



WINTER – 2023 EXAMINATION
Model Answer – Only for the Use of RAC Assessors

Subject Name: Mobile Application Development

Subject Code: 22617

Important Instructions to examiners:

- 1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
- 2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
- 3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills).
- 4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
- 5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
- 6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.
- 7) For programming language papers, credit may be given to any other program based on equivalent concept.
- 8) As per the policy decision of Maharashtra State Government, teaching in English/Marathi and Bilingual (English + Marathi) medium is introduced at first year of AICTE diploma Programme from academic year 2021-2022. Hence if the students in first year (first and second semesters) write answers in Marathi or bilingual language (English + Marathi), the Examiner shall consider the same and assess the answer based on matching of concepts with model answer.

Q. No.	Sub Q. N.	Answer	Marking Scheme
1		Attempt any <u>FIVE</u> of the following:	10 M
	a)	List all tools and software's required for developing an android application.	2 M
	Ans	<ul style="list-style-type: none"> • Android Studio • ADB (Android Debug Bridge) • AVD Manager • Eclipse • Fabric • FlowUp • GameMaker: Studio • Genymotion • Gradle • IntelliJ IDEA 	Any 4 tools One tool for ½ Mark 2 M
	b)	Define emulator.	2 M
	Ans	An Android emulator is a tool that creates virtual Android devices on your computer. The emulator lets you prototype, develop and test Android applications without using a physical device.	Correct Definition 2 M



	c)	List any four attributes of layout.	2 M				
	Ans	<ul style="list-style-type: none">• android:id• android:layout_width• android:layout_height• android:layout_margin• android:layout_marginTop• android:layout_marginBottom• android:layout_marginLeft• android:layout_marginRight• android:background	Any 4 attributes One attribute for ½ Mark 2 M				
	d)	Define Geocoding and Reverse Geocoding.	2 M				
	Ans	<p>Geocoding :</p> <p>Geocoding is the process of transforming a street address or other description of a location into a (latitude, longitude) coordinate.</p> <p>Reverse Geocoding :</p> <p>Reverse geocoding is the process of transforming a (latitude, longitude) coordinate into a (partial) address.</p>	Geocoding:1 M Reverse Geocoding:1 M				
	e)	State intent. List types of intent.	2 M				
	Ans	<p>Intent is the message that is passed between components such as activities.</p> <p>Android uses Intent for communicating between the components of an Application and also from one application to another application.</p> <p>Types:</p> <ul style="list-style-type: none">• Explicit Intent• Implicit Intent	Definition 1 M Types Listing 1 M				
	f)	Write difference between toggle button and radio button.	2 M				
	Ans	<table><tr><td>Toggle button</td><td>Radio button</td></tr><tr><td>1. Toggle Button can be used to display checked/unchecked (On/Off) state on the button. 2. For ON/OFF state two labels can be assign.</td><td>1. Radio Buttons are used to choose a single option from a list 2. There is only one label for radio button.</td></tr></table>	Toggle button	Radio button	1. Toggle Button can be used to display checked/unchecked (On/Off) state on the button. 2. For ON/OFF state two labels can be assign.	1. Radio Buttons are used to choose a single option from a list 2. There is only one label for radio button.	For 2 points : 2 M
Toggle button	Radio button						
1. Toggle Button can be used to display checked/unchecked (On/Off) state on the button. 2. For ON/OFF state two labels can be assign.	1. Radio Buttons are used to choose a single option from a list 2. There is only one label for radio button.						



	g)	Define: i)Fragment ii) Broadcast receiver	2 M																
	Ans	Fragment: Fragment is the part of activity, it is also known as sub-activity. Broadcast receiver: A broadcast receiver is a dormant component of the Android system. The Broadcast Receiver’s job is to pass a notification to the user, in case a specific event occurs.	Fragment :1 M Broadcast receiver : 1 M																
2.		Attempt any <u>THREE</u> of the following:	12 M																
	a)	Explain relative layout with all its attributes.	4 M																
	Ans	In Relative Layout we need to specify the position of child views relative to each other or relative to the parent. In case if we didn’t specify the position of child views, by default all child views are positioned to top-left of the layout. Attributed of Relative Layout: <table><tr><th>Attribute</th><th>Description</th></tr><tr><td>layout_alignParentTop</td><td>If it specified “true”, the top edge of view will match the top edge of parent.</td></tr><tr><td>layout_alignParentBottom</td><td>If it specified “true”, the bottom edge of view will match the bottom edge of parent.</td></tr><tr><td>layout_alignParentLeft</td><td>If it specified “true”, the left edge of view will match the left edge of parent.</td></tr><tr><td>layout_alignParentRight</td><td>If it specified “true”, the right edge of view will match the right edge of parent.</td></tr><tr><td>layout_centerInParent</td><td>If it specified “true”, the view will be aligned to center of parent.</td></tr><tr><td>layout_centerHorizontal</td><td>If it specified “true”, the view will be horizontally centre aligned within its parent.</td></tr><tr><td>layout_centerHorizontal</td><td>If it specified “true”, the view will be</td></tr></table>	Attribute	Description	layout_alignParentTop	If it specified “true”, the top edge of view will match the top edge of parent.	layout_alignParentBottom	If it specified “true”, the bottom edge of view will match the bottom edge of parent.	layout_alignParentLeft	If it specified “true”, the left edge of view will match the left edge of parent.	layout_alignParentRight	If it specified “true”, the right edge of view will match the right edge of parent.	layout_centerInParent	If it specified “true”, the view will be aligned to center of parent.	layout_centerHorizontal	If it specified “true”, the view will be horizontally centre aligned within its parent.	layout_centerHorizontal	If it specified “true”, the view will be	Explanation 1 M Any three Attributes with description- 3 M
Attribute	Description																		
layout_alignParentTop	If it specified “true”, the top edge of view will match the top edge of parent.																		
layout_alignParentBottom	If it specified “true”, the bottom edge of view will match the bottom edge of parent.																		
layout_alignParentLeft	If it specified “true”, the left edge of view will match the left edge of parent.																		
layout_alignParentRight	If it specified “true”, the right edge of view will match the right edge of parent.																		
layout_centerInParent	If it specified “true”, the view will be aligned to center of parent.																		
layout_centerHorizontal	If it specified “true”, the view will be horizontally centre aligned within its parent.																		
layout_centerHorizontal	If it specified “true”, the view will be																		



		<table><tr><td></td><td>vertically centre aligned within its parent.</td></tr><tr><td>layout_above</td><td>It accepts another sibling view id and places the view above the specified view id.</td></tr><tr><td>layout_below</td><td>It accepts another sibling view id and places the view below the specified view id.</td></tr><tr><td>layout_toLeftOf</td><td>It accepts another sibling view id and places the view left of the specified view id.</td></tr><tr><td>layout_toRightOf</td><td>It accepts another sibling view id and places the view right of the specified view id.</td></tr><tr><td>layout_toStartOf</td><td>It accepts another sibling view id and places the view to start of the specified view id.</td></tr><tr><td>layout_toEndOf</td><td>It accepts another sibling view id and places the view to end of the specified view id.</td></tr></table>		vertically centre aligned within its parent.	layout_above	It accepts another sibling view id and places the view above the specified view id.	layout_below	It accepts another sibling view id and places the view below the specified view id.	layout_toLeftOf	It accepts another sibling view id and places the view left of the specified view id.	layout_toRightOf	It accepts another sibling view id and places the view right of the specified view id.	layout_toStartOf	It accepts another sibling view id and places the view to start of the specified view id.	layout_toEndOf	It accepts another sibling view id and places the view to end of the specified view id.	
	vertically centre aligned within its parent.																
layout_above	It accepts another sibling view id and places the view above the specified view id.																
layout_below	It accepts another sibling view id and places the view below the specified view id.																
layout_toLeftOf	It accepts another sibling view id and places the view left of the specified view id.																
layout_toRightOf	It accepts another sibling view id and places the view right of the specified view id.																
layout_toStartOf	It accepts another sibling view id and places the view to start of the specified view id.																
layout_toEndOf	It accepts another sibling view id and places the view to end of the specified view id.																
	b)	Explore all steps to install Android studio and SDK	4 M														
Ans	<p>Pre-Installation Check List</p> <ol style="list-style-type: none">1. Before installing Android SDK, there is need to install Java Development Kit (JDK). Ensure that JDK is at or above 1.8.2. Uninstall older version(s) of "Android Studio" and "Android SDK", if any. <p>We need to install two packages:</p> <ol style="list-style-type: none">1. Android Studio (IDE), which is an Integrated Development Environment (IDE)2. Android SDK (Software Development Kit) for developing and running Android apps. <p>Steps to install Android studio:</p> <p>Download Android Studio</p> <ol style="list-style-type: none">1. Click Download Android Studio. The Terms and Conditions page with the Android Studio License Agreement opens.2. Read the License Agreement.3. At the bottom of the page, if you agree with the terms and conditions, select the I have read and agree with the above terms and		<p>Android studio installation 2 M</p> <p>SDK installation 2 M</p>														



		<p>conditions checkbox.</p> <ol style="list-style-type: none">Click Download Android Studio to start the download.When prompted, save the file to a location where you can easily locate it, such as the Downloads folder.Wait for the download to complete. <p>Install Android Studio</p> <ol style="list-style-type: none">Open the folder where you downloaded and saved the Android Studio installation file.Double-click the downloaded file.If you see a User Account Control dialog about allowing the installation to make changes to your computer, click Yes to confirm the installation.Click Next to start the installation.Accept the default installation settings for all steps. <ol style="list-style-type: none">Click finish when installation is done. <p>Installing Android SDK</p> <p>Within Android Studio, you can install the Android SDK as follows:</p> <ol style="list-style-type: none">Click Tools > SDK Manager.In the SDK Platforms tab, select Android Tiramisu Preview.In the SDK Tools tab, select Android SDK Build.Click OK to install the SDK.	
	c)	Explain the need of Android Operating System. Also describe any four features of android.	4 M
	Ans	<p>Need of Android Operating System</p> <ul style="list-style-type: none">Zero/negligible development cost: The development tools like Android SDK, JDK, and Eclipse IDE etc. are free to download for the android mobile application development.Open Source: The Android OS is an open-source platform based on the Linux kernel and multiple open-source libraries. In this way developers are free to contribute or extend the platform as necessary for building mobile apps which run on	<p>Need : 2 M</p> <p>Any 4 features : 4 M</p>



Android devices.

- Multi-Platform Support

In market, there are a wide range of hardware devices powered by the Android OS, including many different phones and tablet. Even development of android mobile apps can occur on Windows, Mac OS or Linux.

- Multi-Carrier Support

World wide a large number of telecom carriers like Airtel, Vodafone, Idea Cellular, AT&T Mobility, BSNL etc. are supporting Android powered phones.

- Open Distribution Model

Android Market place (Google Play store) has very few restrictions on the content or functionality of an android app. So the developer can distribute theirs app through Google Play store and as well other distribution channels like Amazon's app store.

Four features of android

1) Near Field Communication (NFC)

Most Android devices support NFC, which allows electronic devices to easily interact across short distances.

2) Alternate Keyboards

Android supports multiple keyboards and makes them easy to install; the SwiftKey, Skype, and 8pen apps all offer ways to quickly change up your keyboard style.

3) Infrared Transmission

The Android operating system supports a built-in infrared transmitter, allowing you to use your phone or tablet as a remote control.

4) No-Touch Control

Using Android apps such as Wave Control, users can control their phones touch-free, using only gestures.

5) Automation

The Tasker app controls the app permissions but also automate them

6) Wireless App Downloads

Using the Android Market or third-party options like AppBrain, we can download apps on PC and then automatically sync them with Android, no plugging required.

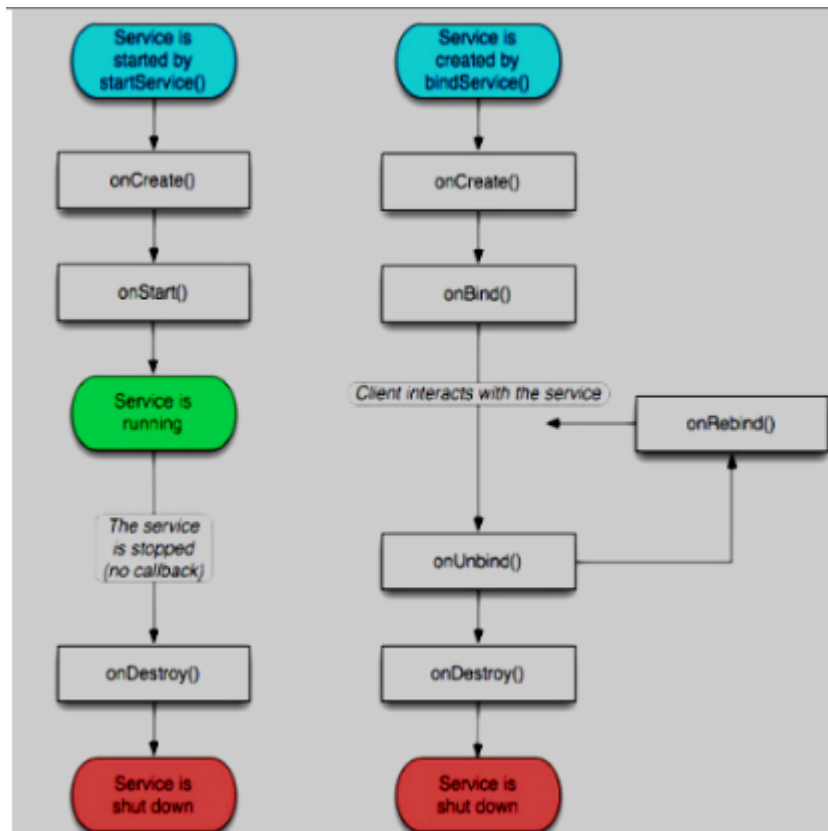


	<p>7) Storage and Battery Swap</p> <p>Android phones also have unique hardware capabilities. Google's OS makes it possible to remove and upgrade your battery or to replace one that no longer holds a charge</p> <p>8) Custom Home Screens</p> <p>While it's possible to hack certain phones to customize the home screen, Android comes with this capability from the get-go</p> <p>9) Widgets</p> <p>Apps are versatile, but sometimes you want information at a glance instead of having to open an app and wait for it to load. Android widgets let you display just about any feature you choose, right on the home screen—including weather apps, music widgets, or productivity tools that helpfully remind you of upcoming meetings or approaching deadlines.</p>	
d)	Develop a program to add "Hello World" marker at (10 ,10) co-ordinates. Write only . java file.	4 M
Ans	<p>Activity_maps.xml</p> <pre>package com.example.googlemap; import com.google.android.gms.maps.GoogleMap; import com.google.android.gms.maps.OnMapReadyCallback; import com.google.android.gms.maps.SupportMapFragment; import com.google.android.gms.maps.model.LatLng; import com.google.android.gms.maps.model.MarkerOptions; public class MapsActivity extends FragmentActivity implements OnMapReadyCallback { private GoogleMap mMap; @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity_maps); // Obtain the SupportMapFragment and get notified when the map is ready to be used. SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager() .findFragmentById(R.id.map); mapFragment.getMapAsync(this); } @Override public void onMapReady(GoogleMap googleMap) {</pre>	Any correct logic program 4 Marks



		<pre> mMap = googleMap; // Add a marker in Sydney and move the camera LatLng hello = new LatLng(10,10); mMap.addMarker(new MarkerOptions().position(hello).title("Marker Hello World")); mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(mumbai,10F)); } } </pre>	
3.		Attempt any <u>THREE</u> of the following:	12 M
	a)	Describe service life cycle with its diagram.	4 M
	Ans	<p>A service is an application component which runs without direct interaction with the user in the background.</p> <ul style="list-style-type: none"> Services are used for repetitive and potentially long running operations, i.e., Internet downloads, checking for new data, data processing, updating content providers and the like. Service can either be started or bound we just need to call either <code>startService()</code> or <code>bindService()</code> from any of our android components. Based on how our service was started it will either be “started” or “bound” <p>Service Lifecycle:</p> <p>1. Started</p> <p>a. A service is started when an application component, such as an activity, starts it by calling <code>startService()</code>.</p> <p>b. Now the service can run in the background indefinitely, even if the component that started it is destroyed.</p> <p>2. Bound</p> <p>a. A service is bound when an application component binds to it by calling <code>bindService()</code>.</p> <p>b. A bound service offers a client-server interface that allows components to interact with the service, send requests, get results, and even do so across processes with InterProcess Communication (IPC).</p> <p>c. Like any other components service also has callback methods. These will be invoked while the service is running to inform the application of its state. Implementing these in our custom service would help you in performing the right</p>	Explanation 2 M, Diagram 2 M

operation in the right state.



d. There is always only a single instance of service running in the app. If you are calling `startService()` for a single service multiple times in our application it just invokes the `onStartCommand()` on that service. Neither is the service restarted multiple times nor are its multiple instances created.

1. `onCreate()`:

This is the first callback which will be invoked when any component starts the service. If the same service is called again while it is still running this method Won't be invoked. Ideally one time setup and intializing should be done in this callback.

2. `onStartCommand() /startSetvice()`

This callback is invoked when service is started by any component by calling `startService()`. It basically indicates that the service has started and can now run indefinitely.

3. `onBind()`

To provide binding for a service, you must implement the `onBind()` callback method. This method returns an `IBinder` object that defines the programming interface that clients can use to interact with the service.



		<p>4. onUnbind()</p> <p>This is invoked when all the clients are disconnected from the service.</p> <p>5. onRebind()</p> <p>This is invoked when new clients are connected to the service. It is called after onRebind</p> <p>6. onDestroy()</p> <p>This is a final clean up call from the system. This is invoked just before the service is being destroyed.</p>	
	b)	Elaborate Android Security Model.	4 M
	Ans	<ul style="list-style-type: none">• The Android security model is primarily based on a sandbox and permission mechanism.• Each application is running in a specific Dalvik virtual machine with a unique user ID assigned to it, which means the application code runs in isolation from the code of all other applications.• Therefore, one application has not granted access to other applications' files.• Android application has been signed with a certificate with a private key. Know the owner of the application is unique.• This allows the author of the application will be identified if needed. When an application is installed in the phone is assigned a user ID, thus avoiding it from affecting other applications by creating a sandbox for it.• This user ID is permanent on which devices and applications with the same user ID are allowed to run in a single process.• This is a way to ensure that a malicious application has Cannot access / compromise the data of the genuine application.• It is mandatory for an application to list all the resources it will Access during installation. Terms are required of an application, in the installation process should be user-based or interactive Check with the signature of the application <p>Declaring and Using Permissions</p> <p>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.</p> <p>Permissions are divided into several protection levels. The protection level</p>	2 M for explanation , 2 M for explaining permissions, any 2 permissions expected



		<p>affects whether runtime permission requests are required. There are three protection levels that affect third party apps: normal, signature, and dangerous permissions.</p> <p>Normal permissions: Normal permissions cover areas where your app needs to access data or resources outside the app's sandbox, but where there's very little risk to the user's privacy or the operation of other apps. For example, permission to set the time zone is a normal permission. If an app declares in its manifest that it needs a normal permission, the system automatically grants the app that permission at install time. The system doesn't prompt the user to grant normal permissions, and users cannot revoke these permissions.</p> <p>Signature permissions: The system grants these app permissions at install time, but only when the app that attempts to use permission is signed by the same certificate as the app that defines the permission.</p> <p>Dangerous permissions: Dangerous permissions cover areas where the app wants data or resources that involve the user's private information, or could potentially affect the user's stored data or the operation of other apps. For example, the ability to read the user's contacts is a dangerous permission. If an app declares that it needs a dangerous permission, the user must explicitly grant the permission to the app. Until the user approves the permission, your app cannot provide functionality that depends on that permission. To use a dangerous permission, your app must prompt the user to grant permission at runtime. For more details about how the user is prompted, see Request prompt for dangerous permission.</p>	
	c)	Write an xml file to create login page using Table Layout.	4 M
	Ans	<pre><TableLayout xmlns:android="http://schemas.android.com/apk/res/android" android:layout_width="match_parent" android:layout_height="match_parent" android:background="#000" android:orientation="vertical" android:stretchColumns="1"> <TableRow android:padding="5dip"> <TextView android:layout_height="wrap_content" android:layout_marginBottom="20dp" android:layout_span="2" android:gravity="center_horizontal"</pre>	2 M for correct structure of Table Layout , 2 M for attributes



		<pre>android:text="@string/loginForm" android:textColor="#0ff" android:textSize="25sp" android:textStyle="bold" /> </TableRow> <TableRow> <TextView android:layout_height="wrap_content" android:layout_column="0" android:layout_marginLeft="10dp" android:text="@string/userName" android:textColor="#fff" android:textSize="16sp" /> <EditText android:id="@+id/userName" android:layout_height="wrap_content" android:layout_column="1" android:layout_marginLeft="10dp" android:background="#fff" android:hint="@string/userName" android:padding="5dp" android:textColor="#000" /> </TableRow> <TableRow> <TextView android:layout_height="wrap_content" android:layout_column="0" android:layout_marginLeft="10dp"</pre>	
--	--	--	--



```
android:layout_marginTop="20dp"

android:text="@string/password"

android:textColor="#fff"

android:textSize="16sp" />

<EditText

    android:id="@+id/password"

    android:layout_height="wrap_content"

    android:layout_column="1"

    android:layout_marginLeft="10dp"

    android:layout_marginTop="20dp"

    android:background="#fff"

    android:hint="@string/password"

    android:padding="5dp"

    android:textColor="#000" />

</TableRow>

<TableRow android:layout_marginTop="20dp">

    <Button

        android:id="@+id/loginBtn"

        android:layout_height="wrap_content"

        android:layout_gravity="center"

        android:layout_span="2"

        android:background="#0ff"

        android:text="@string/login"

        android:textColor="#000"

        android:textSize="20sp"

        android:textStyle="bold" />

    </TableRow>

</TableLayout>
```



	d)	Develop an application to display analog Time Picker. Also display the selected time. (Write only . java file)	4 M
	Ans	MainActivity.java import android.os.Bundle; import android.view.View; import android.widget.Button; import android.widget.TextView; import android.widget.TimePicker; public class MainActivity extends AppCompatActivity { TextView textview1; TimePicker timepicker; Button changetime; @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity_main); textview1=(TextView)findViewById(R.id.textView1); timepicker=(TimePicker)findViewById(R.id.timePicker); //Uncomment the below line of code for 24 hour view timepicker.setIs24HourView(true); changetime=(Button)findViewById(R.id.button1); textview1.setText(getCurrentTime()); changetime.setOnClickListener(new View.OnClickListener(){ @Override public void onClick(View view) { textview1.setText(getCurrentTime()); } }); } };	Code for display time picker 2 M and display time 2 M



		<pre>} public String getCurrentTime(){ String currentTime="Current Time: "+timepicker.getCurrentHour()+":"+timepicker.getCurrentMinute(); return currentTime; } }</pre>															
4.		Attempt any <u>THREE</u> of the following:	12 M														
	a)	Differentiate between JVM and DVM. (Any four points)	4 M														
	Ans	<table><tr><th>DVM</th><th>JVM</th></tr><tr><td>It is Register based which is designed to run on low memory.</td><td>It is Stack based.</td></tr><tr><td>DVM uses its own byte code and runs the “.Dex” file. From Android 2.2 SDK Dalvik has got a Just in Time compiler</td><td>JVM uses java byte code and runs “.class” file having JIT (Just In Time).</td></tr><tr><td>DVM has been designed so that a device can run multiple instances of the VM efficiently. Applications are given their own instance</td><td>A single instance of JVM is shared with multiple applications.</td></tr><tr><td>DVM supports the Android operating system only.</td><td>JVM supports multiple operating systems.</td></tr><tr><td>There is a constant pool for every application.</td><td>It has a constant pool for every class.</td></tr><tr><td>Here the executable is APK.</td><td>Here the executable is JAR</td></tr></table>	DVM	JVM	It is Register based which is designed to run on low memory.	It is Stack based.	DVM uses its own byte code and runs the “.Dex” file. From Android 2.2 SDK Dalvik has got a Just in Time compiler	JVM uses java byte code and runs “.class” file having JIT (Just In Time).	DVM has been designed so that a device can run multiple instances of the VM efficiently. Applications are given their own instance	A single instance of JVM is shared with multiple applications.	DVM supports the Android operating system only.	JVM supports multiple operating systems.	There is a constant pool for every application.	It has a constant pool for every class.	Here the executable is APK.	Here the executable is JAR	Any 4 points of differences :1 M each
DVM	JVM																
It is Register based which is designed to run on low memory.	It is Stack based.																
DVM uses its own byte code and runs the “.Dex” file. From Android 2.2 SDK Dalvik has got a Just in Time compiler	JVM uses java byte code and runs “.class” file having JIT (Just In Time).																
DVM has been designed so that a device can run multiple instances of the VM efficiently. Applications are given their own instance	A single instance of JVM is shared with multiple applications.																
DVM supports the Android operating system only.	JVM supports multiple operating systems.																
There is a constant pool for every application.	It has a constant pool for every class.																
Here the executable is APK.	Here the executable is JAR																
	b)	Explain components of android directory structure.	4 M														
	Ans	<p>The android project contains different types of app modules, source code files, and resource files.</p> <p>Following are the components/ modules of android directory:</p> <p>1) Manifests Folder Manifests folder contains AndroidManifest.xml for creating our android application. This file contains information about our application such as the</p>	1 M for each directory														



		<p>Android version, metadata, states package for Kotlin file, and other application components. It acts as an intermediary between android OS and our application.</p> <p>2) Java folder The Java folder contains all the java source code (.java) files that we create during the app development, including other Test files.</p> <p>3) Resource (res) folder The resource folder is the most important folder because it contains all the non-code sources like images, XML layouts, and UI strings for our android application.</p> <ul style="list-style-type: none">• res/drawable folder It contains the different types of images used for the development of the application. We need to add all the images in a drawable folder for the application development.• res/layout folder The layout folder contains all XML layout files which we used to define the user interface of our application. It contains the activity_main.xml file.• res/mipmap folder This folder contains launcher.xml files to define icons that are used to show on the home screen. It contains different density types of icons depending upon the size of the device such as hdpi, mdpi, xhdpi.• res/values folder Values folder contains a number of XML files like strings, dimensions, colors, and style definitions. One of the most important files is the strings.xml file which contains the resources. <p>4) Gradle Scripts folder Gradle means automated build system and it contains a number of files that are used to define a build configuration that can be applied to all modules in our application. In build.gradle (Project) there are buildscripts and in build.gradle (Module) plugins and implementations are used to build configurations that can be applied to all our application modules.</p>	
	c)	Develop an android application using radio button.	4 M
	Ans	<p>[Consider any relevant example of Radio Button and in XML file, consider minimum attributes]</p> <p>activity_main.xml</p> <pre><?xml version="1.0" encoding="utf-8"?> <RelativeLayout 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</pre>	XML Code 2 M, Java Code 2 M



```
android:layout_width="match_parent"  
android:layout_height="match_parent"  
android:padding="30dp"  
tools:context=".frame">
```

```
<TextView android:id="@+id/text1"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:text="Radio Button"  
android:textSize="20dp"  
android:gravity="center"  
android:textColor="#f00"/>
```

```
<RadioGroup android:id="@+id/group"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_below="@+id/text1">
```

```
<RadioButton android:id="@+id/male"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="Male"/>
```

```
<RadioButton  
android:id="@+id/female"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_below="@+id/male"  
android:text="Female"/>  
</RadioGroup>
```

```
<Button android:id="@+id/submit"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_below="@+id/group"  
android:layout_marginTop="99dp"  
android:layout_centerHorizontal="true"  
android:text="Submit" />
```

```
</RelativeLayout>
```

Java File:

```
package com.example.ifcdiv;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
  
import android.view.View;  
  
import android.widget.Button;
```



```
import android.widget.RadioButton;

import android.widget.Toast;

public class frame extends AppCompatActivity

{

RadioButton male,female;

Button b1;

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.activity_frame);

male=findViewById(R.id.male);

female=findViewById(R.id.female);

b1=findViewById(R.id.submit);

b1.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

String selected; if(male.isChecked())

{

selected="You selected"+male.getText();

}

else

{

selected="You Selected"+female.getText();

}

Toast.makeText(getApplicationContext(),selected,Toast.LENGTH_LONG).show();

}

});

}
```



	d)	Develop an application to send and receive SMS. (Write only Java and permission tag in manifest file)	4 M
	Ans	<p>Permissions and <receiver> tag required in AndroidManifest.xml</p> <pre><uses-permission android:name="android.permission.RECEIVE_SMS" /> <uses-permission android:name="android.permission.SEND_SMS"/> <uses-permission android:name="android.permission.READ_SMS"/> <uses-permission android:name="android.permission.WRITE_SMS"/> <receiver android:name=".SmsReceiver" android:enabled="true" android:exported="true"> <intent-filter> <action android:name="android.provider.Telephony.SMS_RECEIVED" /> </intent-filter> </receiver></pre> <p>MainActivity.java</p> <p>(Considering appropriate layout file with 2 edit text boxes namely for phone number, message and a button for sending sms)</p> <pre>package com.example.testreceivesms; import androidx.appcompat.app.AppCompatActivity; import androidx.core.app.ActivityCompat; import androidx.core.content.ContextCompat; import android.Manifest; import android.content.IntentFilter; import android.content.pm.PackageManager; import android.os.Bundle; import android.telephony.SmsManager; import android.view.View;</pre>	(Permission 2 M, Java Code 2 M)



```
import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    SmsReceiver sms= new SmsReceiver();

    EditText et1,et2;

    Button b1;

    @Override

    protected void onCreate (Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        et1=findViewById(R.id.etPhno);

        et2=findViewById(R.id.etmsg);

        b1=findViewById(R.id.btnSms);

        if(ContextCompat.checkSelfPermission(MainActivity.this,Manifest.permission.SEND_SMS)!=

        PackageManager.PERMISSION_GRANTED)

        {

            ActivityCompat.requestPermissions(MainActivity.this,new

            String[]{Manifest.permission.SEND_SMS},100);

        }

        b1.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View v) {

                try {

                    String phno= et1.getText().toString();

                    String msg=et2.getText().toString();

                    SmsManager smsManager= SmsManager.getDefault();

                    smsManager.sendTextMessage(phno,null,msg,null,null);
```



```
Toast.makeText(MainActivity.this,"Sms sent successfully",
Toast.LENGTH_LONG).show();
}
catch(Exception e)
{
    Toast.makeText(MainActivity.this,"Sms failed to send... try again",
    Toast.LENGTH_LONG).show();
}
});
}

@Override
protected void onStart() {
    super.onStart();

    IntentFilter filter=new
    IntentFilter("android.provider.Telephony.SMS_RECEIVED");

    registerReceiver(sms,filter);
}

@Override
protected void onStop() {
    super.onStop();
    unregisterReceiver(sms);
}
}
```

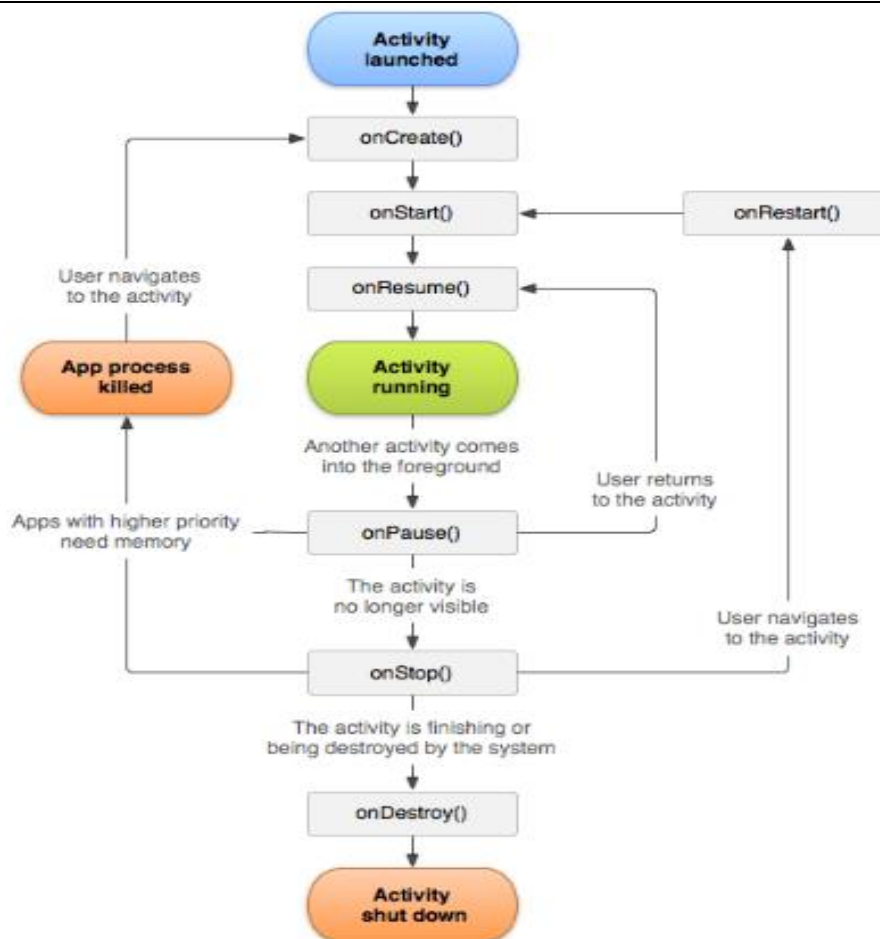
SmsReceiver.java

```
package com.example.testreceivesms;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
```



	<pre>import android.os.Bundle; import android.telephony.SmsMessage; import android.widget.Toast; public class SmsReceiver extends BroadcastReceiver { SmsReceiver(){} @Override public void onReceive(Context context, Intent intent) { Bundle bundle = intent.getExtras(); if (bundle != null) { // Retrieve the SMS Messages received Object[] sms = (Object[]) bundle.get("pdus"); // For every SMS message received for (int i=0; i < sms.length; i++) { // Convert Object array SmsMessage smsMessage = SmsMessage.createFromPdu((byte[]) sms[i]); String phone = smsMessage.getOriginatingAddress(); String message = smsMessage.getMessageBody().toString(); Toast.makeText(context, "Received from "+ phone + ": " + message, Toast.LENGTH_SHORT).show(); } } } }</pre>	
e)	Draw and explain activity life cycle.	4 M
Ans	Activities have a predefined life-cycle methods as follows:	Diagram 2 M, Explanation 2 M



onCreate (): Called then the activity is created. Used to initialize the activity, for example create the user interface.

onStart (): called when activity is becoming visible to the user.

onResume (): Called if the activity get visible again and the user starts interacting with the activity again. Used to initialize fields, register listeners, bind to services, etc.

onPause (): Called once another activity gets into the foreground. Always called before the activity is not visible anymore. Used to release resources or save application data. For example you unregister listeners, intent receivers, unbind from services or remove system service listeners.

onStop (): Called once the activity is no longer visible. Time or CPU intensive shutdown operations, such as writing information to a database should be down in the onStop() method. This method is guaranteed to be called as of API 11.

onDestroy (): called before the activity is destroyed.



5.		Attempt any <u>TWO</u> of the following:	12 M
	a)	Develop a program to perform addition, subtraction, division, multiplication of two numbers and display the result. (Use appropriate UI controls).	6 M
	Ans	<p>Step 1 – Create a new project in Android Studio, go to File ⇒ New Project and fill all required details to create a new project.</p> <p>Step 2 – Add the following code to res/layout/activity_main.xml.</p> <pre><?xml version="1.0" encoding="utf-8"?> <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android" android:layout_width="match_parent" android:layout_height="match_parent" android:padding="4dp"> <TextView android:id="@+id/textResult" android:layout_width="wrap_content" android:layout_height="wrap_content" android:layout_centerHorizontal="true" android:layout_marginTop="70dp" android:background="#008080" android:padding="5dp" android:text="Code4Example" android:textColor="#fff" android:textSize="24sp" android:textStyle="bold" /> <EditText android:id="@+id/editNum1" android:inputType="number" android:layout_width="match_parent"</pre>	<p>Any appropriate UI controls and layout with correct activity_main.xml file-3 M</p> <p>MainActivity.java file-3 M</p>



```
android:layout_height="wrap_content"

android:layout_centerInParent="true" />

<EditText

    android:id="@+id/editNum2"

    android:inputType="number"

    android:layout_width="match_parent"

    android:layout_height="wrap_content"

    android:layout_below="@+id/editNum1"

    android:layout_centerInParent="true" />

<GridLayout

    android:layout_centerHorizontal="true"

    android:layout_centerInParent="true"

    android:layout_below="@+id/editNum2"

    android:columnCount="2"

    android:rowCount="2"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content">

    <Button

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:layout_centerInParent="true"

        android:layout_margin="1dp"

        android:onClick="btnAdd"

        android:text="+" />

    <Button

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:layout_centerInParent="true"
```



```
android:layout_margin="1dp"
```

```
android:onClick="btnSub"
```

```
android:text="-" />
```

```
<Button
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:layout_centerInParent="true"
```

```
android:layout_margin="1dp"
```

```
android:onClick="btnMul"
```

```
android:text="*" />
```

```
<Button
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:layout_centerInParent="true"
```

```
android:layout_margin="1dp"
```

```
android:onClick="btnDiv"
```

```
android:text="/" />
```

```
</GridLayout>
```

```
</RelativeLayout>
```

Step 3 – Add the following code to src/MainActivity.java

```
import androidx.appcompat.app.AppCompatActivity;
```

```
import android.os.Bundle;
```

```
import android.view.View;
```

```
import android.widget.EditText;
```

```
import android.widget.TextView;
```

```
public class MainActivity extends AppCompatActivity {
```

```
    EditText editNum1,editNum2;
```

```
    TextView textResult;
```



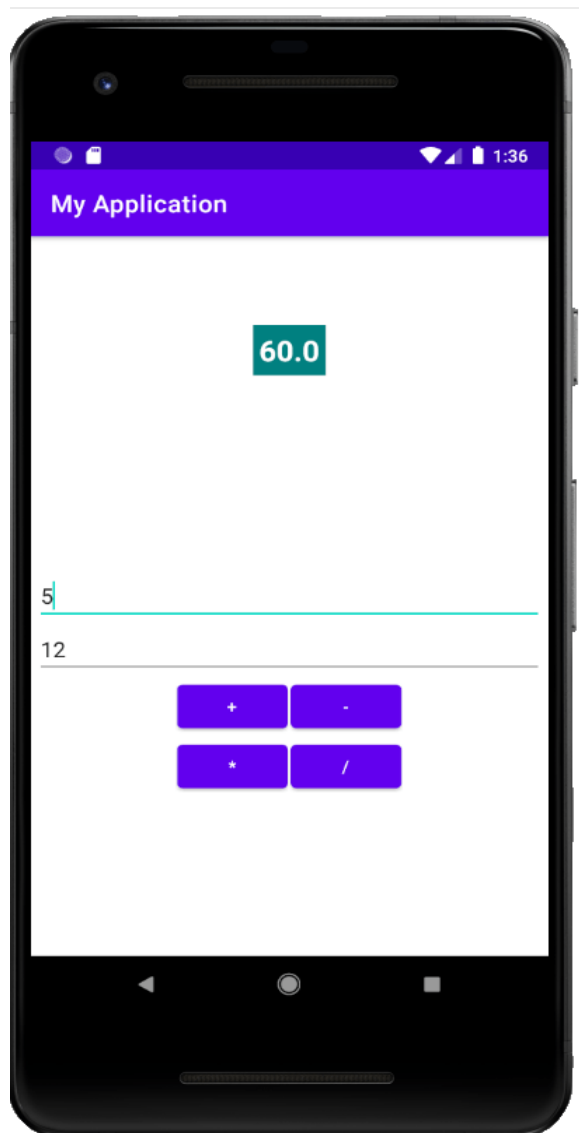
@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    editNum1= findViewById(R.id.editNum1);  
    editNum2= findViewById(R.id.editNum2);  
    textResult= findViewById(R.id.textResult);  
}  
  
public void btnAdd(View view){  
    double num1 = Double.parseDouble(editNum1.getText().toString());  
    double num2 = Double.parseDouble(editNum2.getText().toString());  
    double result = num1 + num2;  
    textResult.setText(Double.toString(result));  
}  
  
public void btnSub(View view){  
    double num1 = Double.parseDouble(editNum1.getText().toString());  
    double num2 = Double.parseDouble(editNum2.getText().toString());  
    double result = num1 - num2;  
    textResult.setText(Double.toString(result));  
}  
  
public void btnMul(View view){  
    double num1 = Double.parseDouble(editNum1.getText().toString());  
    double num2 = Double.parseDouble(editNum2.getText().toString());  
    double result = num1 * num2;  
    textResult.setText(Double.toString(result));  
}  
  
public void btnDiv(View view){
```



```
double num1 = Double.parseDouble(editNum1.getText().toString());  
double num2 = Double.parseDouble(editNum2.getText().toString());  
double result = num1 / num2;  
textResult.setText(Double.toString(result));  
}  
}
```

Output:



		<pre>double num1 = Double.parseDouble(editNum1.getText().toString()); double num2 = Double.parseDouble(editNum2.getText().toString()); double result = num1 / num2; textResult.setText(Double.toString(result)); } }</pre> <p>Output:</p> 	
b)	Develop an application to display a Google Map. (Write JAVA & Manifest file)	6 M	
An	AndroidManifest.xml code:	correct Android Manifest.xml file-3	



S	<pre><?xml version="1.0" encoding="utf-8"?> <manifest xmlns:android="http://schemas.android.com/apk/res/android" package="example.com.mapexample"> <!-- The ACCESS_COARSE/FINE_LOCATION permissions are not requir ed to use Google Maps Android API v2, but you must specify either coarse or fine location permissions for the 'MyLocation' functionality. --> <uses- permission android:name="android.permission.ACCESS_FINE_LOCATION " /> <uses- permission android:name="android.permission.ACCESS_COARSE_LOCAT ION" /> <uses-permission android:name="android.permission.INTERNET" /> <application android:allowBackup="true" android:icon="@mipmap/ic_launcher" android:label="@string/app_name" android:roundIcon="@mipmap/ic_launcher_round" android:supportsRtl="true" android:theme="@style/AppTheme"> <meta-data android:name="com.google.android.geo.API_KEY" android:value="@string/google_maps_key" /> <activity android:name=".MapsActivity" android:label="@string/title_activity_maps"> <intent-filter> <action android:name="android.intent.action.MAIN" /> <category android:name="android.intent.category.LAUNCHER" /> </intent-filter> </activity> </manifest></pre>	M MapsActivity.java file-3 M
---	---	------------------------------------



```
</intent-filter>
</activity>
</application>
</manifest>
```

code of MapsActivity.java :

```
package example.com.mapexample;

import android.support.v4.app.FragmentActivity;
import android.os.Bundle;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;

public class MapsActivity extends FragmentActivity implements OnMapReadyC
allback{
    private GoogleMap mMap;

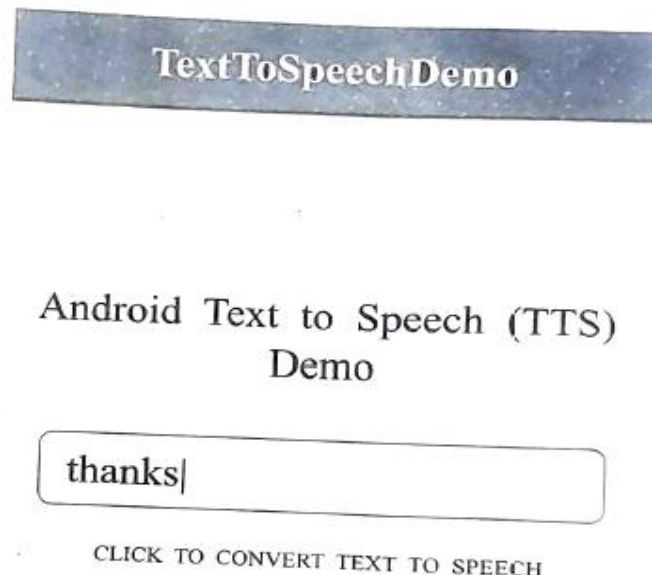
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);

        // Obtain the SupportMapFragment and get notified when the map is ready to be used
        .

        SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragment
Manager()
            .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
    }

    @Override
    public void onMapReady(GoogleMap googleMap) {
```



		<pre>mMap = googleMap; // Add a marker in Sydney and move the camera LatLng sydney = new LatLng(-34, 151); mMap.addMarker(new MarkerOptions().position(sydney).title("Marker in Sydney")); mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney)); } }</pre>	
c)	<p>Develop an application to convert thanks" text to speech as given in the following GUI.</p> 	6 M	
Ans	<p>Code of activity_main.xml</p> <pre><?xml version="1.0" encoding="utf-8"?> <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"</pre>	<p>activity_main.xml file- 2M</p> <p>toolbar_title_layout. xml file-1 M</p> <p>MainActivity.java file-3 M</p> <p>(Any other correct logic can be considered)</p>	



```
android:paddingBottom="@dimen/activity_vertical_margin"

tools:context=".MainActivity"

android:transitionGroup="true">

<TextView

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:text="Android Text to Speech(TTS) Demo"

    android:id="@+id/textView"

    android:layout_below="@+id/textview"

    android:layout_centerHorizontal="true"

    android:textColor="#ff7aff24"

    android:textSize="35dp" />

<EditText

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:id="@+id/editText"

    android:layout_below="@+id/textView"

    android:layout_marginTop="46dp"

    android:text="thanks"

    android:layout_alignParentRight="true"

    android:layout_alignParentEnd="true"

    android:layout_alignParentLeft="true"

    android:layout_alignParentStart="true"

    android:textColor="#ff7aff10"

    android:textColorHint="#ffff23d1" />

    <Button

    android:layout_width="wrap_content"
```




```
android:layout_height="wrap_content"

android:text="CLICK TO CONVERT TEXT TO SPEECH"

android:id="@+id/button"

android:layout_below="@+id/editText"

android:layout_centerHorizontal="true"

android:layout_marginTop="46dp"

android:textSize="15dp" />
```

</RelativeLayout>

Code of toolbar_title_layout.xml file

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

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

    android:layout_height="wrap_content"

    android:orientation="vertical"

    android:gravity="center"

    <TextView

        android:padding="4dp"

        android:layout_width="match_parent"

        android:layout_height="wrap_content"

        android:text="TextToSpeechDemo"

        android:gravity="center"

        android:textSize="16sp"

        android:textStyle="bold"

        android:textColor="@android:color/white"/>

    </LinearLayout>
```



Code of MainActivity.java.

```
package com.example.texttospeech.myapplication;

import android.app.Activity;
import android.os.Bundle;
import android.speech.tts.TextToSpeech;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import java.util.Locale;
import android.widget.Toast;

public class MainActivity extends Activity {

    TextToSpeech t1;
    EditText ed1;
    Button b1;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        getSupportActionBar().setDisplayOptions(ActionBar.DISPLAY_SHOW_CU
        STOM);

        getSupportActionBar().setCustomView(R.layout.toolbar_title_layout);

        setContentView(R.layout.activity_main);

        ed1=(EditText)findViewById(R.id.editText);

        b1=(Button)findViewById(R.id.button);

        t1=new TextToSpeech(getApplicationContext(), new
        TextToSpeech.OnInitListener() {

            @Override

            public void onInit(int status) {
```



```
if(status != TextToSpeech.ERROR) {  
  
    t1.setLanguage(Locale.UK);  
  
    }  
  
    }  
  
});  
  
b1.setOnClickListener(new View.OnClickListener() {  
  
    @Override  
  
    public void onClick(View v) {  
  
        String toSpeak = ed1.getText().toString();  
  
        Toast.makeText(getApplicationContext(),  
toSpeak,Toast.LENGTH_SHORT).show();  
  
        t1.speak(toSpeak, TextToSpeech.QUEUE_FLUSH, null);  
  
        }  
  
    });  
  
}  
  
public void onPause(){  
  
    if(t1 !=null){  
  
        t1.stop();  
  
        t1.shutdown();  
  
        }  
  
        super.onPause();  
  
    }  
  
}
```

6.

Attempt any **TWO** of the following:

12 M

a)

Develop an application to update a record of an employee whose emp.id is 'E101' in SQLite database. Change employee name from "POR" to "XYZ". Also display the updated record (Write .java and .xml files).

6 M



Ans	<p>activity_update_emp.xml file</p> <pre><?xml version="1.0" encoding="utf-8"?> <LinearLayout 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:orientation="vertical" tools:context=".MainActivity"> <!--Edit text to enter employee name--> <EditText android:id="@+id/idEdtEmpName" android:layout_width="match_parent" android:layout_height="wrap_content" android:layout_margin="10dp" android:hint="Enter Employee Name" /> <!--edit text for employee salary--> <EditText android:id="@+id/idEdtEmpSalary" android:layout_width="match_parent" android:layout_height="wrap_content" android:layout_margin="10dp" android:hint="Enter Employee Salary" /> <!--button for adding new employee--> <Button android:id="@+id/idBtnAddCourse" android:layout_width="match_parent"</pre>	<p>activity_update _emp.xmlfile-1 M</p> <p>DBHandler.java file-2 M</p> <p>empRVAdapter.java file-1 M</p> <p>Update java file- 2 M</p>
-----	--	---



```
android:layout_height="wrap_content"  
  
android:layout_margin="10dp"  
  
android:text="Add Employee"  
  
android:textAllCaps="false" />
```

</LinearLayout>

DBHandler.java file

```
import android.content.ContentValues;  
import android.content.Context;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
  
public class DBHandler extends SQLiteOpenHelper {  
    // creating a constant variables for our database.  
    // below variable is for our database name.  
    private static final String DB_NAME = "empdb";  
  
    // below int is our database version  
    private static final int DB_VERSION = 1;  
  
    // below variable is for our table name.  
    private static final String TABLE_NAME = "myemp";  
  
    // below variable is for our id column.  
    private static final String ID_COL = "id";  
  
    // below variable is for our course name column  
    private static final String NAME_COL = "emp_name";
```



```
// below variable is for our employee salary column.

private static final String TRACKS_COL = "emp_salary";


// creating a constructor for our database handler.

public DBHandler(Context context) {

    super(context, DB_NAME, null, DB_VERSION);

}

// below method is for creating a database by running a sqlite query

@Override

public void onCreate(SQLiteDatabase db) {

    // on below line we are creating

    // an sqlite query and we are

    // setting our column names

    // along with their data types.

    String query = "CREATE TABLE " + TABLE_NAME + " ("

        + ID_COL + " INTEGER PRIMARY KEY AUTOINCREMENT, "

        + NAME_COL + " TEXT,"

        + SALARY_COL + " TEXT)";


    // at last we are calling a exec sql

    // method to execute above sql query

    db.execSQL(query);

}


// this method is use to add new EMPLOYEE to our sqlite database.

public void addNewCourse(String empName, String empSalary) {

    // on below line we are creating a variable for
```



```
// our sqlite database and calling writable method

// as we are writing data in our database.

SQLiteDatabase db = this.getWritableDatabase();

// on below line we are creating a

// variable for content values.

ContentValues values = new ContentValues();


// on below line we are passing all values

// along with its key and value pair.

values.put(NAME_COL, empName);

values.put(TRACKS_COL, empSalary);


// after adding all values we are passing

// content values to our table.

db.insert(TABLE_NAME, null, values);


// at last we are closing our

// database after adding database.

db.close();

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

    // this method is called to check if the table exists already.

    db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);

    onCreate(db);

}

}
```



empRVAdapter.java file

```
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    // creating variables for our edittext, button and dbhandler
    private EditText empNameEdt, empSalaryEdt;
    private Button addempBtn;
    private DBHandler dbHandler;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // initializing all our variables.
        empNameEdt = findViewById(R.id.idEdtempName);
        empSalaryEdt = findViewById(R.id.idEdtempSalary);
        addempBtn = findViewById(R.id.idBtnAddemp);
        // creating a new dbhandler class
        // and passing our context to it.
        dbHandler = new DBHandler(MainActivity.this);
        // below line is to add on click listener for our add emp button.
        addCourseBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
```




```
// below line is to get data from all edit text fields.

String empName = empNameEdt.getText().toString();

String empSalary = empSalaryEdt.getText().toString();

// validating if the text fields are empty or not.

if (empName.isEmpty() && empSalary.isEmpty() &&) {

    Toast.makeText(MainActivity.this, "Please enter all the data..",
Toast.LENGTH_SHORT).show();

    return;

}

// on below line we are calling a method to add new

// employee to sqLite data and pass all our values to it.

dbHandler.addNewemp(empName, empSalary);

// after adding the data we are displaying a toast message.

Toast.makeText(MainActivity.this, "Employee has been added.",
Toast.LENGTH_SHORT).show();

empNameEdt.setText("");

empSalaryEdt.setText("");

}

});

}

}

Update employee record java file:

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;
```



```
import androidx.appcompat.app.AppCompatActivity;

public class UpdateCourseActivity extends AppCompatActivity {

    // variables for our edit text, button, strings and dbhandler class.

    private EditText empNameEdt, empSalaryEdt;

    private Button updateempBtn;

    private DBHelper dbHelper;

    String empName, empSalary;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_update_emp);

        // initializing all our variables.

        empNameEdt = findViewById(R.id.idEdtempName);

        empSalaryEdt = findViewById(R.id.idEdtempSalary);

        updateempBtn = findViewById(R.id.idBtnUpdateemp);

        // on below line we are initializing our dbhandler class.

        dbHelper = new DBHelper(UpdateempActivity.this);

        // on below lines we are getting data which

        // we passed in our adapter class.

        empName = getIntent().getStringExtra("emp_name");

        empSalary = getIntent().getStringExtra("emp_salary");

        // setting data to edit text

        // of our update activity.

        empNameEdt.setText(empName);
```



		<pre>empSalaryEdt.setText(empSalary); // adding on click listener to our update course button. updateempBtn.setOnClickListener(new View.OnClickListener() { @Override public void onClick(View v) { // inside this method we are calling an update employee // method and passing all our edit text values. dbHandler.updateemp(empName, empNameEdt.getText().toString(), empSalaryEdt.getText().toString()); // displaying a toast message that our employee database has been updated. Toast.makeText(UpdateempActivity.this, "Employee Record Updated..", Toast.LENGTH_SHORT).show(); // launching our main activity. Intent i = new Intent(UpdateempActivity.this, MainActivity.class); startActivity(i); } }); } }</pre>	
	b)	i) Describe all steps in application deployment on google playstore. ii) Write steps for customized permissions.	6 M
	Ans	a) Application deployment steps: Publishing is the general process that makes your Android app available to users. When you publish an Android app on google play store, you do the following: <ul style="list-style-type: none">• Prepare the app for release.	Application deployment steps:3 M Steps for Customized Permissions:3 M



During the preparation step, you build a release version of your app.

- Release the app to users.

During the release step, you publicize, sell, and distribute the release version of your app, which users can download and install on their Android-powered devices.

Prepare your app for release

1) Preparing your app for release is a multistep process involving the following tasks:

- Configure your app for release.

At a minimum, you need to make sure that logging is disabled and removed and that your release variant has `debuggable false` for Groovy or `isDebuggable = false` for Kotlin script set. You should also set your app's version information.

- Build and sign a release version of your app.

You can use the Gradle build files with the *release* build type to build and sign a release version of your app. For more information, see *Build and run your app*.

- Test the release version of your app.

Before you distribute your app, you should thoroughly test the release version on at least one target handset device and one target tablet device. Firebase Test Lab is useful for testing across a variety of devices and configurations.

- Update app resources for release.

Make sure that all app resources, such as multimedia files and graphics, are updated and included with your app or staged on the proper production servers.

- Prepare remote servers and services that your app depends on.

If your app depends on external servers or services, make sure they are secure and production ready.

2) Signing of Application

- Application signing allows developers to identify the author of the application and to update their application without creating complicated interfaces and permissions.
- Every application that is run on the Android platform must be signed by the developer.



- Applications that attempt to install without being signed will be rejected by either Google Play or the package installer on the Android device.

APK Signing Schemes

- v1 Scheme: based on JAR signing.
- v2 Scheme: APK Signature Scheme v2, which was introduced in Android 7.0.
- v3 Scheme: APK Signature Scheme v3, which was introduced in Android 9.

b) Steps for Customized Permissions:

1)App signing

All APKs must be signed with a certificate whose private key is held by their developer. The certificate does *not* need to be signed by a certificate authority. It's allowable, and typical, for Android apps to use self-signed certificates. The purpose of certificates in Android is to distinguish app authors. This lets the system grant or deny apps access to signature-level permissions and grant or deny an app's request to be given the same Linux identity as another app.

2)Grant signature permissions after device manufacturing time

Starting in Android 12 (API level 31), the `knownCerts` attribute for signature-level permissions lets you refer to the digests of known signing certificates at declaration time.

You can declare the `knownCerts` attribute and use the `knownSigner` flag in your app's `protectionLevel` attribute for a particular signature-level permission. Then, the system grants that permission to a requesting app if any signer in the requesting app's signing lineage, including the current signer, matches one of the digests that's declared with the permission in the `knownCerts` attribute.

The `knownSigner` flag lets devices and apps grant signature permissions to other apps without having to sign the apps at the time of device manufacturing and shipment.

3)User IDs and file access

At install time, Android gives each package a distinct Linux user ID. The identity remains constant for the duration of the package's life on that device. On a different device, the same package might have a different UID—what matters is that each package has a distinct UID on a given device.

Because security enforcement happens at the process level, the code of any two



	<p>packages can't normally run in the same process, since they need to run as different Linux users.</p> <p>Any data stored by an app is assigned that app's user ID and isn't normally accessible to other packages.</p> <p>For example, an app that needs to control which other apps can start one of its activities can declare a permission for this operation as follows:</p> <pre><manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.example.myapplication" > <permission android:name="com.example.myapplication.permission.DEADLY_ACTIVITY" android:label="@string/permlab_deadlyActivity" android:description="@string/permdesc_deadlyActivity" android:permissionGroup="android.permission-group.COST_MONEY" android:protectionLevel="dangerous" /> ... </manifest></pre>	
c)	Develop a program to TURN ON and OFF bluetooth. Write .java file and permission tags.	6 M
Ans	Code of MainActivity.java package com.example.bluetooth.myapplication; import android.app.Activity; import android.bluetooth.BluetoothAdapter; import android.bluetooth.BluetoothDevice; import android.content.Intent; import android.os.Bundle; import android.view.View; import android.widget.AdapterView; import android.widget.Button; import android.widget.ListView; import android.widget.Toast; import java.util.ArrayList; import java.util.Set; public class MainActivity extends Activity { Button b1,b2,b3,b4; private BluetoothAdapter BA; private Set<BluetoothDevice>pairedDevices; ListView lv; 	AndroidManifest.xml file-3 M MainActivity.java file-3 M



```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    b1 = (Button) findViewById(R.id.button);
    b2=(Button)findViewById(R.id.button2);
    b3=(Button)findViewById(R.id.button3);
    b4=(Button)findViewById(R.id.button4);

    BA = BluetoothAdapter.getDefaultAdapter();
    lv = (ListView)findViewById(R.id.listView);
}

public void on(View v){
    if (!BA.isEnabled()) {
        Intent turnOn = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
        startActivityForResult(turnOn, 0);
        Toast.makeText(getApplicationContext(), "Turned
on",Toast.LENGTH_LONG).show();
    } else {
        Toast.makeText(getApplicationContext(), "Already on",
Toast.LENGTH_LONG).show();
    }
}

public void off(View v){
    BA.disable();
    Toast.makeText(getApplicationContext(), "Turned off"
,Toast.LENGTH_LONG).show();
}

public void visible(View v){
    Intent getVisible = new
Intent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);
    startActivityForResult(getVisible, 0);
}

public void list(View v){
    pairedDevices = BA.getBondedDevices();

    ArrayList list = new ArrayList();

    for(BluetoothDevice bt : pairedDevices) list.add(bt.getName());
    Toast.makeText(getApplicationContext(), "Showing Paired
Devices",Toast.LENGTH_SHORT).show();
}
```



```
final      ArrayAdapter      adapter      =      new
ArrayAdapter(this,android.R.layout.simple_list_item_1, list);

    lv.setAdapter(adapter);
}
}
Permission Tags
AndroidManifest.xml file
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.bluetooth.myapplication" >
    <uses-permission android:name="android.permission.BLUETOOTH"/>
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >

        <activity
            android:name=".MainActivity"
            android:label="@string/app_name" >

            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>

        </activity>

    </application>
</manifest>
```