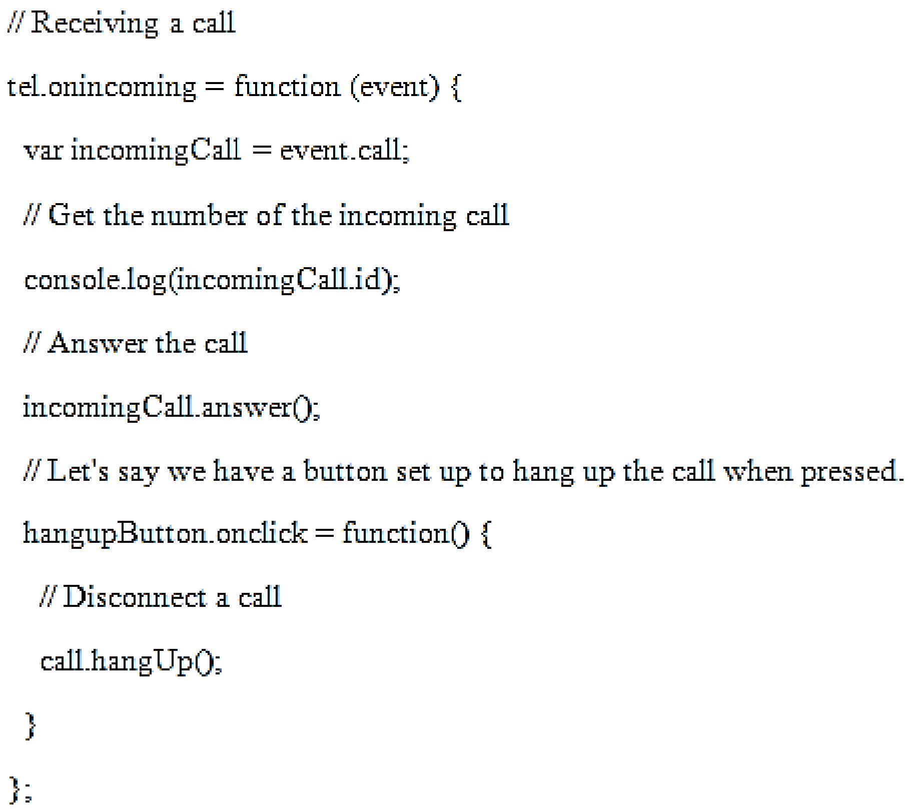
Lab 6: Write a program to describe placing a telephone cell.

Solution:



```
// Place a call
var call = tel.dial("22454541").then(function(call) {
  // Events for that call
  call.onstatechange = function (event) {
     /*
        Possible values for state:
        "dialing", "ringing", "busy", "connecting", "connected",
        "disconnecting", "disconnected", "incoming"
     */
     console.log(event.state);
  };
  // Above options as direct events
  call.onconnected = function () {
     // Call was connected
  };
  call.ondisconnected = function () {
     // Call was disconnected
  };
});
```

Apart from calling, we see that receiving of calls is entirely different where we have to write Telephony.onincoming event listener which ask when call is incoming and has event object such as CallEvent having call property which gives access to call's TelephonyCall object and does work as answering the call etc. The program below describes the code for receiving of call:



```
// Receiving a call
tel.onincoming = function (event) {
  var incomingCall = event.call;
  // Get the number of the incoming call
  console.log(incomingCall.id);
  // Answer the call
  incomingCall.answer();
  // Let's say we have a button set up to hang up the call when pressed.
  hangupButton.onclick = function() {
    // Disconnect a call
    call.hangUp();
  }
};
```

Lab 7: Write a program for android sensor API

Solution:

It is achieved by calling the method getSystemService () and passing the SENSOR_SERVICE constant in it.

SensorManager sm = (SensorManager) getSystemService (SENSOR_SERVICE);

2) Sensor class: The android.hardware.Sensor class will show methods to receive information about sensors which could be sensor name, sensor type, sensor resolution, sensor type etc.

3) SensorEvent class: Its instance is created by the system. It provides information about the sensor.

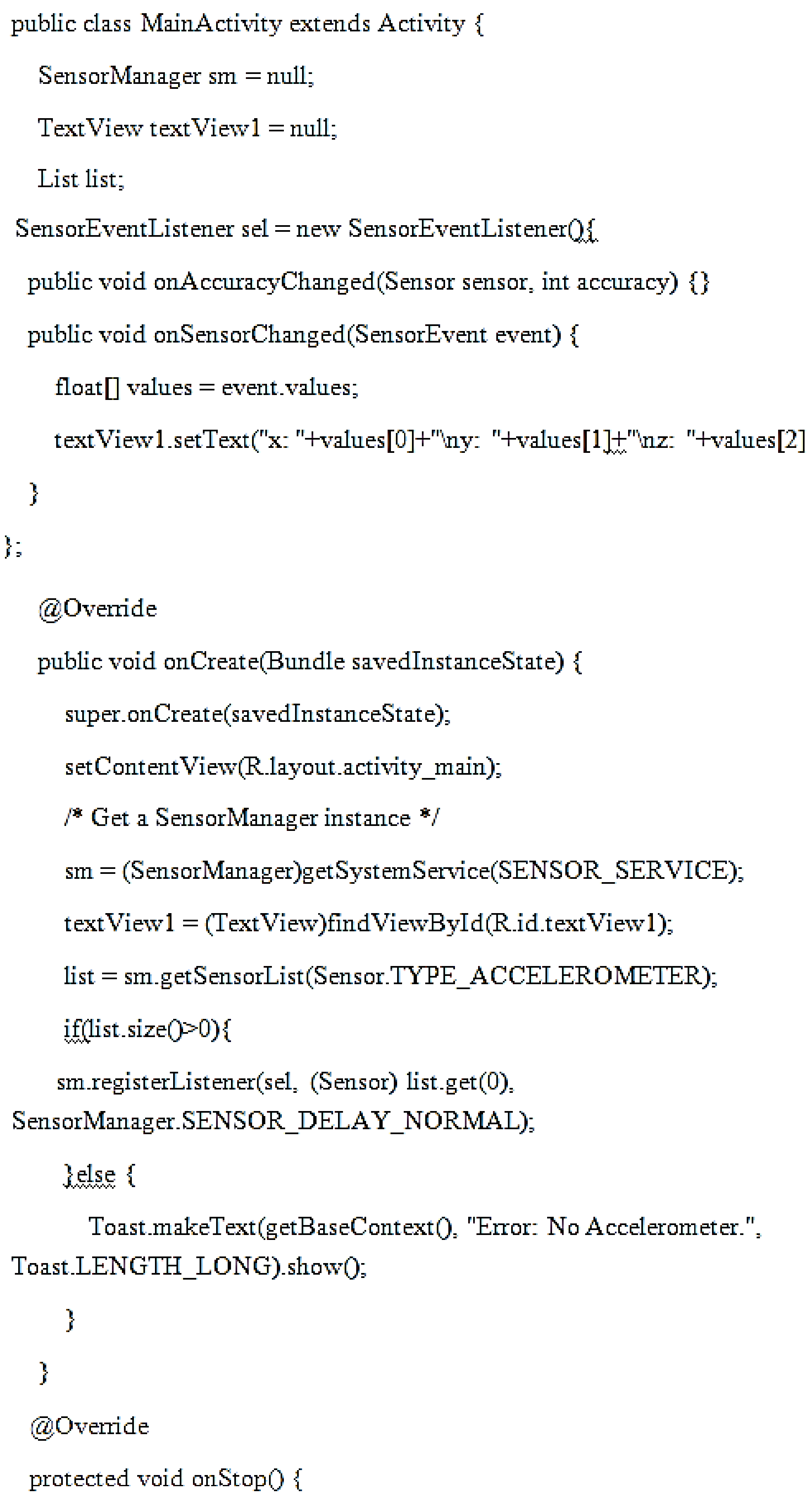
4) SensorEventListener interface: It delivers dual call back methods in order to have information about sensor regarding sensor values (x,y and z) change or sensor accuracy changes.

File: activity_main.xml

```
<RelativeLayout xmlns:androclass="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity" >
<TextView
 android:id="@+id/textView1"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:layout_alignParentLeft="true"
 android:layout_alignParentTop="true"
 android:layout_marginLeft="92dp"
 android:layout_marginTop="114dp"
 android:text="TextView" />
</RelativeLayout>
```

Activity class: Consider a code which gives values of x axis, y axis and z axis.

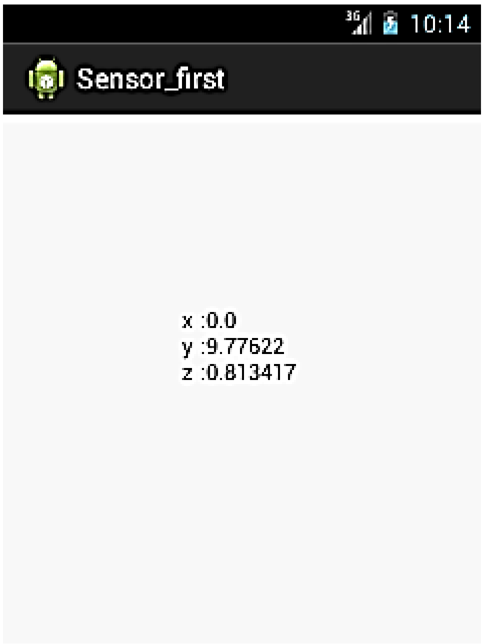
```
 package com.example.sensorsimple;
 import android.app.Activity;
 import android.os.Bundle;
 import android.widget.TextView;
 import android.widget.Toast;
 import android.hardware.SensorManager;
 import android.hardware.SensorEventListener;
 import android.hardware.SensorEvent;
 import android.hardware.Sensor;
 import java.util.List;
```



```
public class MainActivity extends Activity {
    SensorManager sm = null;
    TextView textView1 = null;
    List list;
 SensorEventListener sel = new SensorEventListener(){
   public void onAccuracyChanged(Sensor sensor, int accuracy) {}
   public void onSensorChanged(SensorEvent event) {
       float[] values = event.values;
       textView1.setText("x: "+values[0]+"\ny: "+values[1]+"\nz: "+values[2]
   }
};
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        /* Get a SensorManager instance */
        sm = (SensorManager)getSystemService(SENSOR_SERVICE);
        textView1 = (TextView)findViewById(R.id.textView1);
        list = sm.getSensorList(Sensor.TYPE_ACCELEROMETER);
        if(list.size()>0){
       sm.registerListener(sel, (Sensor) list.get(0),
SensorManager.SENSOR_DELAY_NORMAL);
        }else {
            Toast.makeText(getBaseContext(), "Error: No Accelerometer.",
Toast.LENGTH_LONG).show();
        }
    }
   @Override
   protected void onStop() {
```

```
            if(list.size()>0){
              sm.unregisterListener(sel);
            }
            super.onStop();
        }
    }
```

Output:



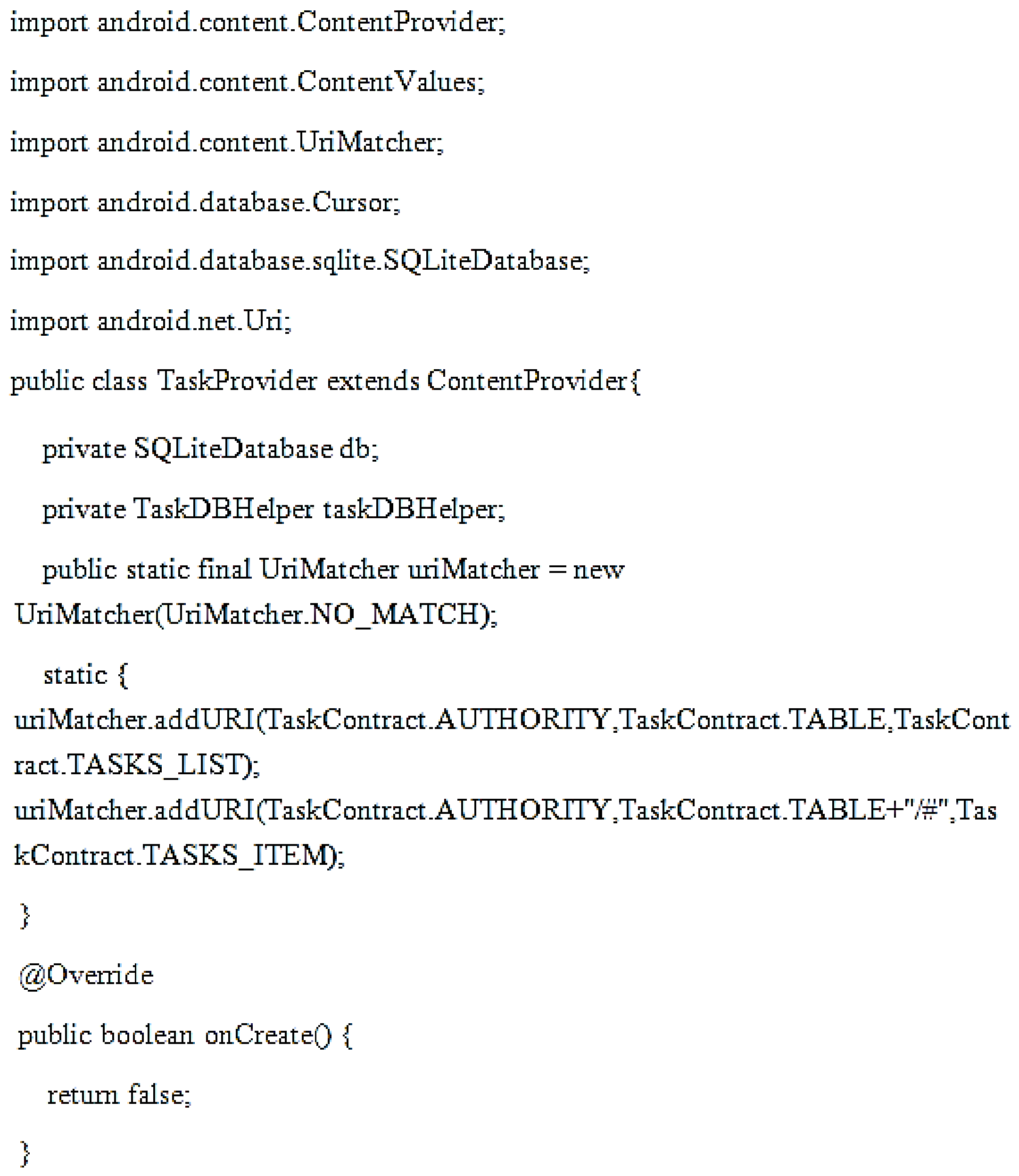
```
x :0.0
y :9.77622
z :0.813417
```

Lab 8: Write a program to creating a class ContentProvider.

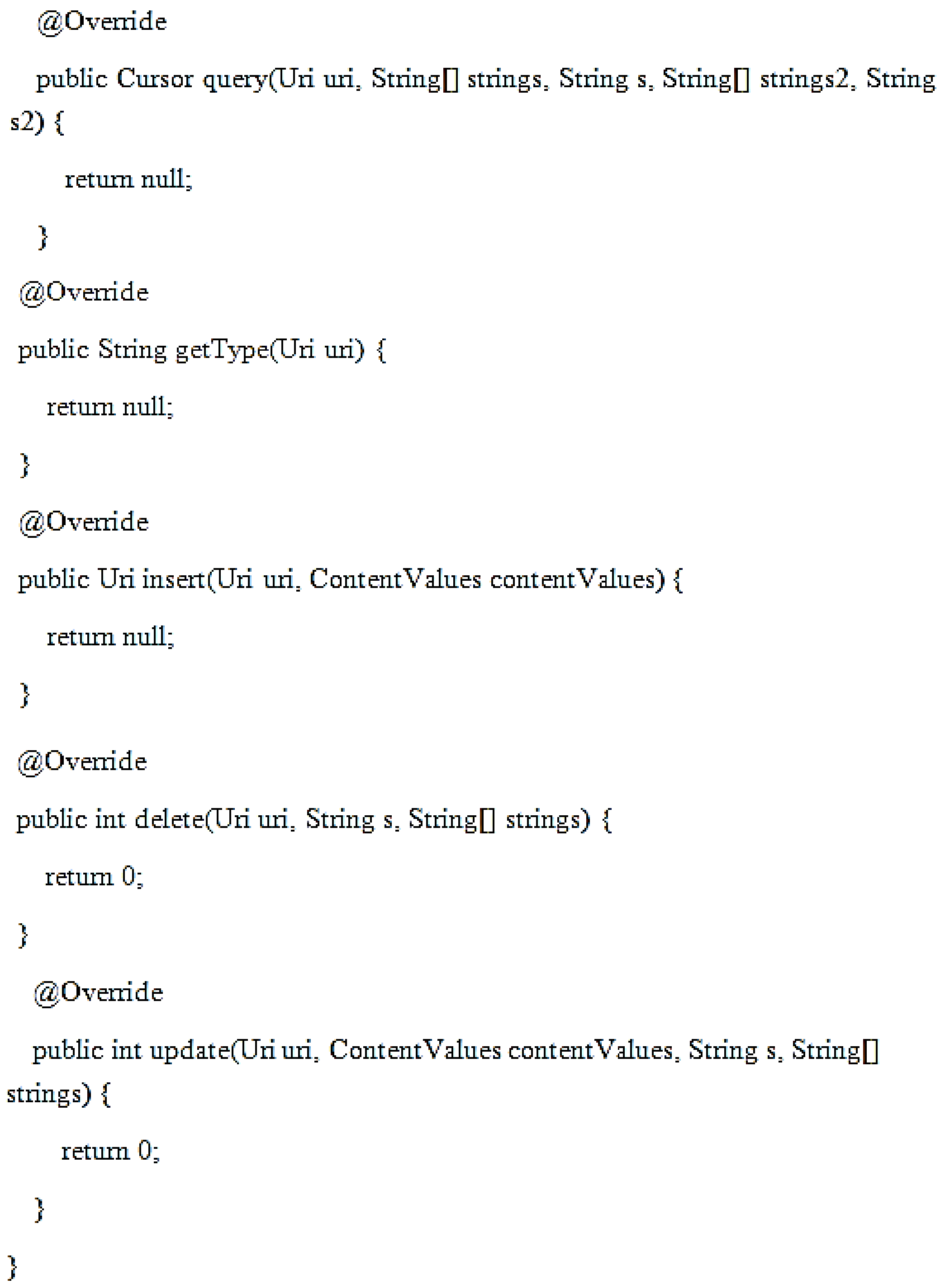
Solution:

It is noted that content provider serves as Java class that widen ContentProvider class and implements its methods. It is called as TaskProvider, which places in db package. For creating new class file in TaskProvider.java you have to use the following code:

package com.example.TodoList.db;



```
import android.content.ContentProvider;
import android.content.ContentValues;
import android.content.UriMatcher;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.net.Uri;
public class TaskProvider extends ContentProvider{
    private SQLiteDatabase db;
    private TaskDBHelper taskDBHelper;
    public static final UriMatcher uriMatcher = new
UriMatcher(UriMatcher.NO_MATCH);
    static {
uriMatcher.addURI(TaskContract.AUTHORITY,TaskContract.TABLE,TaskCont
ract.TASKS_LIST);
uriMatcher.addURI(TaskContract.AUTHORITY,TaskContract.TABLE+"/#",Tas
kContract.TASKS_ITEM);
 }
 @Override
 public boolean onCreate() {
     return false;
 }
```



```
   @Override
   public Cursor query(Uri uri, String[] strings, String s, String[] strings2, String
s2) {
       return null;
   }
 @Override
 public String getType(Uri uri) {
     return null;
 }
 @Override
 public Uri insert(Uri uri, ContentValues contentValues) {
     return null;
 }
 @Override
 public int delete(Uri uri, String s, String[] strings) {
     return 0;
 }
   @Override
   public int update(Uri uri, ContentValues contentValues, String s, String[]
strings) {
       return 0;
   }
}
```

In this, we see that such class will not do much. Since it is a content provider, but it doesn’t provide any content because of non-execution of methods.