CMR INSTITUTE OF TECHNOLOGY	USN										

Internal Assessment Test –III, October 2024				
Sub:	**Mobile Application Development**	**Code:**	**22MCA263**	
	Answer Key	**Marks**	**OBE**	
			CO	RBT
1	Define Fragment. Explain the life cycle of fragments. A fragment has its own layout and its own behaviour with its own life cycle callbacks. You can add or remove fragments in an activity while the activity is running. You can combine multiple fragments in a single activity to build a multi-pane UI. A fragment can be used in multiple activities. Fragment life cycle is closely related to the life cycle of its host activity which means when the activity is paused, all the fragments available in the activity will also be stopped. A fragment can implement a behaviour that has no user interface component. Fragments were added to the Android API in Honeycomb version of Android which API version 11. Android fragments have their own life cycle very similar to an android activity. This section briefs different stages of its life cycle. **Fragment lifecycle** This involves a number of simple steps to create Fragments. onAttach()The fragment instance is associated with an activity instance.The fragment and the activity is not fully initialized. Typically you get in this method a reference to the activity which uses the fragment for further initialization work. onCreate() The system calls this method when creating the fragment. You should initialize essential components of the fragment that you want to retain when the fragment is paused or stopped, then resumed. onCreateView() The system calls this callback when it's time for the fragment to draw its user interface for the first time. To draw a UI for your fragment, you must return a View component from this method that is the root of your fragment's layout. You can return null if the fragment does not provide a UI. onActivityCreated()The onActivityCreated() is called after the onCreateView() method when the host activity is created. Activity and fragment instance have been created as well as the view hierarchy of the activity. At this point, view can be accessed with the findViewById() method. example. In this method you can instantiate objects which require a Context objectonStart()The onStart() method is called once the fragment gets visible. onResume()Fragment becomes active. onPause() The system calls this method as the first indication that the user is	10	CO3	L3

	leaving the fragment. This is usually where you should commit any changes that should be persisted beyond the current user session. onStop()Fragment going to be stopped by calling onStop() onDestroyView()Fragment view will destroy after call this method onDestroy() onDestroy() called to do final clean up of the fragment's state but Not guaranteed to be called by the Android platform.			
2	Develop a Mobile Application to create a list of places in India using spinner view and display the item selected by the user. activity_main.xml Drag the Spinner from the pallete, now the activity_main.xml file will like this: File: activity_main.xml <?xml version="1.0" encoding="utf-8"?> <android.support.constraint.ConstraintLayout 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="example.javatpoint.com.spinner.MainActivity" > <Spinner android:id="@+id/spinner" android:layout_width="149dp" android:layout_height="40dp" android:layout_marginBottom="8dp" android:layout_marginEnd="8dp" android:layout_marginStart="8dp" android:layout_marginTop="8dp" app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintEnd_toEndOf="parent" app:layout_constraintHorizontal_bias="0.502" app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent" app:layout_constraintVertical_bias="0.498" /> </android.support.constraint.ConstraintLayout> Activity class Let's write the code to display item on the spinner and perform event handling. File: MainActivity.java package example.javatpoint.com.spinner;	10	CO4	L2

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Spinner;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity
implements
      AdapterView.OnItemSelectedListener {
   String[] country = { "India", "USA", "China", "Japan", "Other"};

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
     //Getting the instance of Spinner and applying
OnItemSelectedListener on it
      Spinner spin = (Spinner) findViewById(R.id.spinner);
      spin.setOnItemSelectedListener(this);

      //Creating the ArrayAdapter instance having the country
list
      ArrayAdapter aa = new
ArrayAdapter(this,android.R.layout.simple_spinner_item,cou
ntry);

aa.setDropDownViewResource(android.R.layout.simple_spin
ner_dropdown_item);
      //Setting the ArrayAdapter data on the Spinner
      spin.setAdapter(aa);

   }

   //Performing action onItemSelected and onNothing
selected
   @Override
   public void onItemSelected(AdapterView<?> arg0, View
arg1, int position, long id) {

Toast.makeText(getApplicationContext(),country[position] ,
Toast.LENGTH_LONG).show();
   }
   @Override
   public void onNothingSelected(AdapterView<?> arg0) {
      // TODO Auto-generated method stub
   }
}
```

3	Illustrate with an example how to play audio in an android application. activity_main.xml <RelativeLayout xmlns:androclass="http://schemas.android.com/apk/res/andr oid" xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent" android:layout_height="match_parent" android:paddingBottom="@dimen/activity_vertical_margin" android:paddingLeft="@dimen/activity_horizontal_margin" android:paddingRight="@dimen/activity_horizontal_margin" android:paddingTop="@dimen/activity_vertical_margin" tools:context=".MainActivity" > <TextView android:id="@+id/textView1" android:layout_width="wrap_content" android:layout_height="wrap_content" android:layout_alignParentTop="true" android:layout_marginTop="30dp" android:text="Audio Controller" /> <Button android:id="@+id/button1" style="?android:attr/buttonStyleSmall" android:layout_width="wrap_content" android:layout_height="wrap_content" android:layout_alignLeft="@+id/textView1" android:layout_below="@+id/textView1" android:layout_marginTop="48dp" android:text="start" /> <Button android:id="@+id/button2" style="?android:attr/buttonStyleSmall" android:layout_width="wrap_content" android:layout_height="wrap_content" android:layout_alignTop="@+id/button1" android:layout_toRightOf="@+id/button1"	10	CO3	L3

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

    <Button
        android:id="@+id/button3"
        style="?android:attr/buttonStyleSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignTop="@+id/button2"
        android:layout_toRightOf="@+id/button2"
        android:text="stop" />

</RelativeLayout>
```

File: **MainActivity.java**

```
package com.example.audioplay;

import android.media.MediaPlayer;
import android.os.Bundle;
import android.os.Environment;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class MainActivity extends Activity {
    Button start,pause,stop;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        start=(Button)findViewById(R.id.button1);
        pause=(Button)findViewById(R.id.button2);
        stop=(Button)findViewById(R.id.button3);
        //creating media player
        final MediaPlayer mp=new MediaPlayer();
        try{
            //you can change the path, here path is external
directory(e.g. sdcard) /Music/maine.mp3
```

```
mp.setDataSource(Environment.getExternalStorageDirectory().getPath()+"/Music/maine.mp3");

        mp.prepare();
        }
catch(Exception e)
{
e.printStackTrace();
}

  start.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
          mp.start();
        }
      });
      pause.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
          mp.pause();
        }
      });
      stop.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
          mp.stop();
        }
      });
    }
}
```

4 | What is the use of DatePicker Dialog? Explain with an example. | 10 | CO4 | L3

Android DatePicker Example

Let's see the simple example of datepicker widget in android.

activity_main.xml

File: activity_main.xml

```
<?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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"

tools:context="example.javatpoint.com.datepicker.MainActivi
ty">

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/button1"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginBottom="102dp"
        android:layout_marginLeft="30dp"
        android:layout_marginStart="30dp"
        android:text="" />

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="20dp"
        android:text="Change Date" />

    <DatePicker
        android:id="@+id/datePicker"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/textView1"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="36dp" />

</RelativeLayout>
```

Activity class

File: MainActivity.java

```
package example.javatpoint.com.datepicker;
```

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    DatePicker picker;
    Button displayDate;
    TextView textview1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        textview1=(TextView)findViewById(R.id.textView1);
        picker=(DatePicker)findViewById(R.id.datePicker);
        displayDate=(Button)findViewById(R.id.button1);

        textview1.setText("Current Date: "+getCurrentDate());

        displayDate.setOnClickListener(new
View.OnClickListener(){
            @Override
            public void onClick(View view) {

                textview1.setText("Change Date: "+getCurrentDate());
            }

        });

    }
    public String getCurrentDate(){
        StringBuilder builder=new StringBuilder();;
        builder.append((picker.getMonth() + 1)+"/");//month is 0
based
        builder.append(picker.getDayOfMonth()+"/");
        builder.append(picker.getYear());
        return builder.toString();
    }
}
```

5	What is a service? What are the different methods used to create a service? Explain with an example.
	
	CREATING YOUR OWN SERVICES
	A service is an application in Android that runs in the background without needing to interact with the user. For example, while using an application, you may want to play some background music at the same time. In this case, the code that is playing the background music has no need to interact with the user, and hence it can be run as a service. Services are also ideal for situations in which there is no need to present a UI to the user.
	
	Following are the steps involved in creating own service.
	
	• Create a new android application.
	• Add a new class file to the project and name it MyService.java. Write the following code in it:
	package net.learn2develop.Services; import
	android.app.Service;
	import android.content.Intent; import
	android.os.IBinder; import
	android.widget.Toast;
	
	public class MyService extends Service
	{
	@Override
	public IBinder onBind(Intent arg0)
	{
	return null;
	}
	
	public int onStartCommand(Intent intent, int flags, int startId)
	{
	Toast.makeText(this,"ServiceStarted",Toast.LENGTH_LONG).show();
	return START_STICKY;
	}
	
	public void onDestroy()
	{
	super.onDestroy();
	Toast.makeText(this,"ServiceDestroyed",Toast.LENGTH_LONG).show();
	}
	}
	
	The onBind()method enables you to bind an activity to a service. This in turn enables an activity to directly access members and methods inside a service. The onStartCommand() method is called when you start the service explicitly using the startService() method. This method signifies the start of the service, and you code it
		10	CO3	L3

to do the things you need to do for your service. In this method, you returned the constant START_STICKY so that the service will continue to run until it is explicitly stopped. The onDestroy() method is called when the service is stopped using the stopService() method. This is where you clean up the resources used by your service.

- In the AndroidManifest.xml file, add the following statement :

```
<service android:name=".MyService" />
```

- In the activity_main.xml file, add the following statements in bold:

```
<Button android:id="@+id/btnStartService"
android:layout_width="fill_parent"
android:layout_height="wrap_content" android:text="Start
Service" />
<Button android:id="@+id/btnStopService"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="Stop Service" />
```

- Add the following statements in bold to the MainActivity.java file:

```
import android.content.Intent; import
android.view.View; import
android.widget.Button;

public class MainActivity extends Activity
{
    public void onCreate(Bundle savedInstanceState)
    {
     super.onCreate(savedInstanceState);
     setContentView(R.layout.main);

     Button btnStart = (Button) findViewById(R.id.btnStartService);

     btnStart.setOnClickListener(new View.OnClickListener()
     {
            public void onClick(View v)
            {
        startService(new Intent(getBaseContext(), MyService.class));
            }
     });


  Button btnStop = (Button) findViewById(R.id.btnStopService);
  btnStop.setOnClickListener(new View.OnClickListener()
  {
      public void onClick(View v)
      {
          stopService(new Intent(getBaseContext(), MyService.class));
       }
    });
 }
}
```

	• Run the application on the emulator and the output screen will be as shown –  			
6	Explain AsyncTask class with examples. The AsyncTask class not only creates the Thread but manages it and creates an asynchronous task for performing the processing in the background. The AsyncTask class provides us with event handlers that synchronize with the Thread to show the progress and completion of the task. To instantiate an AsyncTask, we extend the AsyncTask class and provide three parameters: Input Parameters, Progress Values, and Result Values. If we don't want to provide any of the parameters, simply replace it with void. The subclass should also override the following event handlers: > doInBackground—This method is executed in the background thread. The code that doesn't interact with the user is placed in this handler. Input Parameters is passed to this method as input. From within this method, the publishProgress() method is called, and the onProgressUpdate() method is executed in the main thread. When we use the publishProgress() and onProgressUpdate() methods, the background thread communicates with the main thread to update the UI elements to indicate progress of the work. Remember, the code in this method runs in a separate background thread. > onProgressUpdate—Override this handler to update the UI to indicate progress in the task. This handler receives the set of parameters passed in to publishProgress (). This handler is synchronized with the thread when executed, so you can safely modify UI elements. > onPostExecute—As the name suggests, this method is called after the doInBackground() method is complete. The Result values returned by the doInBackground () method are	10	CO3	L3

passed in to this event handler. This handler can be used to indicate when the asynchronous task is complete.

The AsyncTask Class also provides the following two helpful callback methods:

> onPreExecute—From the name of the method, it is clear that it is called before the doInBackground() method is called. This method runs in the main thread, and setup or similar tasks are performed in this method.

> onCancelled—Manages the cancellation of the thread. The method interrupts the execution of the thread and also prevents execution of the onPostExecute () method.

LISTING 13.3 Code in the `main.xml` Layout File

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TextView
        android:id="@+id/sequenceview"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
</LinearLayout>
```

LISTING 13.4 Code in the `AsyncAppActivity.java` Java Activity File

```
package com.androidunleashed.asyncapp;

import android.app.Activity;
import android.os.Bundle;
```

```
import android.widget.TextView;
import android.os.AsyncTask;

public class AsyncAppActivity extends Activity {
    TextView seqView;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_async_app);
        seqView=(TextView) findViewById(R.id.sequenceview);
        new PrintSequenceTask().execute(1);
    }

    private class PrintSequenceTask extends AsyncTask<Integer, Integer, Void> {
        @Override
        protected void onPreExecute() {
            seqView.setText("Sequence numbers begins");
        }

        @Override
        protected Void doInBackground(Integer... args) {
            for (int i = args[0]; i <= 10; i++) {
                publishProgress(i);
                try {
                    Thread.sleep(1000);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
            return null;
        }

        @Override
        protected void onProgressUpdate(Integer... args) {
            seqView.setText(args[0].toString());
        }

        @Override
        protected void onPostExecute(Void result) {
            seqView.setText("Sequence numbers over");
        }
    }
}
```

7	What is a content provider? Explain how to create a custom content provider. **What Is a Content Provider?** A content provider acts as a data store and provides an interface to access its contents. Unlike a database, where information can be accessed only by the package in which it was created, information in a content provider can be shared across packages. The following lists a few characteristics of content providers: > Like in a database, we can query, add, edit, delete, and update data in content providers. » Data can be stored in a database, files, and over a network.	10	CO3	L2

>» A content provider acts as a wrapper around the data store to make it resemble web services. That is, the data in content providers is exposed as a service.

Create Custom Content Provider:-

These are the steps taken to create our own content provider:

>» Define the content provider—the class that extends the android.content. Content Provider Class.

> Define the database, URIs, column names, MIME types, and so on.

> Implement the query insert, update, delete, and getType methods to make the content provider functional.

> Register the provider in the manifest file.

Defining a Content Provider

To define our own content provider, we need to create a class that extends the android.content .Content Provider Class. Let's create a new Android project called CreateContentProviderApp. This application provides the STD (subscriber trunk dialing) code of the specified country. The application allows users to enter the STD codes of different countries, display a scrollable list of existing countries, and delete and update STD code information.

Defining a Database

The content provider that we want to create provides information about the countries and their corresponding STD codes. To store country names and their corresponding STD codes, we create a database called countries. In the Countries database, we create a single table called stdinfo. The stdinfo table consists of two columns, country and stdcode, that store the country names and their respective STD codes. The following constants represent the database, its table, and columns:

```
static final String DB_NAME = "Countries.db";

static final String DB_TABLE = "stdinfo";

static final int DB_VERSION = 1;

static final String CREATE_TABLE ="CREATE TABLE " + DB_TABLE +" (_id INTEGER PRIMARY KEY AUTOINCREMENT, country TEXT not null, stdcode TEXT not null);";

static final String ID = "_id";

static final String COUNTRY = "country";

static final String STDCODE = "stdcode";
```

Defining the Content URI

To fetch data from the content provider, a suitable URI is required.

```
static final String AUTHORITY="com.bmharwani.provider.Countries";
static final Uri CONTENT_URI =Uri.parse("content://"+AUTHORITY+"/stdinfo");
static final int ALLROWS = 1;
static final int SINGLEROW = 2;
private static final UriMatcher URIMATCHER;
static{
    URIMATCHER = new UriMatcher(UriMatcher.NO_MATCH);
    URIMATCHER.addURI(AUTHORITY, "stdinfo", ALLROWS);
    URIMATCHER.addURI(AUTHORITY, "stdinfo/#", SINGLEROW);
}
```

URI Matcher analyzes the form of a URI and determines whether the URI is a request for all rows or a single row

Defining MIME Types

After we define the content URI, MIME types are defined for a single row or collection of rows. The provider implementation uses these constants to return the MIME types for the incoming URIs. A content provider returns the MIME type of the data it is returning. The MIME type is either a single row or all the rows for the given URI.

For a single row, the MIME type is

vnd.android.cursor.item/vnd.company_name.content_ type

For a collection of rows, the MIME type is

vnd.android.cursor.dir/vnd.company_name.content_ type

Implementing the getType, query, insert, update, and delete Methods

To make the content provider functional, we need to implement the getType(), query(), insert (), update(), and delete() methods.

To identify the data type of the content provider, the getType () method is overridden.

```
public String getType(Uri uri) {
```

```
    switch (URIMATCHER.match(uri)){
      case ALLROWS:
        return "vnd.android.cursor.dir/vnd.countries.stdinfo";
      case SINGLEROW:
        return "vnd.android.cursor.item/vnd.countries.stdinfo";
      default:
        throw new IllegalArgumentException("Unsupported URI: " + uri);
    }
  }
```

To enable users to query the content provider for the desired row(s), the query() method is overridden. The method definition for the query () method is

```
public Cursor query(Uri uri, String[] projection, String criteria, String[]
criteriaValues, String sortColumn) {
    SQLiteQueryBuilder queryBuilder = new SQLiteQueryBuilder();
    queryBuilder.setTables(DB_TABLE);
    if (URIMATCHER.match(uri) == SINGLEROW)
      queryBuilder.appendWhere(ID + " = " + uri.getPathSegments().get(1));
    if (sortColumn==null || sortColumn=="")
      sortColumn = "country";
    Cursor c =
queryBuilder.query(CountriesDB,projection,criteria,criteriaValues,null,null,sortCol
umn);
    c.setNotificationUri(getContext().getContentResolver(), uri);
    return c;
  }
```

To insert a new row into the content provider, the insert () method is overridden. A ContentValues object by name contentValues is passed, containing the information for the new row, as shown here:

```
public Uri insert(Uri uri, ContentValues contentValues) {
    long rowID = CountriesDB.insert(DB_TABLE,null,contentValues);
    if (rowID >0) {
       Uri _uri = ContentUris.withAppendedId(CONTENT_URI, rowID);
       getContext().getContentResolver().notifyChange(_uri, null);
       return _uri;
    }
    throw new SQLException("Error: New row could not be inserted ");
  }
```

The update () method is overridden to update the information in the content provider.

```
public int update(Uri uri, ContentValues contentValues, String criteria, String[]
criteriaValues) {
     int count = 0;
     switch (URIMATCHER.match(uri)){
        case ALLROWS:
           count =
CountriesDB.update(DB_TABLE,contentValues,criteria,criteriaValues);
           break;
        case SINGLEROW:
           count = CountriesDB.update(DB_TABLE, contentValues, ID + " = " +
uri.getPathSegments().get(1) +(!    TextUtils.isEmpty(criteria) ? " AND (" +criteria +
')': ""),criteriaValues);
           break;
        default: throw new IllegalArgumentException("URI not found: " + uri);
     }
     getContext().getContentResolver().notifyChange(uri, null);
     return count;
  }
```

The delete () method is overridden to delete information in the content provider.

```
public int delete(Uri rowUri, String criteria, String[] criteriaValues) {
      int count=0;
      switch (URIMATCHER.match(rowUri)){
         case ALLROWS:
            count = CountriesDB.delete(DB_TABLE, criteria, criteriaValues);
            break;
         case SINGLEROW:
            String id = rowUri.getPathSegments().get(1);
            count = CountriesDB.delete(DB_TABLE, ID + " = " + id
+(!TextUtils.isEmpty(criteria) ? " AND (" +criteria + ')': ""),criteriaValues);
            break;
         default: throw new IllegalArgumentException("URI not found: " + rowUri);
      }
      getContext().getContentResolver().notifyChange(rowUri, null);
      return count;
   }
```

Registering Content Providers

The complete provider tag used to register our content provider is

```
<provider android:name="CountriesProvider"
android:authorities="com.bmharwani.provider.Countries">
      </provider>
```

8	Explain the process of sending an email with an example? **activit_main.xml**	10	CO4	L4

```
<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:paddingBottom="@dimen/activity_vertical_margin"
   android:paddingLeft="@dimen/activity_horizontal_margin"
   android:paddingRight="@dimen/activity_horizontal_margin"
   android:paddingTop="@dimen/activity_vertical_margin"
   tools:context="com.example.m5_emails.MainActivity" >

   <Button
      android:id="@+id/btnSendEmail"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentLeft="true"
      android:layout_alignParentTop="true"
      android:layout_marginLeft="100dp"
      android:layout_marginTop="140dp"
      android:text="send email" />

</RelativeLayout>
```

MainActivity.java

```
package com.example.m5_emails;

import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;


public class MainActivity extends Activity {

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
   }
public void onClick(View v)
{
```

	String[] to= {"daya.thummar@gmail.com"};
	String[] cc= {"daya.thummar@gmail.com"};
	sendEmail(to,cc,"hello","how r u");
	
	}
	private void sendEmail(String[] emailAddresses, String[] carbonCopies, String subject, String message) {
	// TODO Auto-generated method stub
	Intent emailIntent = new Intent(Intent.ACTION_SEND);
	emailIntent.setData(Uri.parse("mailto:"));
	String[] to=emailAddresses;
	String[] cc=carbonCopies;
	emailIntent.putExtra(Intent.EXTRA_EMAIL,to);
	emailIntent.putExtra(Intent.EXTRA_CC,cc);
	emailIntent.putExtra(Intent.EXTRA_SUBJECT,subject);
	emailIntent.putExtra(Intent.EXTRA_TEXT, message);
	emailIntent.setType("message/rfc822");
	startActivity(Intent.createChooser(emailIntent, "Email"));
	
	}			

9	Develop a mobile application to display web pages.
	Activity_main.xml
	
	<WebView
	android:id="@+id/webview"
	android:layout_width="match_parent"
	android:layout_height="match_parent"/>
	
	MainActivity.java
	
	import android.webkit.WebSettings;
	import android.webkit.WebView;
	
	public class MainActivity extends Activity {
	
	private WebView webView;
	@Override
	protected void onCreate(Bundle savedInstanceState) {
	super.onCreate(savedInstanceState);
	setContentView(R.layout.activity_main);
	webView = (WebView)findViewById(R.id.webview);
	
	WebSettings webSettings = webView.getSettings();
	webSettings.setBuiltInZoomControls(true);
		10	CO4	L3

	// Load a URL
	webView.loadUrl("http://www.google.com");
	// webView.loadData("<html><body>Hello, WebView!</body></html>", "text/html", "UTF-8");
	// webView.loadUrl("file:///android_asset/localfile.html");
	}
	@Override
	public void onBackPressed() {
	// Check if there's a history to go back to
	if (webView.canGoBack()) {
	webView.goBack();
	} else {
	super.onBackPressed();
	}
	}
	
	AndroidManifest.xml
	<uses-permission android:name="android.permission.INTERNET" />			

10	Develop a mobile application to send and receive SMS. **1. Sending SMS** ***activity.xml*** <EditText android:id="@+id/editText1" android:layout_width="wrap_content" android:layout_height="wrap_content" android:layout_alignParentRight="true" android:layout_alignParentTop="true" android:layout_marginRight="20dp" android:ems="10" /> <EditText android:id="@+id/editText2" android:layout_width="wrap_content" android:layout_height="wrap_content" android:layout_alignLeft="@+id/editText1" android:layout_below="@+id/editText1" android:layout_marginTop="26dp" android:ems="10" android:inputType="textMultiLine" /> <TextView android:id="@+id/textView1" android:layout_width="wrap_content" android:layout_height="wrap_content"	10	CO4	L4

```
        android:layout_alignBaseline="@+id/editText1"
        android:layout_alignBottom="@+id/editText1"
        android:layout_toLeftOf="@+id/editText1"
        android:text="Mobile No:" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/editText2"
        android:layout_alignBottom="@+id/editText2"
        android:layout_alignLeft="@+id/textView1"
        android:text="Message:" />

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/editText2"
        android:layout_below="@+id/editText2"
        android:layout_marginLeft="34dp"
        android:layout_marginTop="48dp"
        android:text="Send SMS" />
```

MainActivity.java

```
import android.telephony.SmsManager;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import android.view.View.OnClickListener;

public class MainActivity extends Activity {

        EditText mobileno,message;
         Button sendsms;

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

        mobileno=(EditText)findViewById(R.id.editText1);
```

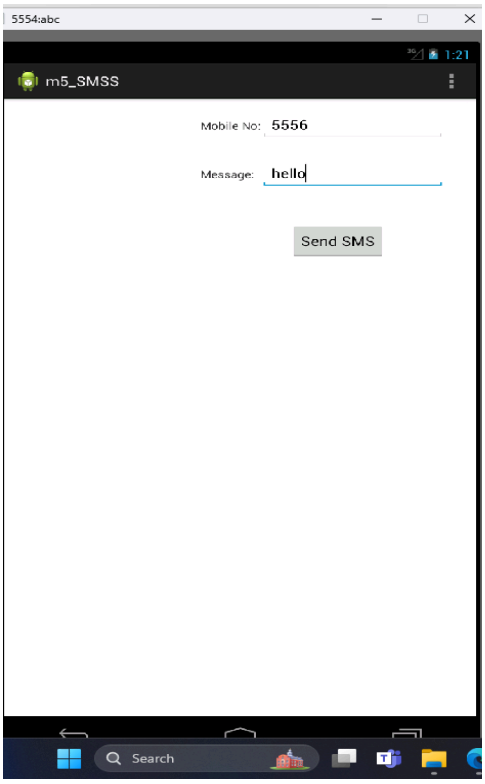
```
        message=(EditText)findViewById(R.id.editText2);
        sendsms=(Button)findViewById(R.id.button1);

    //Performing action on button click
        sendsms.setOnClickListener(new OnClickListener() {

            public void onClick(View arg0) {
                String no=mobileno.getText().toString();
                String msg=message.getText().toString();


                 SmsManager sms=SmsManager.getDefault();
                sms.sendTextMessage(no, null, msg, null, null);

                Toast.makeText(getApplicationContext(), "Message Sent
successfully!",
                    Toast.LENGTH_LONG).show();
            }
        });
    }
```

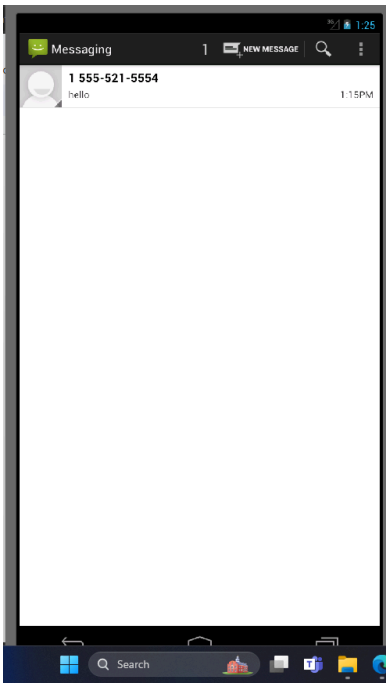
AndroidManifest.xml

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

How to execute

On the first android emulator(5554), click the send SMS button to send an SMS message to the second emulator(5556), to show the SMS message received by the second emulator.





2. Receiving SMS Messages

SMSReceiver.java

```
import android.app.Activity;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.Menu;
```

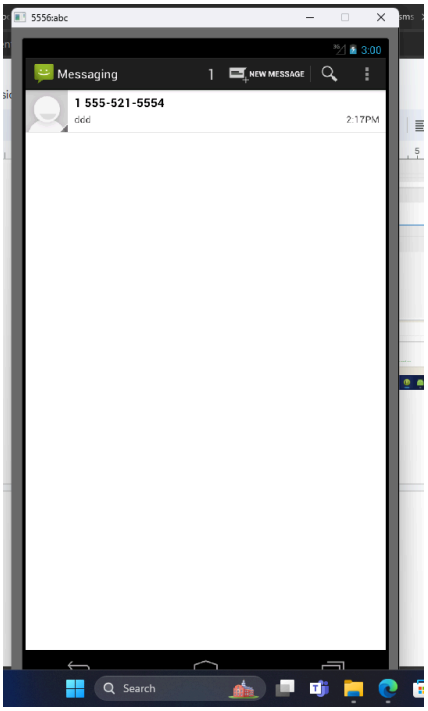
```
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import android.view.View.OnClickListener;


public class MainActivity extends Activity {

          EditText mobileno,message;
            Button sendsms;


   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);
      sendsms=(Button)findViewById(R.id.button1);

   //Performing action on button click
      sendsms.setOnClickListener(new OnClickListener() {

         public void onClick(View arg0) {
            String no=mobileno.getText().toString();
            String msg=message.getText().toString();

            SmsManager sms=SmsManager.getDefault();
            sms.sendTextMessage(no, null, msg, null, null);

            Toast.makeText(getApplicationContext(), "Message Sent
successfully!",
                Toast.LENGTH_LONG).show();
         }
      });
   }
```

Note:- FIRST Run Emulator

Open DDMS goto windows-> open perspective->DDMS

Send msg by using DDMs

Incoming number 5554


5556:abc
3:00
Messaging
1
NEW MESSAGE
1 555-521-5554
ddd
2:17PM
Search