CMR
INSTITUTE OF
TECHNOLOGY

USN

Internal Assessment Test 2– August. 2023

Sub:	**Mobile Applications**							**Sub Code:**	**22MCA263**
Date:	**30/08/2023**	**Duration:**	**90 min's**	**Max Marks:**	**50**	**Sem:**	**II**	**Branch:**	**MCA**

Note : Answer FIVE FULL Questions, choosing ONE full question from each Module

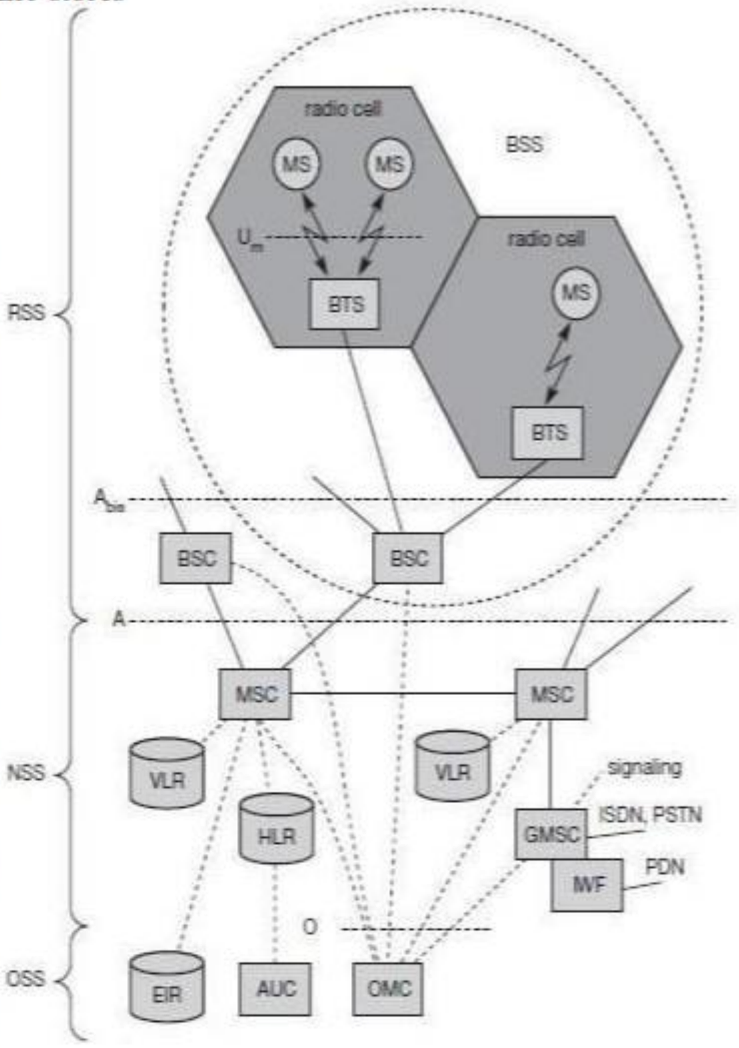
		MARKS	OBE	
			CO	RBT
	PART I			
1	Explain GSM Architecture with a neat diagram.	[10]	CO1 CO2	L3
	OR			
2	Discuss Radio Interface of GSM.	[10]	CO1 CO2	L2
	PART II			
3	What are the different protocols used in GSM?	[10]	CO1 CO2	L2
	OR			
4	Write a program to draw different shapes (oval, rectangle, line) using Bitmap.	[10]	CO3 CO4	L4
	PART III			
5	Write a program to demonstrate the basics of drawing a shape with Java and the Graphics API.	[10]	CO1 CO2	L4
	OR			
6	What is Handover? Explain different possible Handover scenarios of GSM?	[10]	CO1 CO2	L3
	PART IV			
7	Develop an application to list the tourist places of Karnataka using List View.	[10]	CO3 CO4	L4
	OR			
8	Explain the Life Cycle of an Activity?	[10]	CO1 CO2	L1 L2
	PARTV			
9	Design UI for Resume with different view components.	[10]	CO4	L3
	OR			
10	Develop a standard calculator application to perform basic calculations like addition, subtraction, multiplication and division.	[10]	CO3 CO4	L4

Q1) Explain GSM Architecture with a neat diagram.

A GSM system consists of three subsystems, the radio sub system (RSS), the network and switching subsystem (NSS), and the operation subsystem (OSS).

Network Switching Subsystem: The NSS is responsible for performing call processing and subscriber related functions. The switching system includes the following functional units:

- Home location register (HLR): The HLR has a database that used for storage and management of subscriptions. HLR stores all the relevant subscriber data including a subscribers service profile such as call forwarding, roaming, location information and activity status.
- Visitor location register (VLR): It is a dynamic real-time database that stores both permanent and temporary subscribers data which is required for communication b/w the coverage area of MSC and VLR.
- Authentication center (AUC): A unit called the AUC provides authentication and encryption parameters that verify the users identity and ensure the confidentiality of each call.
- Equipment identity register (EIR): It is a database that contains information about the identity of mobile equipment that prevents calls from stolen, unauthorized or defective mobile stations.
- Mobile switching center (MSC): The MSC performs the telephony switching functions of the system. It has various other functions such as 1. Processing of signal. 2. Control calls to and from other telephone and data systems. 3. Call changing, multi-way calling, call forwarding, and other supplementary services. 4. Establishing and terminating the connection b/w MS and a fixed line phone via GMSC.



Radio Subsystem (RSS): The **radio subsystem (RSS)** comprises all radio specific entities, i.e., the **mobile stations (MS)** and the **base station subsystem (BSS)**. The figure shows the connection between the RSS and the NSS via the A **interface** (solid lines) and the connection to the OSS via the **O interface** (dashed lines).

- Base station subsystem (BSS): A GSM network comprises many BSSs, each controlled by a base station controller (BSC). The BSS performs all functions necessary to maintain radio connections to an MS, coding/decoding of voice, and rate adaptation to/from the wireless network part. Besides a BSC, the BSS contains several BTSs.
- Base station controllers (BSC): The BSC provides all the control functions and physical links between the MSC and BTS. It is a high capacity switch that provides functions such as handover, cell configuration data, and control of radio frequency (RF) power levels in BT". A number of BSC's are served by and MSC.
- Basetransceiver station(BTS):The BTS handles the radio interface to the mobile station. A BTS can form a radio cell or, using sectorized antennas, several and is connected to MS via the Um interface, and to the BSC via the A BTS interface. The Um interface contains all the mechanisms necessary for wireless transmission (TDMA, FDMA etc.). The BTS is the radio equipment (transceivers and antennas) needed to service each cell in the network. A group of BTS's are controlled by an BSC.

Operation Service Subsystem (OSS): The OSS facilitates the operations of MSCs. The OSS also handles the Operation and maintenance (OMC) of the entire network.

Operation and Maintenance Centre (OMC): An OMC monitors and controls all other network entities through the 0 interface. The OMC also includes management of status reports, traffic monitoring, subscriber security management, and accounting and billing. The purpose of OSS is to offer the customer cost-effective support for centralized, regional and local operational and maintenance activities that are required for a GSM network. OSS provides a network overview and allows engineers to monitor, diagnose and troubleshoot every aspect of the GSM network.

The **mobile station (MS)** consists of the mobile equipment (the terminal) and a smart card called the Subscriber Identity Module (SIM). The SIM provides personal mobility, so that the user can have access to subscribed services irrespective of a specific terminal. By inserting the SIM card into another GSM terminal, the user is able to receive calls at that terminal, make calls from that terminal, and receive other subscribed services.

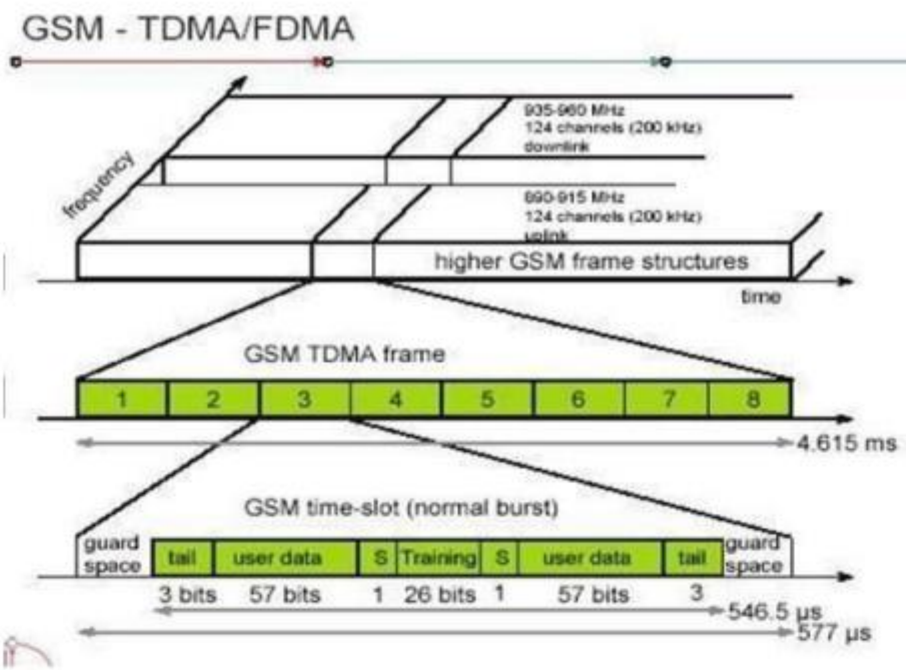
The mobile equipment is uniquely identified by the International Mobile Equipment Identity (IMEI). The SIM card contains the International Mobile Subscriber Identity (IMSI) used to identify the subscriber to the system, a secret key for authentication, and other information. The IMEI and the IMSI are independent, thereby allowing personal mobility. The SIM card may be protected against unauthorized use by a password or personal identity number.

Q2) Discuss Radio Interface of GSM

Radio Interface The most interesting interface in a GSM system is the radio interface, as it contains various multiplexing and media access mechanisms.

Electric signals are given to antenna. The antenna radiates the electromagnetic waves. Electromagnetic waves propagate b/w the transmitter and receiver. Two electrical signals of two sources are not have same frequency at the same time. GSM TDMA Frame, Slots and Bursts

In the below figure, the GSM implements SDMA using cells with BTS and assigns an MS to a BTS. The diagram shows GSM TDMA frame. A frame is again subdivided into 8 GSM time slots, where each slot represents a physical TDM channel and lasts for 577 µs. Each TDM channel occupies the 200 kHz carrier for 577 µs every 4.615 ms. Data is transmitted in small portions, called bursts. As shown, the burst is only 546.5 µs long and contains 148 bits. The remaining 30.5 µs are used as guard space to avoid overlapping with other bursts due to different path delays and to give the transmitter time to turn on and off.



The first and last three bits of a normal burst (**tail**) are all set to 0 and can be used to enhance the receiver performance. The **training** sequence in the middle of a slot is used to adapt the parameters ofthe receiver to thecurrent pathpropagation characteristics and to select thestrongest signal in case ofmulti-pathpropagation. Aflag **S** indicateswhetherthe **data** field contains user or network control data.

Apart from the normal burst, ETSI (1993a) defines four more bursts for data transmission: a **frequency correction** burst allows the MS to correct the localoscillator to avoid interference with neighbouring channels, a **synchronization burst** with an extended training sequencesynchronizes the MS withthe BTS in time, an **access burst** is used for the initialconnectionsetupbetweenMS and BTS, andfinallya**dummyburst** isused ifno datais available for aslot.

Physical, logical channels and frame hierarchy: Twotypesofchannels,namelyphysicalchannels andlogicalchannelsarepresent.

Physical channel: channel defined by specifying both, a carrier frequency and a TDMA timeslot number.

Logic channel: logical channels are multiplexed into the physical channels. Each logic channel performs a specific task. Consequently the data of a logical channel is transmitted in the corresponding timeslots of the physical channel. During this process, logical channels can occupy a part of the physical channel or even the entire channel.

Q3) What are the different protocols used in GSM?

GSM Protocols: The signaling protocol in GSM is structured into three general layers depending on the interface, as shown below.
Layer 1 is the physical layer that handles all radio-specific functions. This includes the creation of bursts according to the five different formats, multiplexing of bursts into a TDMA frame, synchronization with the BTS, detection of idle channels, and measurement of the channel quality on the downlink.
The main tasks of the physical layer contain channel coding and error detection/ correction, which are directly combined with the coding mechanisms. Channel coding using different forward error correction (FEC) schemes.

Signaling between entities in a GSM network requires higher layers. For this purpose, the **LAPDm** protocol has been defined at the Um interface for **layer two**. LAPDm has been derived from link access procedure for the D-channel (**LAPD**) in ISDN systems, which is a version of HDLC.

The **network layer** in GSM contains several sub-layers. The lowest sub-layer is the radio resource management (RR). The functions of RR' are supported by the BSC via the BTS management (BTSM). The maintasksofRRare setup, maintenance, and release of radio channels. Mobility management (MM) contains functions for registration, authentication, identification, location updating, and the provision of a temporary mobile subscriber identity (TMSI).

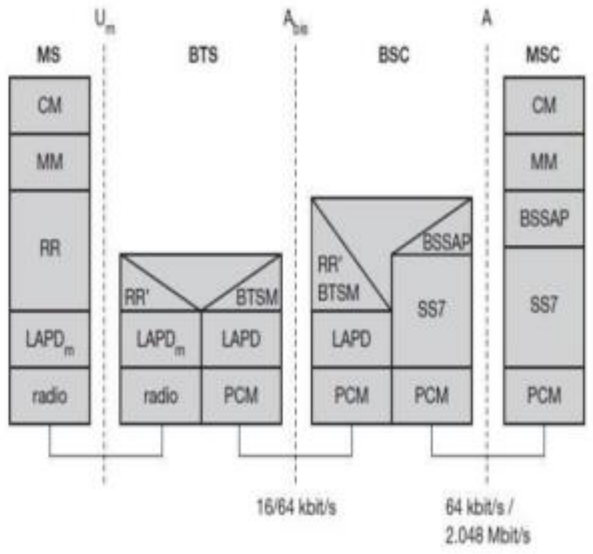
Finally, the call management (CM) layer contains three entities: call **control (CC), short message service (SMS**), and **supplementary service (SS).**

SMS allows for message transfer using the control channels SDCCH and SACCH, while SS offers the services like user identification, call redirection, or forwarding of ongoing calls.

CC provides a point-to-point connection between two terminals and is used by higher layers for call establishment, call clearing and change of call parameters.

Data transmission at the physical layer typically uses pulse code modulation (PCM) systems. LAPD is used for layer two at Abis, BTSM for BTS management.

Signaling system No. 7 (SS7) is used for signaling between an MSC and a BSC. This protocol also transfers all management information between MSCs, HLR, VLRs, AuC, EIR, and OMC. An MSC can also control a BSS via a BSS application part (**BSSAP**).



Q4) Write a program to draw different shapes (oval, rectangle, line) using Bitmap.

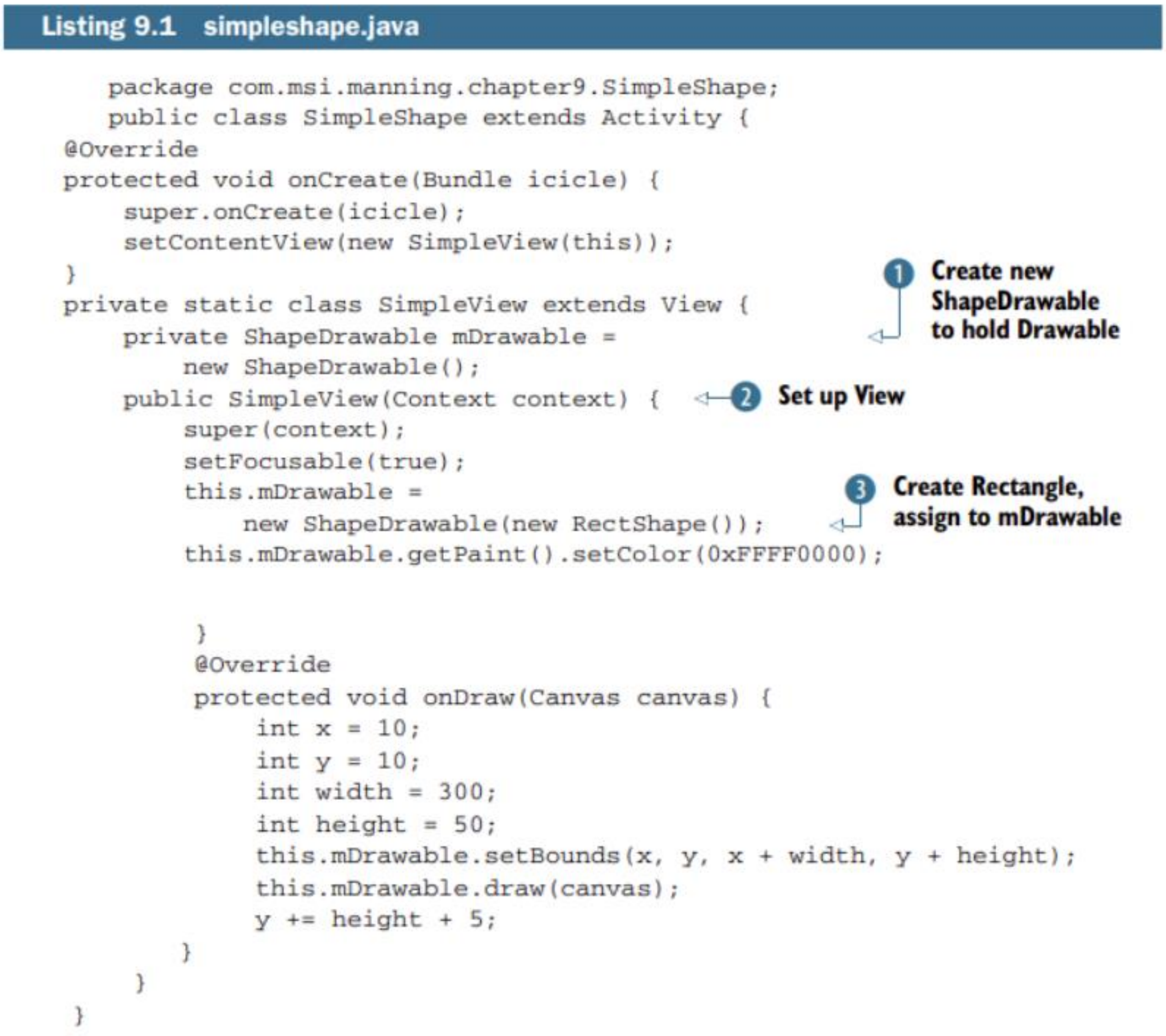
Activity_main.xml:

```
<?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">

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/imageView" />
</RelativeLayout>
```

MainActivity.java:

```
package com.example.ashwini.program4;

import android.app.Activity;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.drawable.BitmapDrawable;
import android.os.Bundle;
import android.widget.ImageView;

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

        //Creating a Bitmap
        Bitmap bg = Bitmap.createBitmap(720, 1280, Bitmap.Config.ARGB_8888);

        //Setting the Bitmap as background for the ImageView
        ImageView i = (ImageView) findViewById(R.id.imageView);
        i.setBackgroundDrawable(new BitmapDrawable(bg));

        //Creating the Canvas Object
        Canvas canvas = new Canvas(bg);



        //Creating the Paint Object and set its color & TextSize
        Paint paint = new Paint();
        paint.setColor(Color.BLUE);
        paint.setTextSize(50);

        //To draw a Rectangle
        canvas.drawText("Rectangle", 420, 150, paint);
        canvas.drawRect(400, 200, 650, 700, paint);

        //To draw a Circle
        canvas.drawText("Circle", 120, 150, paint);
        canvas.drawCircle(200, 350, 150, paint);

    }
}
```

Q5) Write a program to demonstrate the basics of drawing a shape with Java and the Graphics API.



Listing 9.1 simpleshape.java

```
   package com.msi.manning.chapter9.SimpleShape;
   public class SimpleShape extends Activity {
@Override
protected void onCreate(Bundle icicle) {
    super.onCreate(icicle);
    setContentView(new SimpleView(this));
}
private static class SimpleView extends View {
    private ShapeDrawable mDrawable =
        new ShapeDrawable();
    public SimpleView(Context context) {
        super(context);
        setFocusable(true);
        this.mDrawable =
            new ShapeDrawable(new RectShape());
        this.mDrawable.getPaint().setColor(0xFFFF0000);


        }
        @Override
        protected void onDraw(Canvas canvas) {
            int x = 10;
            int y = 10;
            int width = 300;
            int height = 50;
            this.mDrawable.setBounds(x, y, x + width, y + height);
            this.mDrawable.draw(canvas);
            y += height + 5;
        }
     }
  }
```

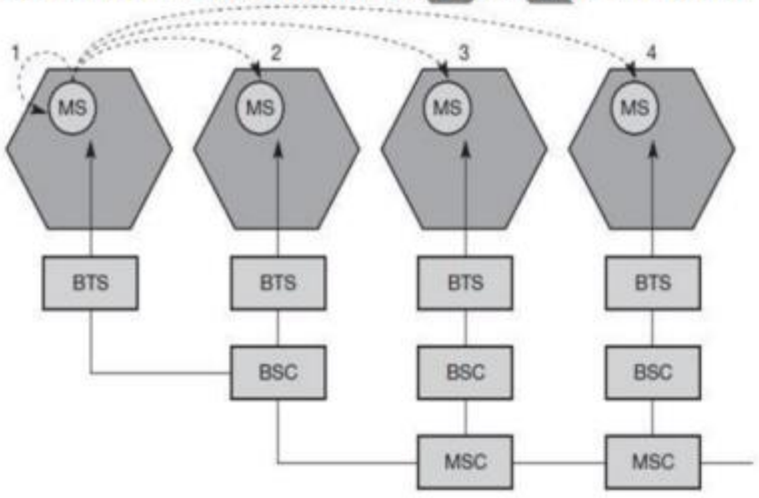
1 Create new ShapeDrawable to hold Drawable
2 Set up View
3 Create Rectangle, assign to mDrawable

Q6) What is Handover? Explain different possible Handover scenarios of GSM?

Handover: Cellular systems require handover procedures, as single cells do not cover the whole service area. However, a handover should not cause a cut-off, also called call drop.

GSM aims at maximum handover duration of 60 ms. There are two basic reasons for a handover:

1. The mobile station moves out of the range of a BTS, decreasing the received signal level increasing the error rate thereby diminishing the quality of the radio link.

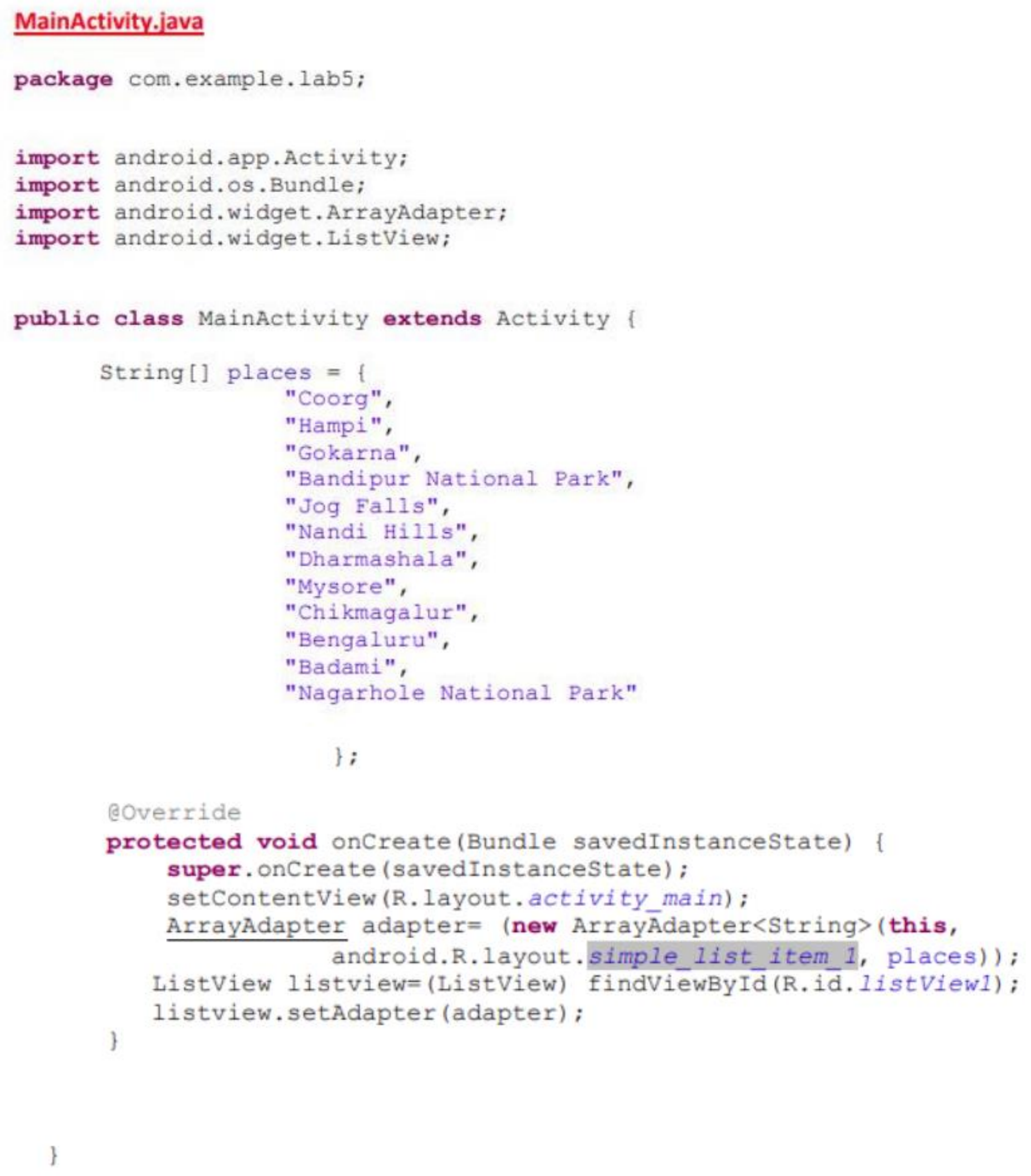


2. Handover may be due to load balancing, when an MSC/BSC decides the traffic is too high in one cell
   and shifts some MS to other cells with a lower load.

The four possible handover scenarios of GSM are shown below:

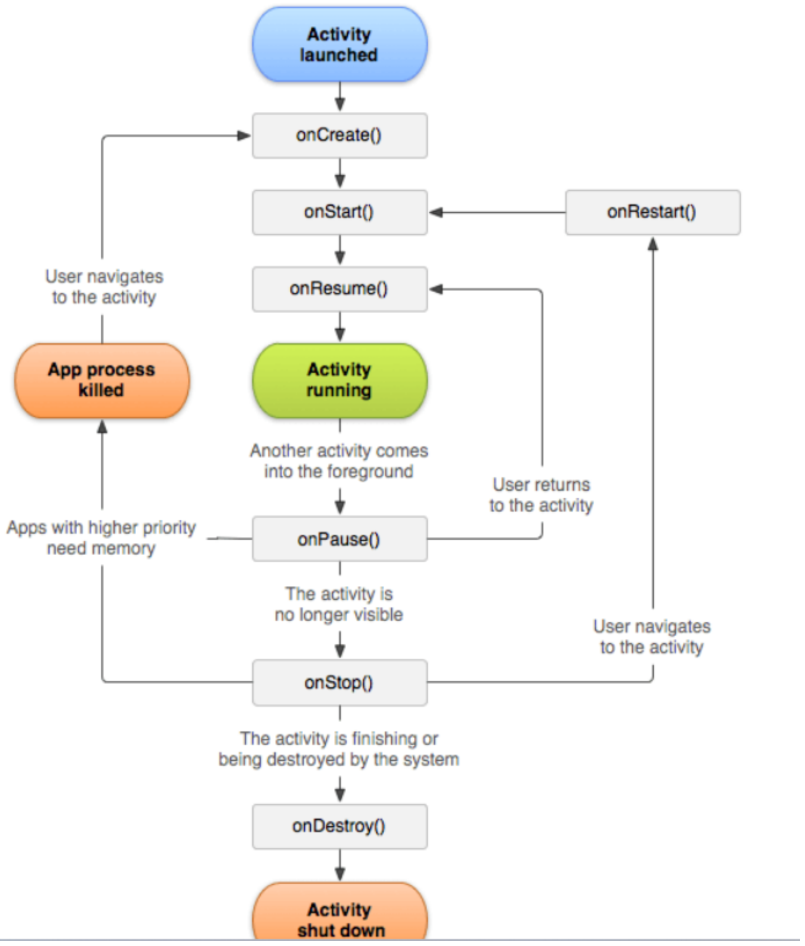
- **Intra-cell handover:** Withinacell, narrow-band interference could maketransmission at a certain frequency impossible. The BSC could then decide to change the carrier frequency (scenario
- **Inter-cell, intra-BSC handover:** This is a typical handover scenario. The mobile station moves from one cell to another, but stays within the control of the same BSC. The BSC then performs a handover, assignsanewradiochannelinthenewcellandreleasesthe old one.
- **Inter-BSC, intra-MSC handover:** As a BSC only controls a limited number of cells; GSM also has to perform handovers between cells controlled by different BSCs. This handover then has to be controlled by the MSC.
- **Inter MSC handover:** A handover could be required between two cells belonging to different MSCs. Now both MSCs perform the handover together

Q7) Develop an application to list the tourist places of Karnataka using List View.


MainActivity.java

```java
package com.example.lab5;


import android.app.Activity;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;


public class MainActivity extends Activity {

      String[] places = {
                  "Coorg",
                  "Hampi",
                  "Gokarna",
                  "Bandipur National Park",
                  "Jog Falls",
                  "Nandi Hills",
                  "Dharmashala",
                  "Mysore",
                  "Chikmagalur",
                  "Bengaluru",
                  "Badami",
                  "Nagarhole National Park"

                     };

      @Override
      protected void onCreate(Bundle savedInstanceState) {
          super.onCreate(savedInstanceState);
          setContentView(R.layout.activity_main);
          ArrayAdapter adapter= (new ArrayAdapter<String>(this,
                      android.R.layout.simple_list_item_1, places));
        ListView listview=(ListView) findViewById(R.id.listView1);
        listview.setAdapter(adapter);
      }



   }
```

Q8) Explain the Life Cycle of an Activity?

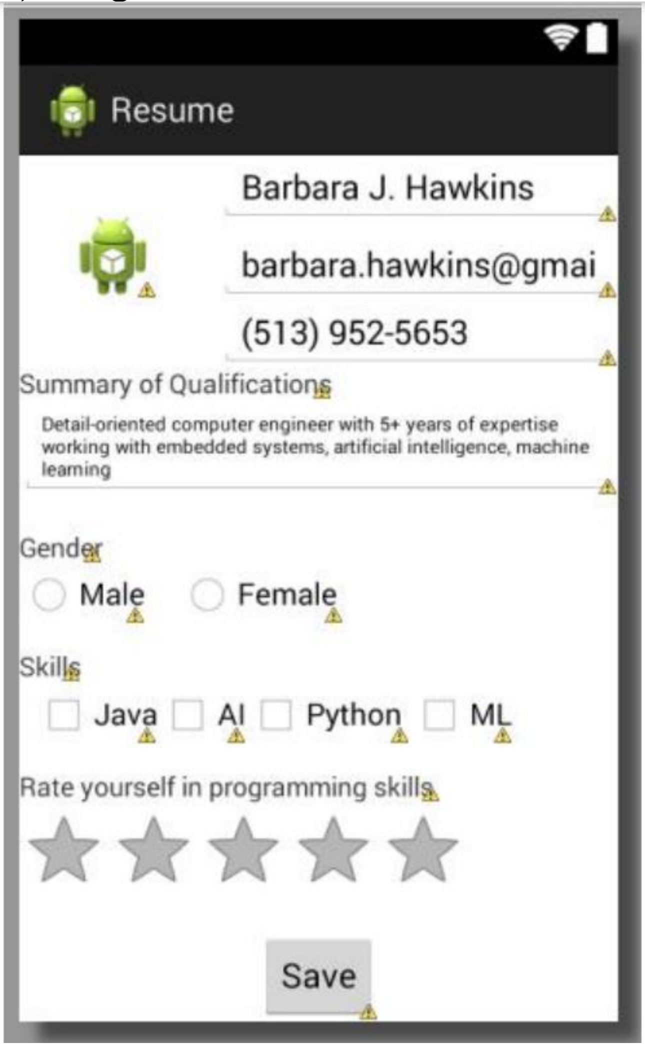

Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to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

The Activity base class defines a series of events that governs the life cycle of an activity. Below Figure shows the life cycle of an activity and the various stages it goes through — from when the activity is started until it ends.

- onCreate() Called when the Activity is created. Setup is done here. Also provides access to any previously stored state in the form of a Bundle, which can be used to restore what the user was doing before this Activity was destroyed.
- onRestart() Called if the Activity is being restarted, if it's still in the stack, rather than starting new.
- onStart() Called when the Activity is becoming visible on the screen to the user.
- onResume() Called when the Activity starts interacting with the user. (This method is always called, whether starting or restarting.)
- onPause() Called when the Activity is pausing or reclaiming CPU and other resources. This method is where you should save state information so that when an Activity is restarted, it can start from the same state it was in when it quit.
- onStop() Called to stop the Activity and transition it to a nonvisible phase and subsequent lifecycle events.

- onDestroy() Called when an Activity is being completely removed from system memory. This method is called either because onFinish() is directly invoked or because the system decides to stop the Activity to free up resources.

Q9) Design UI for Resume with different view components.


Resume
Barbara J. Hawkins
barbara.hawkins@gmai
(513) 952-5653
Summary of Qualifications
Detail-oriented computer engineer with 5+ years of expertise working with embedded systems, artificial intelligence, machine learning
Gender
Male
Female
Skills
Java
AI
Python
ML
Rate yourself in programming skills
Save

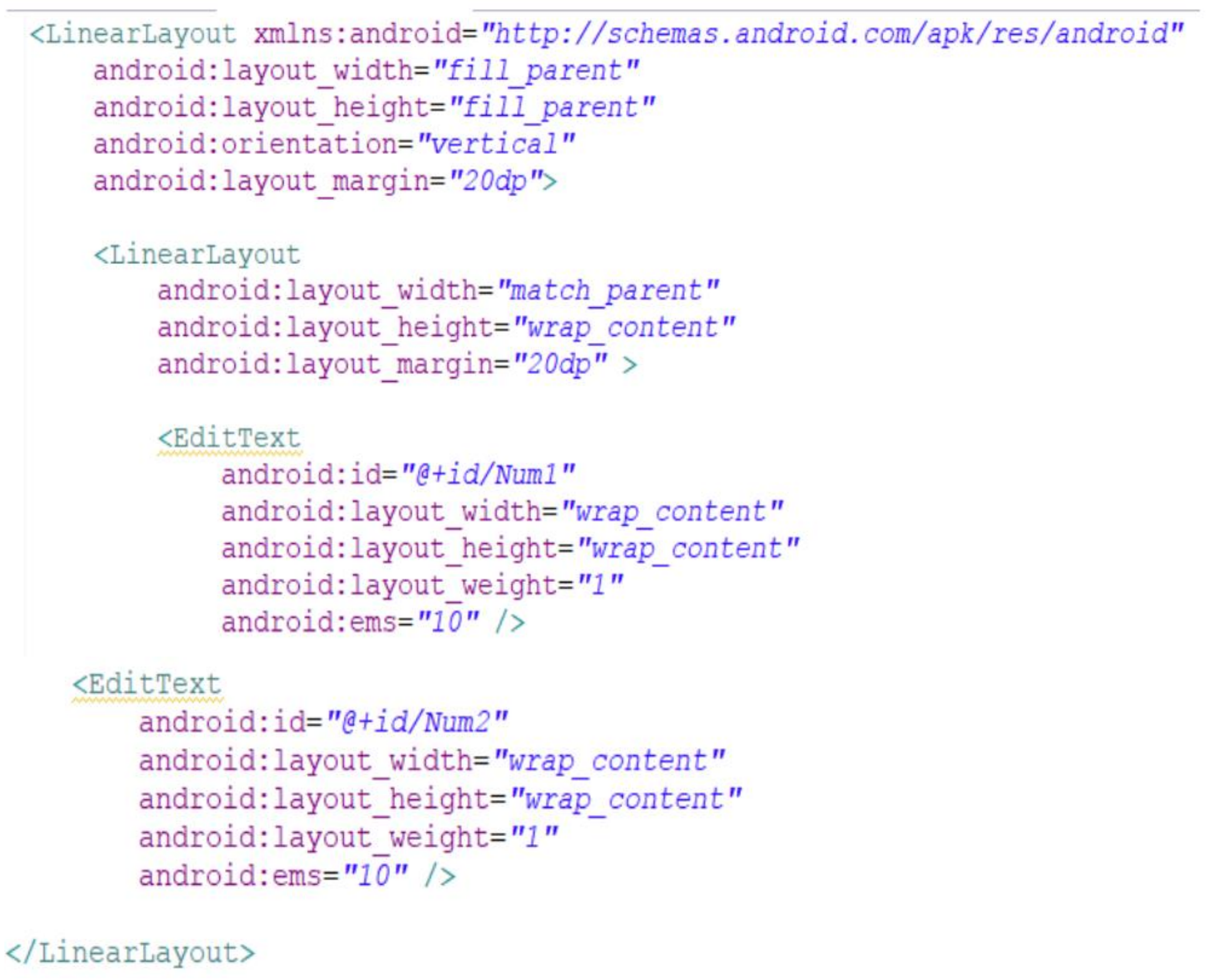
```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginLeft="26dp"
        android:layout_marginTop="29dp"
        android:src="@drawable/ic_launcher" />

    <EditText
        android:id="@+id/editText1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:ems="10"
        android:text="Barbara J. Hawkins" >

    </EditText>

    <EditText
        android:id="@+id/editText2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBottom="@+id/imageView1"
        android:layout_alignParentRight="true"
        android:ems="10"
        android:text="barbara.hawkins@gmail.com"
        android:inputType="textEmailAddress" />

    <EditText
        android:id="@+id/editText3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/editText2"
        android:layout_below="@+id/editText2"
        android:ems="10"
        android:text="(513) 952-5653"
        android:inputType="phone" />

    <EditText
        android:id="@+id/editText4"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/textView1"
        android:ems="5"
        android:inputType="textMultiLine"
        android:textSize="10"
        android:text="Detail-oriented computer engineer with 5+
years of expertise working with embedded systems, artificial
intelligence, machine learning" />
```

```
    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/editText4"
        android:layout_marginTop="19dp"
        android:text="Gender" />

    <RadioButton
        android:id="@+id/radioButton1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/textView2"
        android:text="Male" />

    <RadioButton
        android:id="@+id/radioButton2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/radioButton1"
        android:layout_alignBottom="@+id/radioButton1"
        android:layout_marginLeft="17dp"
        android:layout_toRightOf="@+id/radioButton1"
        android:text="Female " />

  <Button
      android:id="@+id/button1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentBottom="true"
      android:layout_centerHorizontal="true"
      android:text="Save" />

  <TextView
      android:id="@+id/textView3"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_above="@+id/checkBox2"
      android:layout_alignParentLeft="true"
      android:text="Skills" />

  <CheckBox
      android:id="@+id/checkBox1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
```

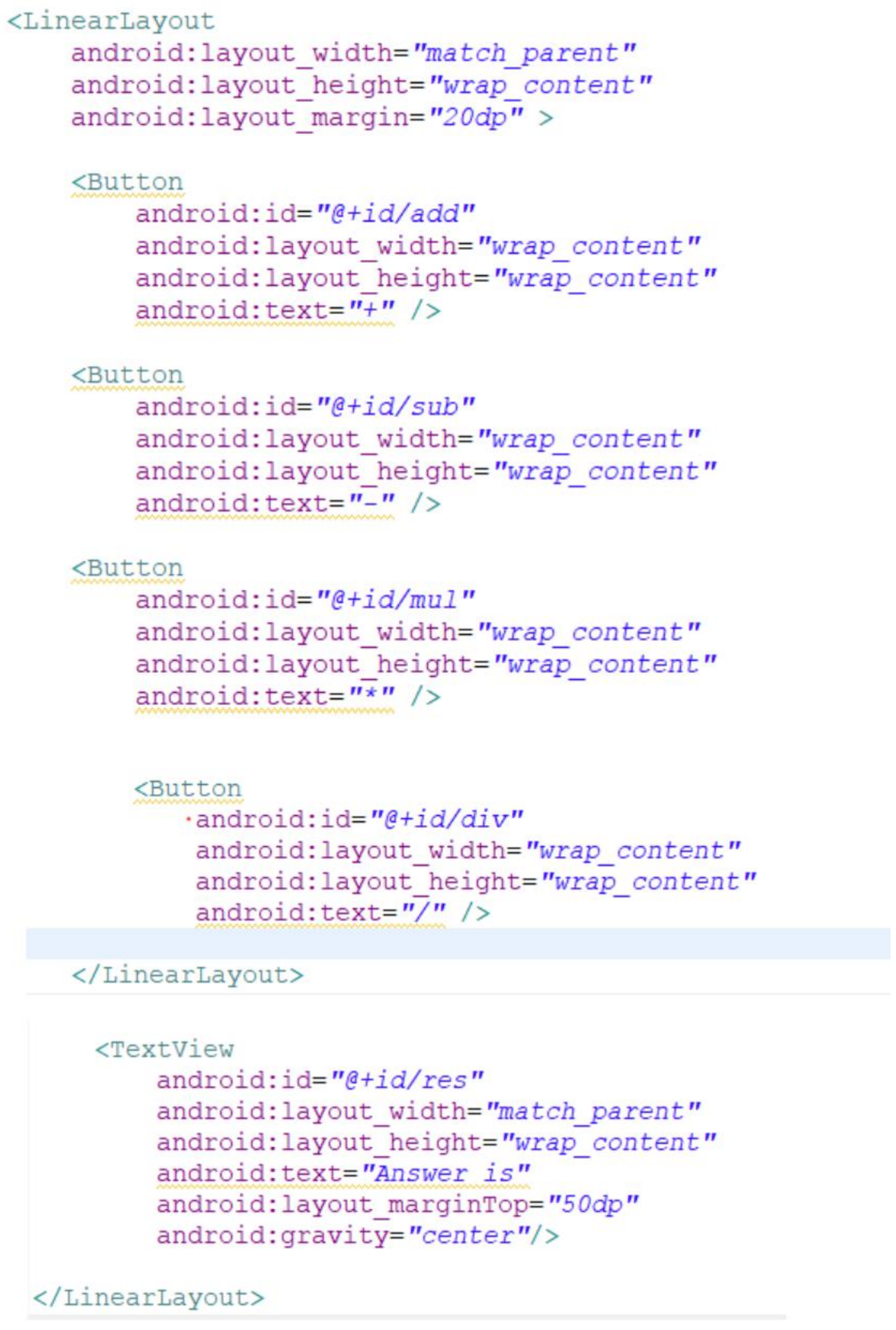
```
        android:layout_alignRight="@+id/imageView1"
        android:layout_below="@+id/radioButton1"
        android:layout_marginTop="31dp"
        android:text="Java" />

    <CheckBox
        android:id="@+id/checkBox3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/checkBox2"
        android:layout_alignBottom="@+id/checkBox2"
        android:layout_toRightOf="@+id/checkBox2"
        android:text="Python " />

    <CheckBox
        android:id="@+id/checkBox2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/checkBox1"
        android:layout_alignBottom="@+id/checkBox1"
        android:layout_toRightOf="@+id/checkBox1"
        android:text="AI" />

  <CheckBox
      android:id="@+id/checkBox4"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignBaseline="@+id/checkBox3"
      android:layout_alignBottom="@+id/checkBox3"
      android:layout_toRightOf="@+id/checkBox3"
      android:text="ML" />

  <TextView
      android:id="@+id/textView1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentLeft="true"
      android:layout_below="@+id/editText3"
      android:text="Summary of Qualifications" />

  <TextView
      android:id="@+id/textView4"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentLeft="true"
      android:layout_below="@+id/checkBox1"
      android:layout_marginTop="14dp"
```

```xml
            android:text="Rate yourself in programming skills " />

        <RatingBar
            android:id="@+id/ratingBar1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentLeft="true"
            android:layout_below="@+id/textView4" />

    </RelativeLayout>
```

Q10) Develop a standard calculator application to perform basic calculations like addition, subtraction, multiplication and division

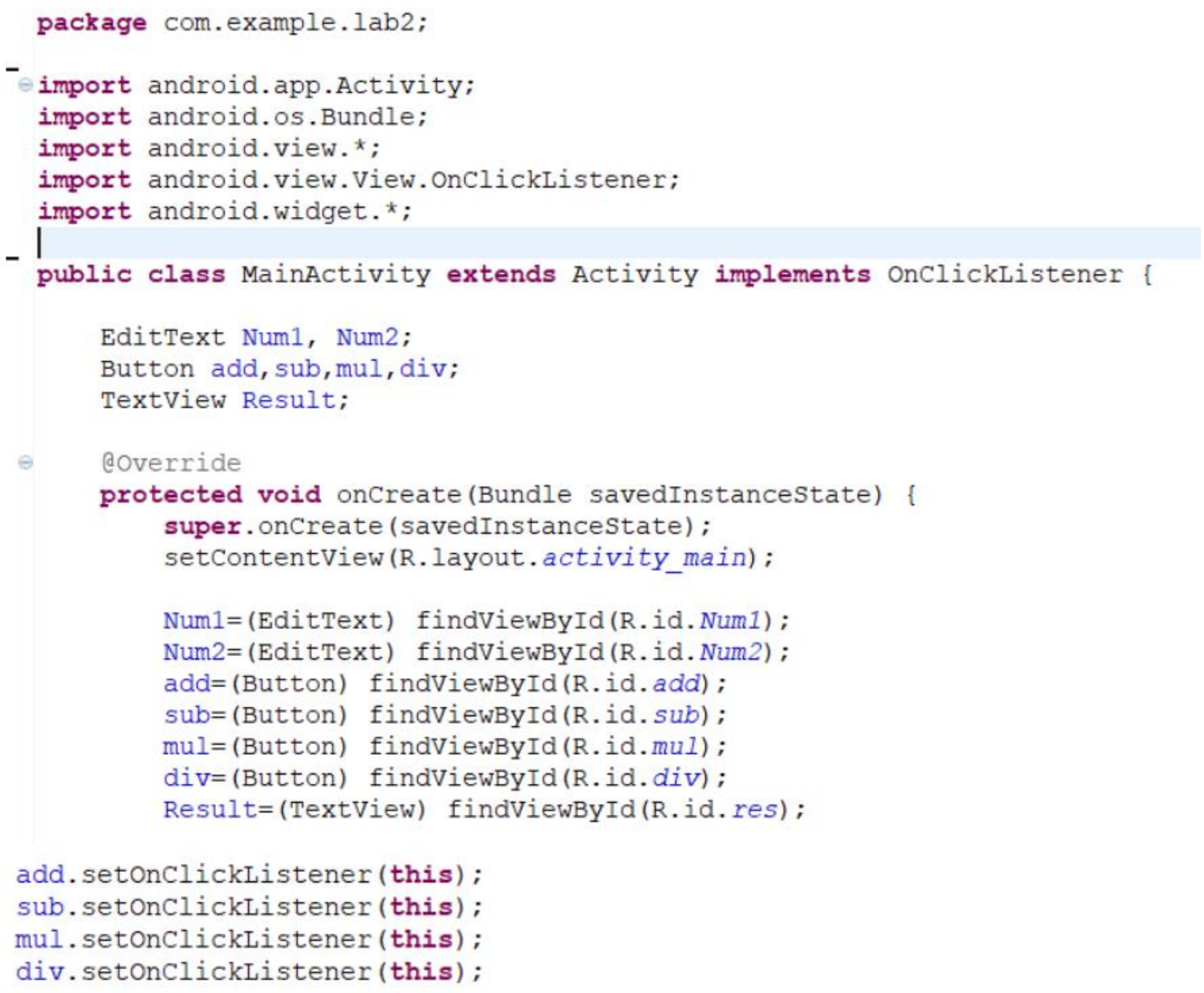


```xml
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:layout_margin="20dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="20dp" >

        <EditText
            android:id="@+id/Num1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:ems="10" />

   <EditText
       android:id="@+id/Num2"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:layout_weight="1"
       android:ems="10" />

</LinearLayout>
```

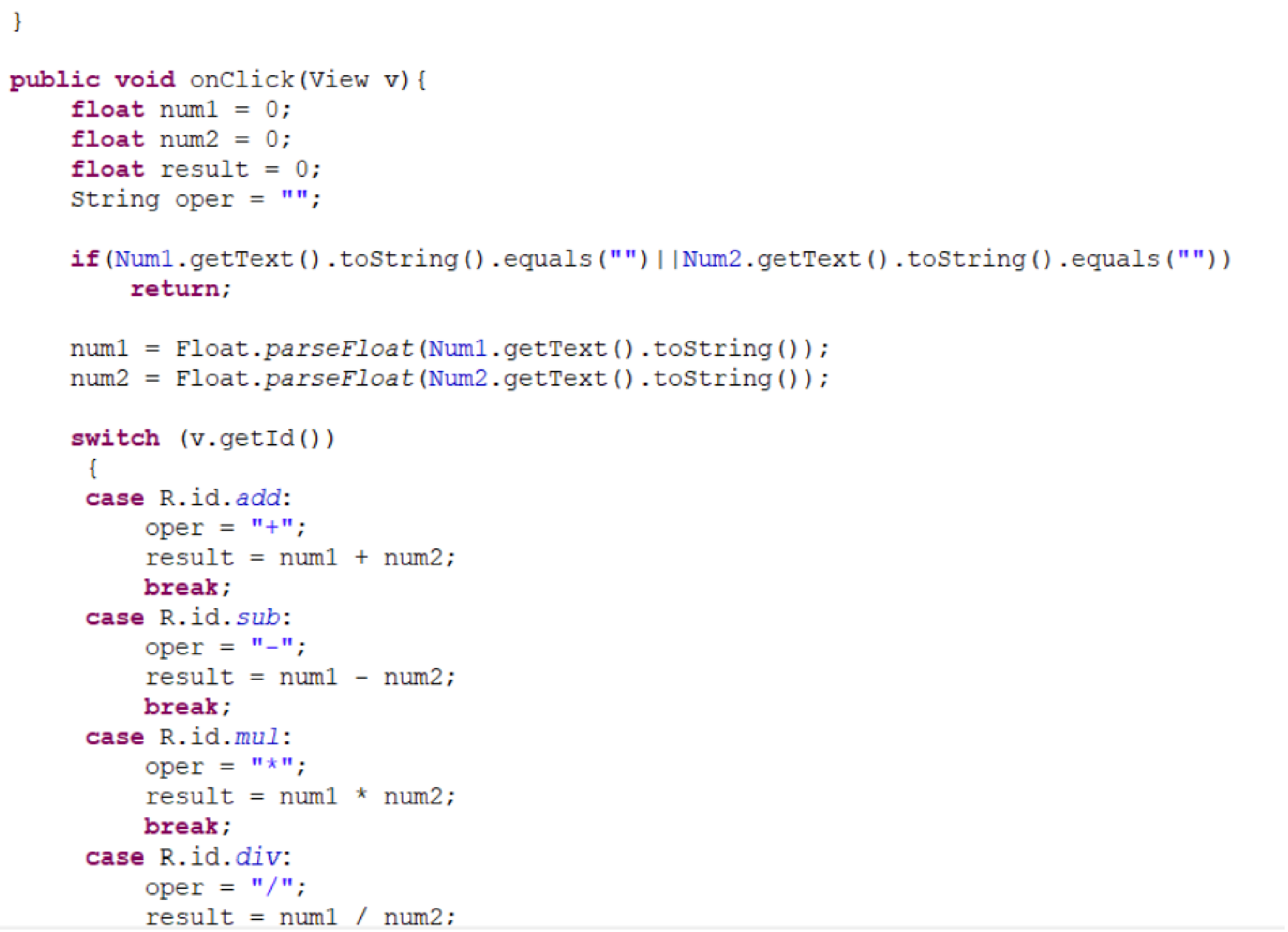


```xml
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="20dp" >

    <Button
        android:id="@+id/add"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="+" />

    <Button
        android:id="@+id/sub"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="-" />

    <Button
        android:id="@+id/mul"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="*" />


        <Button
            android:id="@+id/div"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="/" />

    </LinearLayout>

     <TextView
         android:id="@+id/res"
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:text="Answer is"
         android:layout_marginTop="50dp"
         android:gravity="center"/>

</LinearLayout>
```

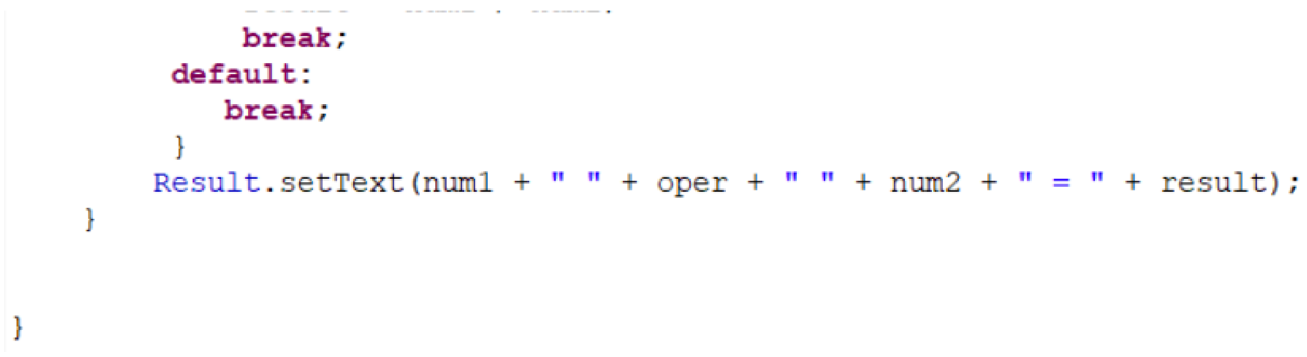
Code for MainActivity.java:



```
package com.example.lab2;

import android.app.Activity;
import android.os.Bundle;
import android.view.*;
import android.view.View.OnClickListener;
import android.widget.*;

public class MainActivity extends Activity implements OnClickListener {

    EditText Num1, Num2;
    Button add,sub,mul,div;
    TextView Result;

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

        Num1=(EditText) findViewById(R.id.Num1);
        Num2=(EditText) findViewById(R.id.Num2);
        add=(Button) findViewById(R.id.add);
        sub=(Button) findViewById(R.id.sub);
        mul=(Button) findViewById(R.id.mul);
        div=(Button) findViewById(R.id.div);
        Result=(TextView) findViewById(R.id.res);

add.setOnClickListener(this);
sub.setOnClickListener(this);
mul.setOnClickListener(this);
div.setOnClickListener(this);
```



```
}

public void onClick(View v){
     float num1 = 0;
     float num2 = 0;
     float result = 0;
     String oper = "";

     if(Num1.getText().toString().equals("")||Num2.getText().toString().equals(""))
         return;

     num1 = Float.parseFloat(Num1.getText().toString());
     num2 = Float.parseFloat(Num2.getText().toString());

     switch (v.getId())
      {
      case R.id.add:
           oper = "+";
           result = num1 + num2;
           break;
      case R.id.sub:
           oper = "-";
           result = num1 - num2;
           break;
      case R.id.mul:
           oper = "*";
           result = num1 * num2;
           break;
      case R.id.div:
           oper = "/";
           result = num1 / num2;
```



```
               break;
           default:
              break;
           }
         Result.setText(num1 + " " + oper + " " + num2 + " = " + result);
    }


}
```