

MSBTE SAMPLE QUESTION PAPER WITH ANSWERS

Q.1) Attempt any FIVE of the following.

(10 Marks)

a) List features of Android Operating System.

Ans:

Features of Android:

Android is a powerful open source operating system which provides a lot of great features, those are

1. It's open source and we can customize the OS based on our requirements.
2. It supports connectivity for GSM, CDMA, WIFI, NFC, Bluetooth, etc. for telephony or data transfer. It will allow us to make or receive a calls / SMS messages and we can send or retrieve a data across mobile networks
3. By using WIFI technology we can pair with other devices using apps
5. We can perform all data storage related activities by using light weight database SQLite.

b) Define Android Virtual Devices (AVD).

An Android Virtual Device (AVD) is a device configuration that is run with the Android emulator. It works with the emulator to provide a virtual device-specific environment in which to install and run Android apps.

c) Write the directory path where images are stored while developing Android Application.

Ans: When you need to display static images in your app, you can use the Drawable class and its subclasses to draw shapes and images.

Path for static images:

<Android Project>

AndroidManifest.xml

java.<package>

ActivityMain.java

<res>

/drawable > <images>

/layout

/mipmap

/values

d) List all attributes to develop a simple button .

Ans:

To display a button in an activity, add a button to the activity's layout XML file:

```
<Button
    android:id="@+id/button_id"
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:text="Submit" />
```

e) Write the syntax for Intent-Filter tag.

Ans:

Syntax for <intent-filter> tag:

```
<intent-filter android:icon="drawable resource"
    android:label="string resource"
    android:priority="integer" >

    . . .
</intent-filter>
```

f) Define services in Android operating system.

Ans: Android service is a component that is used to perform operations on the background such as playing music, handling network transactions, interacting content providers etc. It doesn't have any UI (user interface). The service runs in the background indefinitely even if application is destroyed.

g) Enlist the steps to publish the Android application.

- Step 1: Sign up. Sign up for an account on the Android Developer Console.
- Step 2: Create a new application.
- Step 3: Prepare multimedia.
- Step 4: Prepare code for release.
- Step 5: Build a release-ready APK.
- Step 6: Upload APK.

Q.2) Attempt any THREE of the following.

(12 Marks)

a) Describe the Android architecture in detail.

Ans:

Android architecture is a software stack of components to support a mobile device needs. Android software stack contains a Linux Kernel, a collection of c/c++ libraries which is exposed through an application framework, services, runtime and application.

Following are main components of android architecture those are

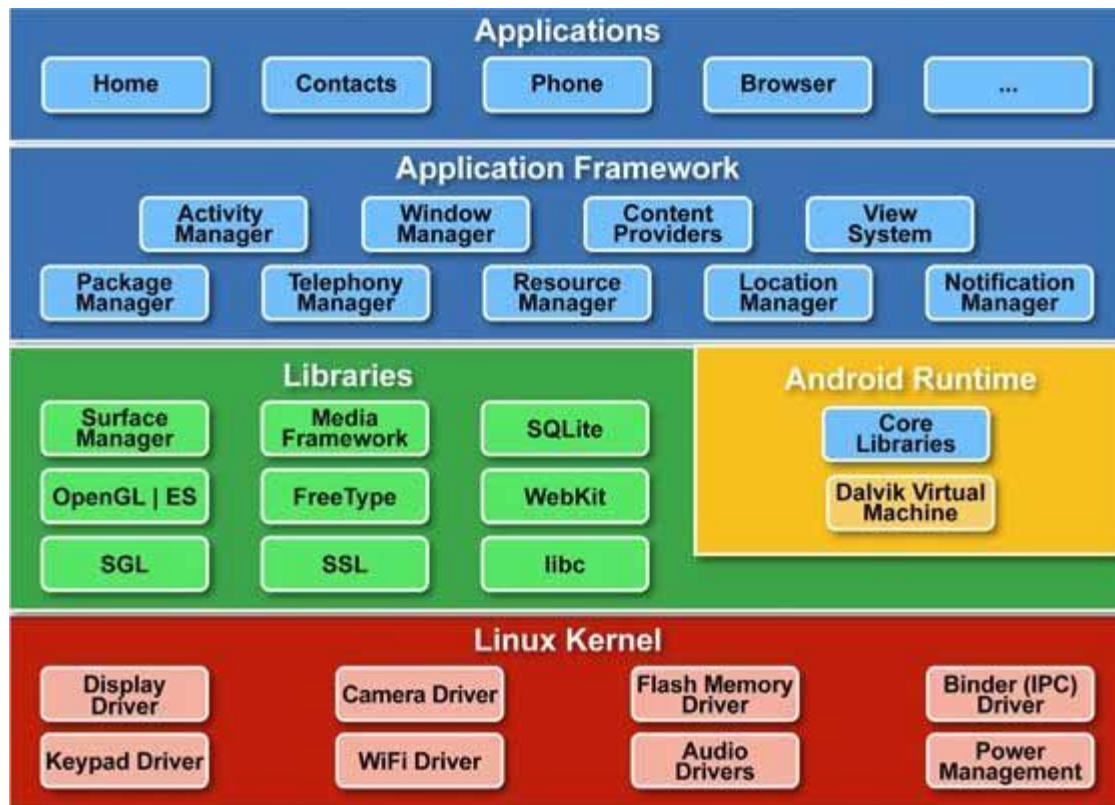


Fig: Components of Android Architecture

1) Linux kernel

It is the heart of android architecture that exists at the root of android architecture. Linux kernel is responsible for device drivers, power management, memory management, device management and resource access.

2) Native Libraries

On top of the linux kernel, there are Native libraries such as WebKit, OpenGL, FreeType, SQLite, Media, C runtime library (libc) etc.

The WebKit library is responsible for browser support, SQLite is for database, FreeType for font support, Media for playing and recording audio and video formats.

3) Android Runtime

In android runtime, there are core libraries and DVM (Dalvik Virtual Machine) which is responsible for running android applications. DVM is like JVM but it is

optimized for mobile devices. It consumes less memory and provides fast performance.

4) Android Framework

On the top of Native libraries and android runtime, there is an android framework. Android framework includes Android APIs such as UI (User Interface), telephony, resources, locations, Content Providers (data) and package managers. It provides a lot of classes and interfaces for android application development.

5) Applications

On the top of the android framework, there are applications. All applications such as home, contact, settings, games, browsers are using android framework that uses android runtime and libraries. Android runtime and native libraries are using linux kernel.

b) Differentiate between JVM and DVM.

Ans:

Java Virtual Machine	Dalvik Virtual Machine
1. It is stack based.	It is registered based which is designed to run on low memory.
2. JVM uses java bytecode	DVM uses its own bytecode
3. It runs “.class” file	It runs the “.dex” file.
4. Single instance shared with multiple applications.	DVM has been designed so that a device can run multiple instances of the VM efficiently.
5. JVM supports multiple operating systems.	DVM supports Android operating systems only.
6. For JVM many re-tools are available.	For DVM very few Re-tools are available.

7. It has a constant pool for every class.	There is a constant pool for every application.
8. Executable file is JAR file	Here the executable is APK.

c) Explain the activity life cycle.

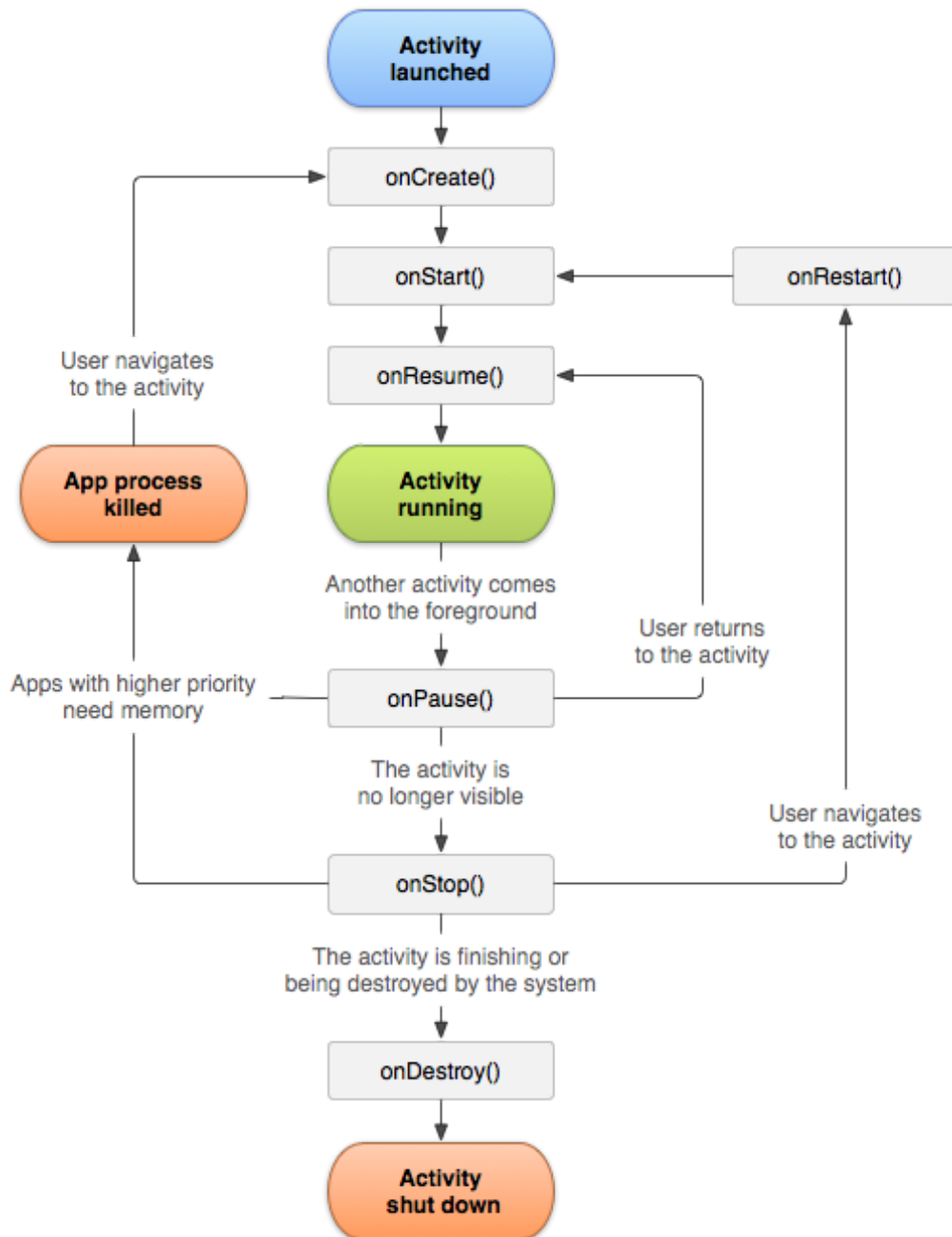


Fig: Activity Life-Cycle

In android, Activity class have 7 callback methods like onCreate(), onStart(), onPause(), onRestart(), onResume(), onStop() and onDestroy() to describe how the activity will behave at different stages.

By using activity call-back methods we can define how our activity can behave when the user enters or leaves our application.

1. onCreate()

This is the first callback method and it fires when the system creates an activity for the first time. During the activity creation, activity entered into a Created state.

2. onStart()

The onStart() callback method will invoke when an activity enters into Started State by completing onCreate() method. The onStart() method will make an activity visible to the user and this method execution will finish very quickly.

3. onResume()

When an activity enters into a Resume state, the system invokes onResume() call back method. In this state activity starts interacting with the user that means the user can see the functionality and designing part of an application on the single screen.

4. onPause()

Whenever the user leaves an activity or the current activity is being Paused then the system invokes onPause() method. The onPause() method is used to pause operations like stop playing the music when the activity is in a paused state or pass an activity while switching from one app to another app because every time only one app can be focused.

5. onStop()

The system will invoke onStop() callback method when an activity no longer visible to the user, the activity will enter into Stopped state. This happens due to current activity entered into Resumed state or newly launched activity covers complete screen or it's been destroyed.

6. onRestart()

The system will invoke onRestart() method when an activity restarts itself after stopping it. The onRestart() method will restore the state of activity from the time that is being stopped.

7. onDestroy()

The system will invoke onDestroy() method before an activity is destroyed and this is the final callback method received by the android activity.

Whenever the user is trying to leave an activity like switching from one app to another app, the system will use callback methods to dismantle the activity completely or partially to resume the activity from where the user left off. Based on our requirements we can implement the activity in the android app using the callback method and it's not necessary to use all callback methods in each android application

d) Discuss the need of permissions in Android. Describe the permissions to set system functionalities like bluetooth, camera.

1. The purpose of a permission is to protect the privacy of an Android user. Android apps must request permission to access sensitive user data (such as contacts and SMS), as well as certain system features (such as camera and internet). Depending on the feature, the system might grant the permission automatically or might prompt the user to approve the request.
2. A central design point of the Android security architecture is that no app, by default, has permission to perform any operations that would adversely impact other apps, the operating system, or the user. This includes reading or writing the user's private data (such as contacts or emails), reading or writing another app's files, performing network access, keeping the device awake, and so on.
3. An app must publicize the permissions it requires by including `<uses-permission>` tags in the app manifest. For example, an app that needs to send SMS messages would have this line in the manifest:
Example: `<uses-permission android:name="android.permission.SEND_SMS"/>`
4. Access to some hardware features (such as Bluetooth or the camera) require an app permission. However, not all Android devices actually have these hardware features. So if your app requests the CAMERA permission, it's important that you also include the `<uses-feature>` tag in your manifest to declare whether or not this feature is actually required.
For example:
`<uses-feature android:name = "android.hardware.camera" android:required = "false" />`
5. If you declare `android:required="false"` for the feature, then Google Play allows your app to be installed on devices that don't have the feature. You then must check if the current device has the feature at runtime by calling `PackageManager.hasSystemFeature()`, and gracefully disable that feature if it's not available.
6. If you don't provide the `<uses-feature>` tag, then when Google Play sees that your app requests the corresponding permission, it assumes your app requires this feature. So it filters your app from devices without the feature, as if you declared `android:required="true"` in the `<uses-feature>` tag.

Q.3) Attempt any THREE of the following. (12 Marks)

a) Explore the Steps to install and configure Android Studio and SDK.

Ans:

Step I Install the Java SE Development Kit

Step II Install and Set Up Android Studio

1. Install Android Studio
2. Setting up Android Studio takes just a few clicks.
3. First, be sure you download the latest version of Android Studio.
4. To install Android Studio on Windows, proceed as follows
5. If you downloaded an .exe file (recommended), double-click to launch it.
6. If you downloaded a .zip file, unpack the ZIP, copy the android-studio folder into your Program Files folder, and then open the android-studio > bin folder and launch studio64.exe (for 64-bit machines) or studio.exe (for 32-bit machines).
7. Follow the setup wizard in Android Studio and install any SDK packages that it recommends.

Step III Installing Android SDK You can install Android SDK platform packages and tools from Android Studio.

Install Android SDK Platform Packages and Tools

1. Start Android Studio.
2. To open SDK Manager, do any of these: On Android Studio landing page, select Configure > SDK Manager.
3. In the Default Settings dialog box, click these tabs to install Android SDK platform packages and developer tools. SDK Platforms: Select the latest Android SDK package.
4. Click Apply.
5. Click OK.

b) Explain Date and Time picker with its methods.

Ans: DatePickerDialog and TimePickerDialog classes have **onDateSetListener()** and **onTimeSetListener()** callback methods respectively. These callback methods are invoked when the user is done with filling the date and time respectively.

The DatePickerDialog class consists of a 5 argument constructor with the parameters listed below.

1. Context: It requires the application context
2. Callback Function: onDateSet() is invoked when the user sets the date with the following parameters:
 - int year : It will be store the current selected year from the dialog
 - int monthOfYear : It will be store the current selected month from the dialog
 - int dayOfMonth : It will be store the current selected day from the dialog
3. int mYear : It shows the the current year that's visible when the dialog pops up

4. `int mMonth` : It shows the the current month that's visible when the dialog pops up
 5. `int mDay` : It shows the the current day that's visible when the dialog pops up
- The `TimePickerDialog` class consists of a 5 argument constructor with the parameters listed below.

1. `Context`: It requires the application context
2. `CallBack Function`: `onTimeSet()` is invoked when the user sets the time with the following parameters:
 - `int hourOfDay` : It will be store the current selected hour of the day from the dialog
 - `int minute` : It will be store the current selected minute from the dialog
3. `int mHours` : It shows the the current Hour that's visible when the dialog pops up
4. `int mMinute` : It shows the the current minute that's visible when the dialog pops up
5. `boolean false` : If it's set to false it will show the time in 24 hour format else not

c) Describe the significance of SQLite database in Android.

Ans: SQLite is a software library that provides a relational database management system. The lite in SQLite means light weight in terms of setup, database administration, and required resource.

SQLite has the following significance: self-contained, serverless, zero-configuration, transactional, SQLite has distinctive features.

1. **Self-Contained:** SQLite is self-contained means it requires minimal support from the operating system or external library. This makes SQLite usable in any environment, especially in embedded devices like iPhones, Android phones, game consoles, handheld media players, etc.
2. **Serverless:** SQLite does NOT require a server to run.
SQLite database is integrated with the application that accesses the database. The applications interact with the SQLite database read and write directly from the database files stored on disk.
3. **Zero-configuration:** Because of the serverless architecture, you don't need to "install" SQLite before using it. There is no server process that needs to be configured, started, and stopped. In addition, SQLite does not use any configuration files.
4. **Transactional:** All transactions in SQLite are fully ACID-compliant. It means all queries and changes are Atomic, Consistent, Isolated, and Durable.
5. **SQLite has distinctive features:**
 - SQLite uses dynamic types for tables. It means you can store any value in any column, regardless of the data type.

- SQLite allows a single database connection to access multiple database files simultaneously. This brings many nice features like joining tables in different databases or copying data between databases in a single command.
- SQLite is capable of creating in-memory databases which are very fast to work with.

d) Discuss Developer console with its purpose.

Developer Console

The main purpose of Google Play Developer Console is for publishing operations and tools. Upload apps, build your product pages, configure prices and distribution, and publish. You can manage all phases of publishing on Google Play through the Developer Console, from any web browser.

Once you've registered and received verification by email, you can sign in to your Google Play Developer Console.

The purpose of Developer Console is:

1. List All Applications

Start in All Applications, which gives you a quick overview of your apps, lets you jump to stats, reviews, and product details, or upload a new app.

2. Fill and manage Your Account Details

Specify basic developer profile information about yourself or your company on the accounts detail page. This identifies you to Google Play and your customers.

3. Linking Your Merchant Account

If you want to sell apps or in-app products, link your Google Wallet Merchant Account to your developer profile.

4. Manage Multiple User Accounts

Set up user accounts for other team members to access different parts of your Developer Console.

5. Store Listing Details

Use the Developer Console to set up a Store Listing page. This is the home for your app in Google Play.

6. Upload and Instantly Publish

From the Developer Console you can quickly upload and publish a release-ready Android application package file.

7. Alpha and Beta Testing

It's always valuable to get real-world feedback from users, especially before launch.

8. Staged Rollouts

You can also stage the rollout of your apps using the Production tab in the APK section of your Google Play Developer Console. Here you can define the percentage of users who'll be able to download your app.

9. Multiple APK Support

In most cases, a single app package (APK) is all you need, and it's usually the easiest way to manage and maintain the app. However, if you need to deliver a different APK to different devices, Google Play provides a way to do that.

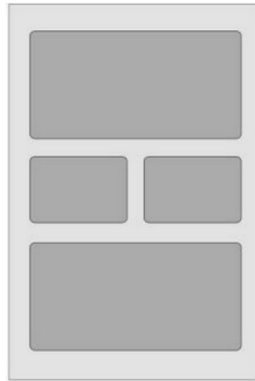
10. App Statistics

You get detailed statistics on the install performance of your app.

Q.4) Attempt any THREE of the following.

(12 Marks)

a) Observe the following GUI and write an XML file using relative layout to create the Same.



Ans:

The .xml file code for displaying shown image layout is:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="10sp"
    android:layout_marginTop="60sp">
    <Button
        android:id="@+id/btn1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:minHeight="250sp"
        android:layout_alignParentLeft="true"
        android:text="Button1" />
    <Button
        android:id="@+id/btn2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:width="200sp"
        android:minHeight="150sp"
```

```

        android:layout_below="@+id/btn1"
        android:text="Button2" />
<Button
    android:id="@+id/btn3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/btn1"
    android:minHeight="150sp"
    android:width="200sp"
    android:layout_toRightOf="@+id/btn2"
    android:text="Button3" />
<Button
    android:id="@+id/btn4"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/btn2"
    android:minHeight="200sp"
    android:text="Button4" />
</RelativeLayout>

```

b) Write a program to display a circular progress bar.

Ans: By default the ProgressBar will be displayed as a spinning wheel [Circular progress bar]

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="10sp"
    android:layout_marginTop="60sp">
    <ProgressBar
        android:id="@+id/pBar3"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:minHeight="50dp"
        android:minWidth="250dp"
        android:max="100"
        android:indeterminate="true"
        android:progress="1" />
</RelativeLayout>

```

android:id	It is used to uniquely identify the control
android:minHeight	It is used to set the height of the progress bar.
android:minWidth	It is used to set the width of the progress bar.
android:max	It is used to set the maximum value of the progress bar.
android:progress	It is used to set the default progress value between 0 and max. It must be an integer value.
android:indeterminate	we use the Indeterminate progress mode in progress bar when we don't know how long an operation will take or how much work has been done.

c) List sensors in Android and explain any one in detail.

Ans: The Android platform supports three broad categories of sensors:

1. Motion sensors
2. Environmental sensors
3. Position sensors

Motion Sensors:

1. The Android platform provides several sensors that let you monitor the motion of a device.
The sensors' possible architectures vary by sensor type:
 - The gravity, linear acceleration, rotation vector, significant motion, step counter, and step detector sensors are either hardware-based or software-based.
 - The accelerometer and gyroscope sensors are always hardware-based.
2. Motion sensors are useful for monitoring device movement, such as tilt, shake, rotation, or swing.
3. The movement is usually a reflection of direct user input (for example, a user steering a car in a game or a user controlling a ball in a game), but it can also be a reflection of the physical environment in which the device is sitting (for example, moving with you while you drive your car).
4. In the first case, you are monitoring motion relative to the device's frame of reference or your application's frame of reference; in the second case you are monitoring motion relative to the world's frame of reference.
5. Motion sensors by themselves are not typically used to monitor device position, but they can be used with other sensors, such as the geomagnetic field sensor, to determine a device's position relative to the world's frame of reference.

6. All of the motion sensors return multi-dimensional arrays of sensor values for each SensorEvent. For example, during a single sensor event the accelerometer returns acceleration force data for the three coordinate axes, and the gyroscope returns rate of rotation data for the three coordinate axes. These data values are returned in a float array (values) along with other SensorEvent parameters.

d) Explain zoom control (IN / OUT) with the help of an example.

Ans: In Android, Zoom Controls class displays a simple set of controls that is used for zooming and provides a callback to register for events. Zoom Controls has two buttons ZoomIn and ZoomOut which are used to control the zooming functionality.

Important Methods Of Zoom Controls:

1. hide(): This method is used to hide the ZoomControls from the screen. In some cases we need to hide the ZoomControls from the screen so that we use this function.
2. show(): This method is used to show the ZoomControls which we hide from the screen by using the hide method.
3. setOnZoomInClickListener(OnClickListenerlistener): This is a listener event automatically called when we click on the Zoom In button of ZoomControls. In this listener we add the code to zoom in an image.
4. setOnZoomOutClickListener(OnClickListenerlistener): This is a listener event automatically called when we click on the Zoom Out button of ZoomControls. In this listener we add the code for zooming out an image.
5. setIsZoomInEnabled(boolean isEnabled): This method is used to enable or disable the zoom In button of ZoomControls.
6. setIsZoomOutEnabled(boolean isEnabled): This method is used to enable or disable the zoom Out button of ZoomControls.

Zoom Controls example code in XML:

```
<ZoomControls
    android:id="@+id/simpleZoomControl"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

e) Develop an application to send and receive SMS. (Write ONLY .java and permission tag in manifest file)

Ans:

Step I: Permission for SMS SEND and RECEIVE:

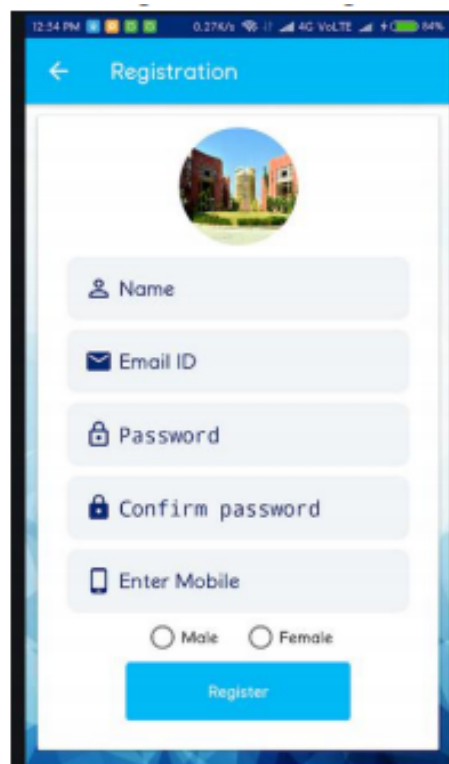
```
<uses-permission android:name="android.permission.SEND_SMS"/>
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
```

Step II: Code for MainActivity.java file --->

```

public class MainActivity extends Activity {
    EditText mobileno, message;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mobileno = (EditText) findViewById(R.id.editText1);
        message = (EditText) findViewById(R.id.editText2);
    }
    public void sendSMS(View v) {
        String s1 = mobileno.getText().toString();
        String s2 = message.getText().toString();
        SmsManager sms = SmsManager.getDefault();
        sms.sendTextMessage(s1, null, s2, null, null);
        Toast.makeText(MainActivity.this, "SMS Sent", Toast.LENGTH_SHORT).show();
    }
}

```

Q.5) Attempt any TWO of the following.**(12 Marks)****a) Develop the registration form using the following GUI.**

Ans: <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
 android:layout_width="match_parent"
 android:layout_height="match_parent"
 android:padding="10sp"
 android:layout_marginTop="60sp">
 <ImageView android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:minHeight="100sp"
 android:id="@+id/userImage"
 android:src="@mipmap/ic_launcher" />
 <EditText
 android:id="@+id/name"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:width="300sp"
 android:layout_marginTop="20sp"
 android:layout_centerInParent="true"
 android:layout_below="@+id/userImage"
 android:inputType="textPersonName"
 android:drawableLeft="@android:drawable/btn_star_big_on"
 android:hint="Name" />
 <EditText
 android:id="@+id/email"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:width="300sp"
 android:layout_marginTop="20sp"
 android:layout_centerInParent="true"
 android:layout_below="@id/name"
 android:drawableLeft="@android:drawable/ic_dialog_email"
 android:hint="Email ID "
 android:inputType="textEmailAddress" />
 <EditText
 android:id="@+id/pass"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:width="300sp"
 android:layout_marginTop="20sp"
 android:layout_centerInParent="true"
 android:layout_below="@id/email"
 android:drawableLeft="@android:drawable/ic_lock_idle_lock"
 android:hint="Password"
 android:inputType="textPassword" />


```
<EditText
    android:id="@+id/confirmpass"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:width="300sp"
    android:layout_marginTop="20sp"
    android:layout_centerInParent="true"
    android:layout_below="@id/pass"
    android:hint="Confirm Password"
    android:drawableLeft="@android:drawable/ic_lock_lock"
    android:inputType="textPassword" />
<EditText
    android:id="@+id/mobile"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:width="300sp"
    android:layout_marginTop="20sp"
    android:layout_centerInParent="true"
    android:hint="Enter Mobile"
    android:drawableLeft="@android:drawable/sym_action_call"
    android:layout_below="@id/confirmpass"
    android:inputType="number" />
<RadioGroup
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/mobile"
    android:layout_marginTop="20sp"
    android:id="@+id/gender">
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:minWidth="200sp"
        android:layout_marginLeft="100sp">
        <RadioButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/male"
            android:text=" Male " />
        <RadioButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/female"
```

```
        android:text=" Female "/>
    </LinearLayout>
</RadioGroup>
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Register"
    android:minWidth="250sp"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="20sp"
    android:id="@+id/reg"
    android:layout_below="@id/gender"/>
</RelativeLayout>
```

b) Write a program to capture an image using a camera and display it.

Ans: Code for activity_main.xml file

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:orientation="vertical">
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/cameraImg"
        android:minHeight="350sp"
        android:layout_marginTop="200dp" />
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/sBtn"
        android:text="Take Picture"
        Android: onClick="clickPhoto"
        android:layout_marginTop="10dp" />
</LinearLayout>
```

// Code for MainActivity.java file

```
public class MainActivity extends AppCompatActivity {
    ImageView iv;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```

    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    iv = (ImageView) findViewById(R.id.cameraImg);
}
public void clickPhoto(View v) {
    Intent cameraIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    startActivityForResult(cameraIntent,0);
}
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    Bitmap bm = (Bitmap) data.getExtras().get("data");
    iv.setImageBitmap(bm);
}
}

```

c) Develop a program to send an email.

Ans: Code for activity_main.xml file →

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="20dp"
    android:paddingRight="20dp"
    android:orientation="vertical" >
    <EditText
        android:id="@+id/txtTo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="To"/>
    <EditText
        android:id="@+id/txtSub"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Subject"/>
    <EditText
        android:id="@+id/txtMsg"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:gravity="top"

```

```
        android:hint="Message"/>
    <Button
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_gravity="right"
        android:text="Send"
        android:id="@+id/btnSend"/>
</LinearLayout>
```

Code for MainActivity.java file --->

```
public class MainActivity extends AppCompatActivity {
    EditText eTo, eSubject, eMsg;
    Button btn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        eTo = (EditText)findViewById(R.id.txtTo);
        eSubject = (EditText)findViewById(R.id.txtSub);
        eMsg = (EditText)findViewById(R.id.txtMsg);
        btn = (Button)findViewById(R.id.btnSend);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(Intent.ACTION_SEND);
                i.putExtra(Intent.EXTRA_EMAIL, new String[]{eTo.getText().toString()});
                i.putExtra(Intent.EXTRA_SUBJECT,eSubject.getText().toString());
                i.putExtra(Intent.EXTRA_TEXT,eMsg.getText());
                i.setType("message/rfc822");
                startActivity(Intent.createChooser(i,"Choose Mail Option"));
            }
        });
    }
}
```

Q.6) Attempt any TWO of the following.

(12 Marks)

a) Develop an application to store student details like roll no, name, branch, marks,percentage and retrieve student information using roll no. in SQLite databases.

Code for activity_main.xml file -->

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:layout_marginLeft="75sp"
    android:layout_marginTop="50sp">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Student Details"
        android:textSize="30sp" />
    <EditText
        android:id="@+id/Rollno"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:inputType="number"
        android:hint="Enter Rollno:"
        android:textSize="20sp" />
    <EditText
        android:id="@+id/Name"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:inputType="text"
        android:hint="Enter Name"
        android:textSize="20sp" />
    <EditText
        android:id="@+id/Branch"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:inputType="text"
        android:hint="Enter Branch"
        android:textSize="20sp" />
    <EditText
        android:id="@+id/Marks"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
```

```

        android:inputType="text"
        android:hint="Enter Marks"
        android:textSize="20sp" />
<EditText
    android:id="@+id/Percentage"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:inputType="text"
    android:hint="Enter Percentage"
    android:textSize="20sp" />
<Button
    android:id="@+id/Insert"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:text="Insert"
    android:textSize="30dp"
    android:onClick="insertData"/>
<Button
    android:id="@+id/View"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:text="View"
    android:textSize="30dp"
    android:onClick="viewData"/>
</LinearLayout>
code for MainActivity.java
public class MainActivity extends AppCompatActivity
{
    EditText Rollno,Name,Branch, Marks, Percentage;
    SQLiteDatabase db;
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Rollno=(EditText) findViewById(R.id.Rollno);
        Name=(EditText) findViewById(R.id.Name);
        Branch=(EditText) findViewById(R.id.Branch);
        Marks=(EditText) findViewById(R.id.Marks);
        Percentage=(EditText) findViewById(R.id.Percentage);
        db=openOrCreateDatabase("studentdb", Context.MODE_PRIVATE, null);
        db.execSQL("CREATE TABLE IF NOT EXISTS student(rollno VARCHAR,name
VARCHAR,branch VARCHAR,marks VARCHAR,percentage VARCHAR);");
    }
    public void insertData(View v){

```

```

db.execSQL("INSERT INTO student VALUES('"+Rollno.getText()+"','"+Name.getText()+"',
"+Branch.getText()+"','"+Marks.getText()+"','"+Percentage.getText()+"');");
}
public void viewData(View v) {
Cursor c=db.rawQuery("SELECT * FROM student WHERE rollno='"+Rollno.getText()+"'",
null);
if(c.moveToFirst())
{
    Name.setText(c.getString(1));
    Branch.setText(c.getString(2));
    Marks.setText(c.getString(3));
    Percentage.setText(c.getString(4));
}
}
}

```

b) Write a program to locate the user's current location. (Write ONLY .java and manifest file)

Ans: code for permission of AndroidManifest.xml file --->

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

code for MainActivity.java file --->

```

public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {
    private GoogleMap mMap;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        SupportMapFragment mapFragment =
(SupportMapFragment)getSupportFragmentManager()
        .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
    }
    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;
        LatLng city = new LatLng(20.04, 74.48);
        mMap.addMarker(new MarkerOptions().position(city).title("Marker in City"));
        mMap.moveCamera(CameraUpdateFactory.newLatLng(city));
    }
}

```

c) Develop a simple calculator using table layout.

Ans: Code for activity_main.xml --->

```
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TableRow>
        <EditText
            android:id="@+id/fn"
            android:layout_width="150dp"
            android:layout_height="wrap_content"
            android:inputType="number"
            android:hint="First No"
            android:textSize="20sp" />
        <EditText
            android:id="@+id/sn"
            android:layout_width="150dp"
            android:layout_height="wrap_content"
            android:inputType="text"
            android:hint="Second No"
            android:textSize="20sp" />
        <EditText
            android:id="@+id/result"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:hint="Result"
            android:textSize="20sp" />    </TableRow>
    <TableRow android:weightSum="4">
        <Button
            android:id="@+id/add"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Add"
            android:textSize="20dp"
            android:onClick="addNo"
            android:layout_weight="2" />
        <Button
            android:id="@+id/sub"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Sub"
            android:textSize="20dp"
            android:onClick="subNo"
            android:layout_weight="2" />
        <Button
            android:id="@+id/mul"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Mul"
            android:textSize="20dp"
            android:onClick="mulNo" />
```



```

<Button
    android:id="@+id/div"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Div"
    android:textSize="20dp"
    android:onClick="divNo" />    </TableRow>
</TableLayout>

```

Code for MainActivity.java file --->

```

public class MainActivity extends AppCompatActivity
{
    EditText fn,sn,ans;
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        fn=(EditText) findViewById(R.id.fn);
        sn=(EditText) findViewById(R.id.sn);
        ans=(EditText) findViewById(R.id.result);
    }
    public void addNo(View v){
        int a,b,c;
        a = Integer.parseInt(fn.getText().toString());
        b = Integer.parseInt(sn.getText().toString());
        c = a + b;
        ans.setText("Ans : " + c );
    }
    public void subNo(View v) {
        int a,b,c;
        a = Integer.parseInt(fn.getText().toString());
        b = Integer.parseInt(sn.getText().toString());
        c = a - b;
        ans.setText("Ans : " + c );
    }
    public void mulNo(View v){
        int a,b,c;
        a = Integer.parseInt(fn.getText().toString());
        b = Integer.parseInt(sn.getText().toString());
        c = a * b;
        ans.setText("Ans : " + c );
    }
    public void divNo(View v) {
        int a,b,c;
        a = Integer.parseInt(fn.getText().toString());
        b = Integer.parseInt(sn.getText().toString());
        c = a / b;
        ans.setText("Ans : " + c );
    }
}

```