Vidyavardhini's College of Engineering & Technology
Department of Computer Engineering
Academic Year : 2020-21

Class / Sem : B.E. /VII

Lab Name : Networking Lab

Course Name: Mobile Application Development Lab

Course code: CSL702

Index

Subject In charge

Prof. Sneha Mhatre

EXPERIMENT NO:1

AIM: To understand the cellular frequency reuse concept to find the co-channel cells for a particular cell.

Theory:

Frequency Reuse:

In mobile communication systems a slot of a carrier frequency / code in a carrier frequency is a radio resource unit. This radio resource unit is assigned to a user in order to support a call/ session. The number of available such radio resources at a base station thus determines the number of users who can be supported in the call. Since in wireless channels a signal is "broadcast" i.e. received by all entities therefore one a resource is allocated to a user it cannot be reassigned until the user finished the call/ session. Thus the number of users who can be supported in a wireless system is highly limited.

In order to support a large no. of users within a limited spectrum in a region the concept of frequency re-use is used.

The signal radiated from the transmitter antenna gets attenuated with increasing distance. At a certain distance the signal strength falls below noise threshold and is no longer identifiable. In this region when the signal attenuates below noise floor the same radio resource may be used by another transmission to send different information. In term of cellular systems, the same radio resource (frequency) can use by two base stations which a sufficient spaced apart. In this way the same frequency gets reused in a layer- geographic area by two or more different base station different users simultaneously.

Now what is important is to select the set of base stations which will use the same set of radio resources/ channel of frequencies or technically the co- channel cells. In this context the minimum adjacent set cells which use different frequencies each is calls a cluster.

The cellular concept is the major solution of the problem of spectral congestion and user capacity. Cellular radio relies on an intelligent allocation and channel reuse throughout a large geographical coverage region.

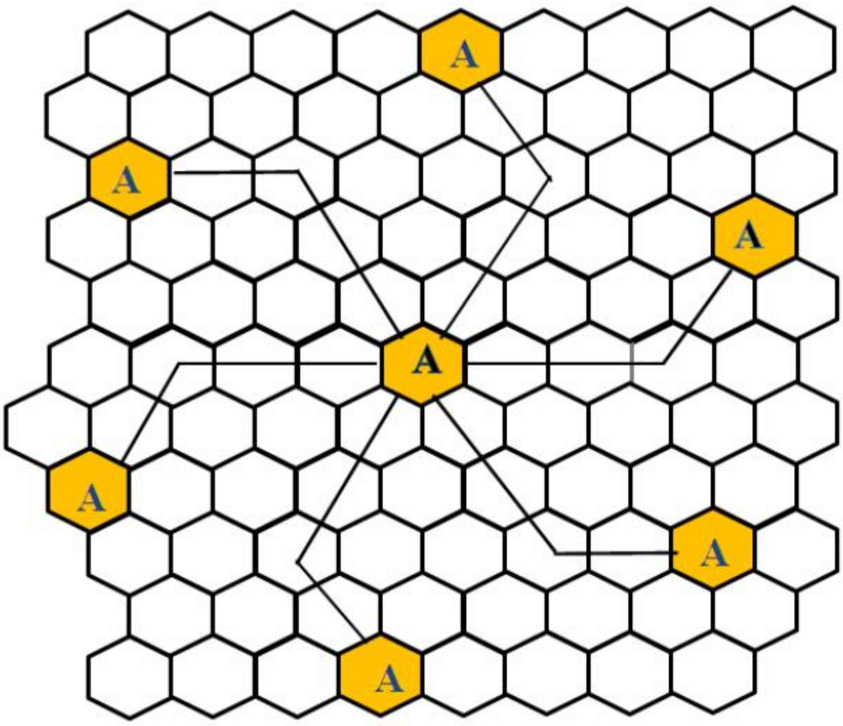
1.1 Cellular Frequency Reuse:

Each cellular base station is allocated a group of radio channels to be used within a small geographic area called a cell. Base stations in adjacent cells are assigned channel groups which contain completely different channels than neighboring cells. Base station antennas are designed to achieve the desired coverage within a particular cell. By limiting the coverage area within the boundaries of a cell, the same group of channels may be used to cover different cells that are separated from one another by geographic distances large enough to keep interference levels within tolerable limits. The design process of selecting and allocating channel groups for all cellular base stations within a system is called frequency reuse or frequency planning.

1.4 Co-channel Cells:

A larger cluster size causes the ratio between the cell radius and the distance between co-channel cells to decrease reducing co-channel interference. The value of N is a function of how much interference a mobile or base station can tolerate while maintaining a sufficient quality of communications. Since each hexagonal cell has six equidistant neighbors and the line joining the centers of any cell and each of its neighbors are separated by multiples of 60 degrees, only certain cluster sizes and cell layouts are possible. To connect without gaps between adjacent cells, the geometry of hexagons is such that the numbers of cells per cluster, N, can only have values that satisfy,

$N=i^2+ij+j^2$

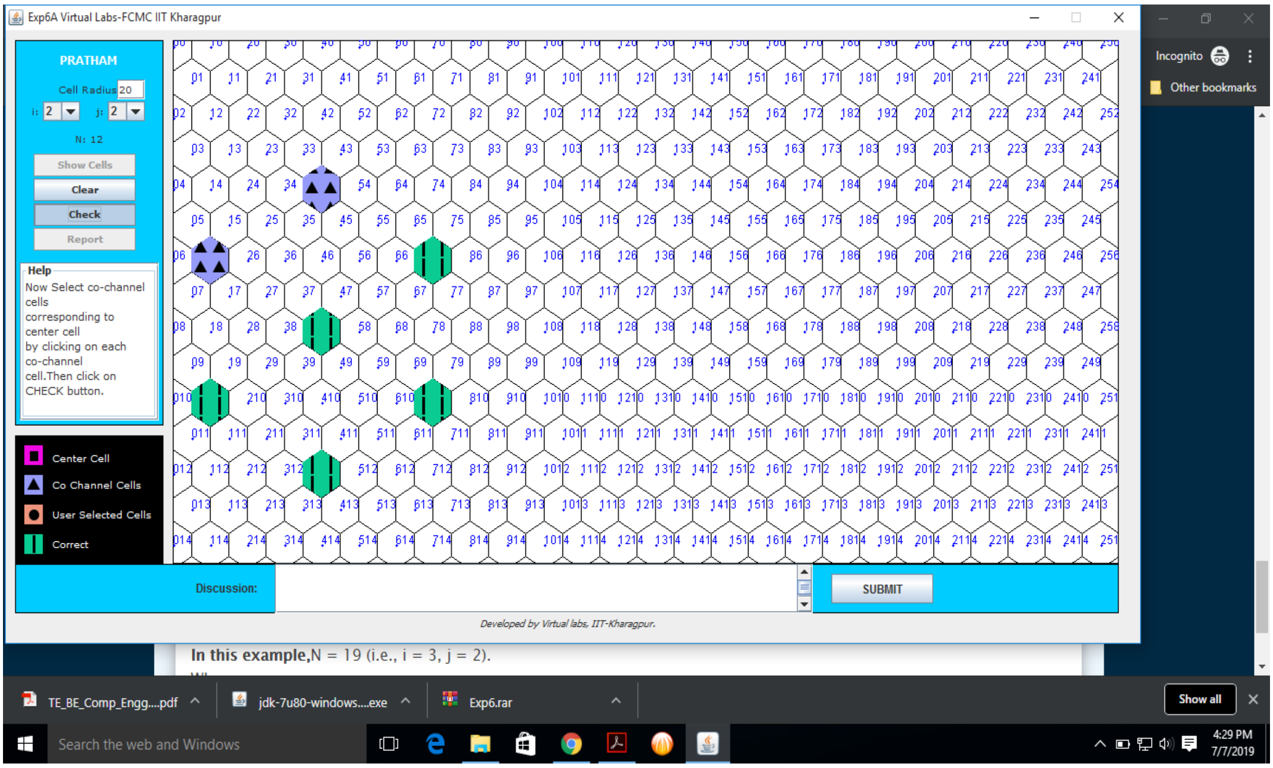


Method of locating co-channel cells in a cellular system. In this figure, N=19(i.e, i =3, j=2).

In this example,N = 19 (i.e., i = 3, j = 2).
Where,
i and j are non-negative integers.
To find the nearest co-channel neighbours of a particular cell,
a. move i cells along any chain of hexagons then,
b. turn 60 degrees counter-clockwise and move j cells.

Link for DEMO Video:
http://vlabs.iitkgp.ernet.in/fcmc/exp6/index.html#

OUTPUT:



Exp6A Virtual Labs-FCMC IIT Kharagpur

PRATHAM

Cell Radius 20

i: 2 j: 2

N: 12

Help

Now Select co-channel cells corresponding to center cell by clicking on each co-channel cell.Then click on CHECK button.

Discussion:

Developed by Virtual labs, IIT-Kharagpur.

In this example,N = 19 (i.e., i = 3, j = 2).

CONCLUSION:

Thus Game based Application to find the co-channel cells for a particular cell executed successfully.

EXPERIMENT NO:2

AIM: To understand the cellular frequency reuse concept to find the cell clusters within certain geographic area.

Theory:

Hexagonal Cell Structure:

In below figure, cells labeled with the same letter use the same group of channels. The hexagonal cell shape is conceptual and is the simplistic model of the radio coverage for each base station. It has been universally adopted since the hexagon permits easy and manageable analysis of a cellular system. The actual radio coverage of a system is known as the footprint and is determined from old measurements and propagation prediction models. Although the real footprint is amorphous in nature, a regular cell shape is needed for systematic system design and adaptation for future growth.

If a circle is chosen to represent the coverage area of a base station, adjacent circles overlaid upon a map leave gaps or overlapping regions. A square, an equilateral triangle and a hexagon can cover the entire area without overlap and with equal area. A cell must serve the weakest mobiles typically located at the edge of the cell within the foot print. For a given distance between the center of a polygon and its farthest perimeter points, the hexagon has the largest area of the three. Thus, with hexagon, the fewest number of cell scan cover a geographic region and close approximation of a circular radiation pattern that occurs for an Omni directional base antenna and free space propagation is possible.

Base station transmitters are situated either at the center of the cell (center-excited cells) or at three of the six cell vertices (edge-excited cells). Normally, omnidirectional antennas are used in center-exited cells and sectored directional antennas are used in edge-exited cells. Practical system design considerations permit a base station to be positioned up to one-fourth the cell radius away from the ideal location.

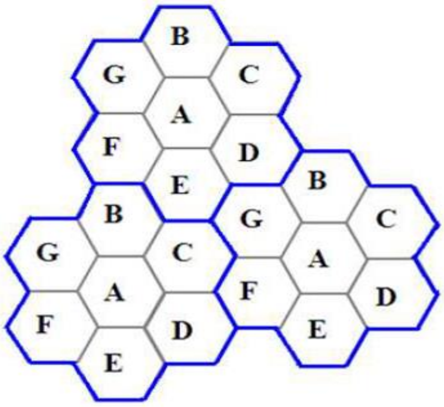
Cell Cluster:

Considering a cellular system that has a total of S duplex radio channels. If each cell is allocated a group of k channels and (k<S) if the S channels are divided among N cells into unique and disjoint channel groups of same number of channels, then,

$S=kN$

The N cells that collectively use the complete set of available frequencies are called a cluster. If a cluster is replicated M times within the system, the total number of duplex channels or capacity,

$C=MkN=MS$

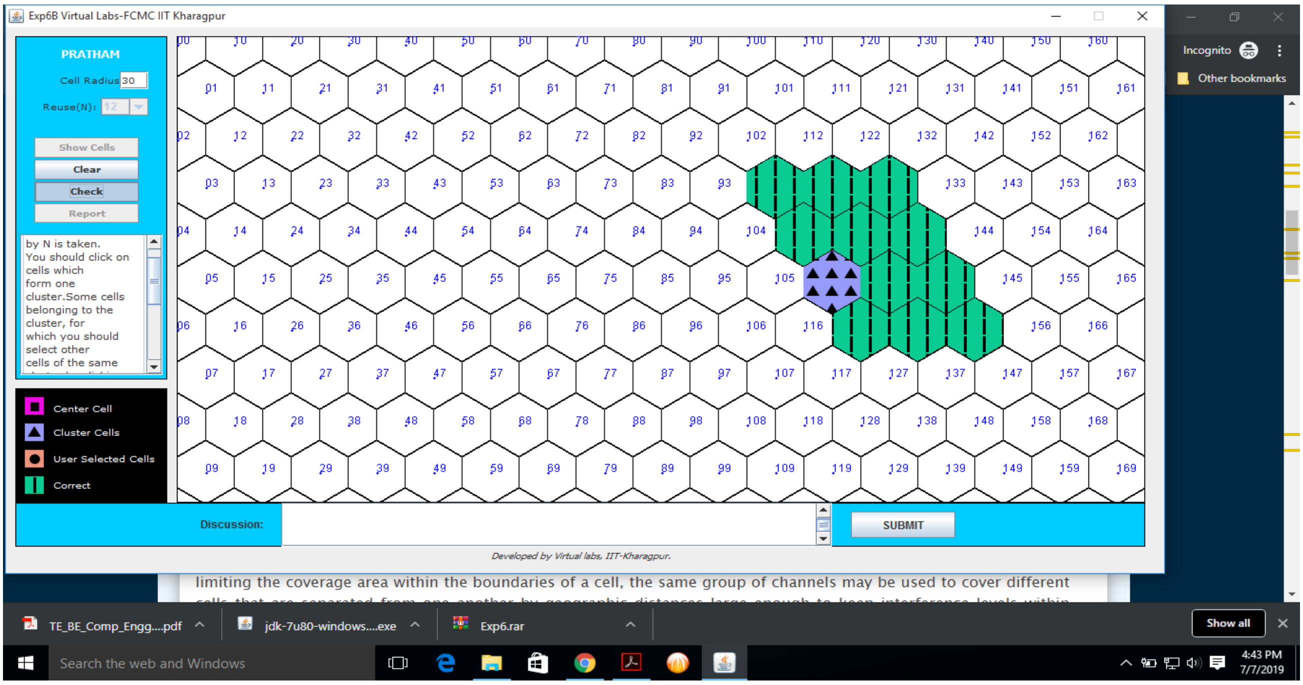


Frequency reuse concept, Cells with the same letter use the same set of frequencies. A cell cluster is outline in blue color and replicated over the coverage area.

In this example,
The cluster size N = 7 and the frequency reuse factor is 1/7 since each cell contains one-seventh of the total number of available channels.
The capacity is directly proportional to M. The factor N is called the cluster size and is typically 4, 7 or 12. If the cluster size N is reduced while the cell size is kept constant, more clusters are required to cover a given area and hence more capacity is achieved from the design viewpoint, the smallest possible value of N is desirable to maximize capacity over a given coverage area. The frequency reuse factor of a cellular system is 1/N, since each cell within a cluster is assigned 1/N of the total available channels in the system.

Link for DEMO Video:
http://vlabs.iitkgp.ernet.in/fcmc/exp6B/index.html#

OUTPUT:



Cell Radius 30

Reuse(N): 12

by N is taken.
You should click on cells which form one cluster.Some cells belonging to the cluster, for which you should select other cells of the same

Discussion:

Developed by Virtual labs, IIT-Kharagpur.

limiting the coverage area within the boundaries of a cell, the same group of channels may be used to cover different

CONCLUSION:
Thus Game based Application to find the cell clusters within certain geographic area executed successfully.

EXPERIMENT NO:3

AIM: To implement Mobile node discovery

Program:

```
package mypackage;
import java.io.IOException;
import javax.bluetooth.*;
import javax.bluetooth.DiscoveryListener;
import javax.microedition.lcdui.*;
import javax.microedition.midlet.*;
public class discover_device extends MIDlet implements
CommandListener,DiscoveryListener {
private final List deviceList;
private final Command Exit,Refresh;
private String deviceName;
private DiscoveryAgent agent;
private Alert dialog;
public discover_device()
{
 deviceList = new List("List of Devices",List.IMPLICIT);
 Exit= new Command("Exit",Command.EXIT, 0);
 Refresh = new Command("Refresh",Command.SCREEN, 1);
  deviceList.addCommand(Exit);
 deviceList.addCommand(Refresh);
 deviceList.setCommandListener(this);
 Display.getDisplay(this).setCurrent(deviceList);
}
public void startApp() {
try {
deviceList.deleteAll();
LocalDevice local = LocalDevice.getLocalDevice();
local.setDiscoverable(DiscoveryAgent.GIAC);
deviceName = local.getFriendlyName();
agent = local.getDiscoveryAgent();
}
catch (BluetoothStateException ex) {
ex.printStackTrace();
}
try {
 agent.startInquiry(DiscoveryAgent.GIAC, this);
}
catch (BluetoothStateException ex) {
ex.printStackTrace();
}
}
public void pauseApp() {
}
public void destroyApp(boolean unconditional) {
}
public void commandAction(Command c, Displayable d) {
if(c==Exit)
{
this.destroyApp(true);
notifyDestroyed();
}
if(c==Refresh){
this.startApp();
}
}
public void deviceDiscovered(RemoteDevice btDevice, DeviceClass cod) {
String deviceaddress = null;
try {
deviceaddress = btDevice.getBluetoothAddress();//btDevice.getFriendlyName(true);
} catch (Exception ex) {
ex.printStackTrace();
}
deviceList.insert(0, deviceaddress , null);
}
public void servicesDiscovered(int transID, ServiceRecord[] servRecord) {
 throw new UnsupportedOperationException("Not supported yet.");
}
 public void serviceSearchCompleted(int transID, int respCode) {
        throw new UnsupportedOperationException("Not supported yet.");
    }
public void inquiryCompleted(int discType) {
        Alert dialog = null;
        if (discType != DiscoveryListener.INQUIRY_COMPLETED) {
            dialog = new Alert("Bluetooth Error","The inquiry failed to complete normally",null,
AlertType.ERROR);
          }
         else {
              dialog = new Alert("Inquiry Completed","The inquiry completed normally",
null,AlertType.INFO);
              }
              dialog.setTimeout(500);
              Display.getDisplay(this).setCurrent(dialog);
    }
}
```

Blue.java

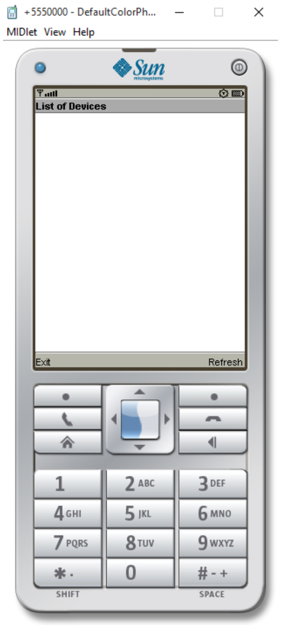
```
package mypackage;
import java.io.*;
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import javax.microedition.io.*;
import javax.bluetooth.*;
import java.util.*;
public class Blue extends MIDlet implements CommandListener,DiscoveryListener
{
        private List activeDevices;
        private Command select,exit;
        private Display display;
        private LocalDevice local=null;
        private DiscoveryAgent agent = null;
        private Vector devicesFound = null;
        private ServiceRecord[] servicesFound = null;
        private String connectionURL = null;
        public void startApp() {
                display = Display.getDisplay(this);
                activeDevices = new List("Active Devices", List.IMPLICIT);
                select = new Command("Search Again", Command.OK, 0);
                exit = new Command("Exit", Command.EXIT, 0);
                activeDevices.addCommand(exit);
                activeDevices.setCommandListener(this);
                try {
                        local = LocalDevice.getLocalDevice(); }
                catch (Exception e) {}
                doDeviceDiscovery();
                display.setCurrent(activeDevices);
        }
        public void pauseApp() {}
        public void destroyApp(boolean unconditional) { notifyDestroyed(); }
        public void commandAction(Command cmd, Displayable disp) {
                if (cmd == select && disp == activeDevices) {
                        activeDevices.deleteAll();
                         doDeviceDiscovery();
                }
                if (cmd == exit) { destroyApp(false); }
        }
        public void inquiryCompleted(int param) {
                try {
                switch (param) {
                         case DiscoveryListener.INQUIRY_COMPLETED:

                                        if (devicesFound.size() > 0) {
                                                activeDevices.addCommand(select);

       activeDevices.setSelectCommand(select);
                                        }
                                        else { activeDevices.append("No Devices
Found", null); }
                                        break; }
                }
                catch (Exception e) {}
        }
        public void serviceSearchCompleted(int transID, int respCode) {}
        public void servicesDiscovered(int transID, ServiceRecord[] serviceRecord) {}
        public void deviceDiscovered(RemoteDevice remoteDevice, DeviceClass
deviceClass) {
```

```
                        String str = null;
                        try {
                                 str = remoteDevice.getBluetoothAddress() + " - ";
                                 str += remoteDevice.getFriendlyName(true);
                        } catch (Exception e) {}
                        activeDevices.append(str, null);
                        devicesFound.addElement(remoteDevice);
try {
        if (!agent.startInquiry(DiscoveryAgent.GIAC, this)) {}
} catch (BluetoothStateException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
}
         }
         private void doDeviceDiscovery() {
                        try {
                                        local = LocalDevice.getLocalDevice();
                                        agent = local.getDiscoveryAgent();
                                        devicesFound = new Vector();
                        } catch (Exception e) {}
         }
 }
```

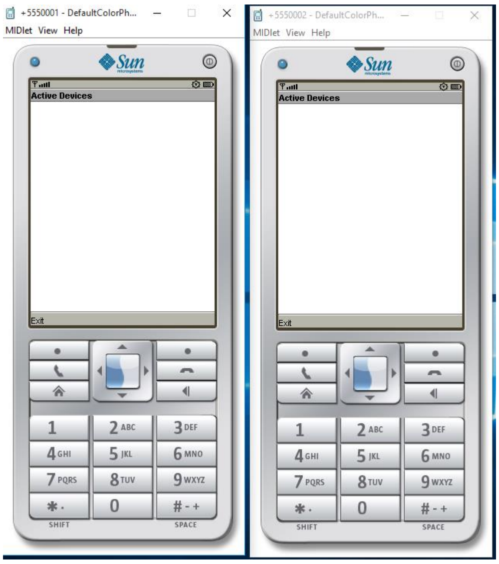
OUTPUT:

Mobile Node Discovery

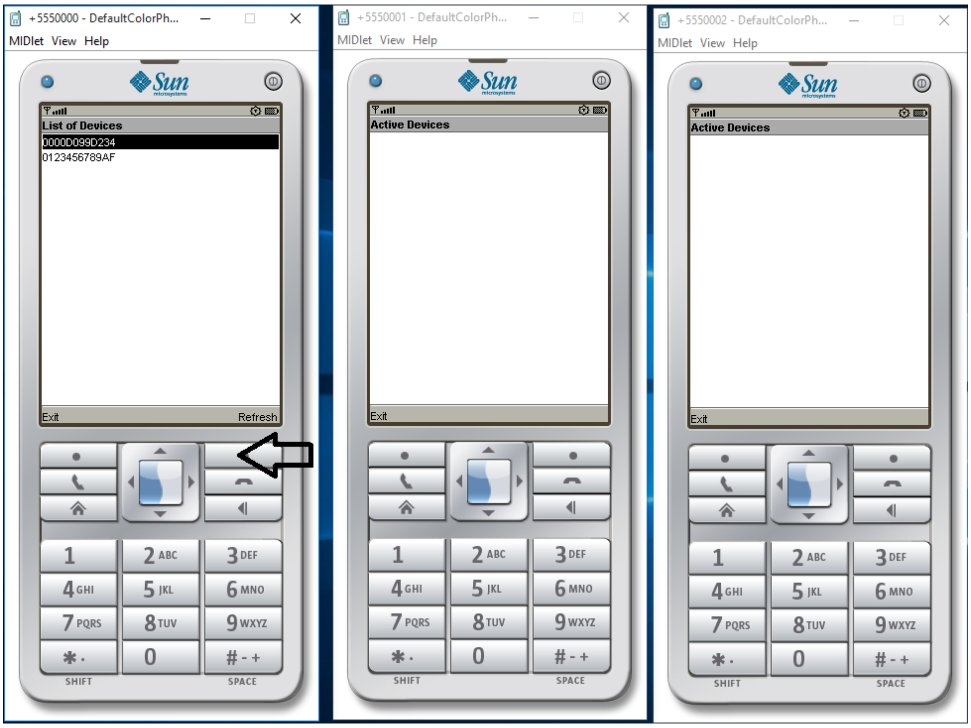
Step 1: First run discover_device.java. You will see this as the output. **(Hereafter referred as Device 0)**



Step 2: Run Blue.java file (You can run this file any number of times). For example, I have run this file for two times. So here I can see that two devices are running. **(Hereafter referred as Device 1 and Device 2 respectively).**



Step 3: Click in **Device 0** to the button pointed by the arrow. After clicking you will be able to see the Device number of **Device 1 & Device 2** as shown below.



List of Devices
0000D099D234
0123456789AF

CONCLUSION:

Thus Mobile node discovery program implemented successfully.

EXPERIMENT NO:4

AIM: To develop a Android Application that writes data to the SD Card.

Procedure:

- Open eclipse or android studio and create new project
- Select our project in the project explorer
- Go to res folder and select layout Double click the main xml file
- Type the code for main.xml or drag and drop various
- components used in our program
- Drag and drop relative layout and change its properties
- Drag and drop image view and change its properties according
- to our programs
- Screen layout can be viewed by clicking graphics layout tab
- Include necessary files
- Override OnCreate() function
- Create Image view and initialize its using id of some
- components used in the xml program
- Save the program
- Run the program
- Output can be viewed in the android emulator

Code for Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_margin="20dp"
android:orientation="vertical"
<EditText
android:id="@+id/editText"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:singleLine="true"
android:textSize="30dp" />
<Button
android:id="@+id/button"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_margin="10dp"
android:text="Write Data"
android:textSize="30dp" />
<Button
android:id="@+id/button2"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_margin="10dp"
android:text="Read data"
```

```
android:textSize="30dp" />
<Button
android:id="@+id/button3"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_margin="10dp"
android:text="Clear"
android:textSize="30dp" />
</LinearLayout>
```

Code for AndroidManifest.xml:

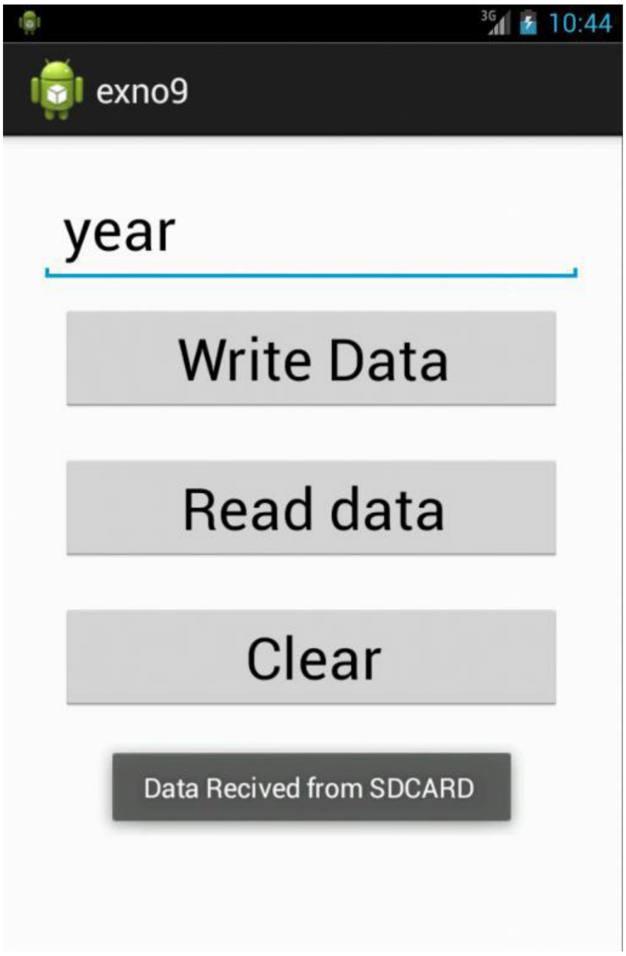
```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.exno9" >
<uses-permission android:name="android.permission.INTERNET
"></uses-permission>
<application
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:supportsRtl="true"
android:theme="@style/AppTheme" >
<activity android:name=".MainActivity" >
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category
android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
</application>
</manifest>
```

Code for MainActivity.java:

```
package com.example.exno9;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStreamReader;
public class MainActivity extends AppCompatActivity
{
EditText e1;
Button write,read,clear;
```

```
@Override
protected void onCreate(Bundle savedInstanceState){
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
e1= (EditText) findViewById(R.id.editText);
write= (Button) findViewById(R.id.button);
read= (Button) findViewById(R.id.button2);
clear= (Button) findViewById(R.id.button3);
write.setOnClickListener(new View.OnClickListener()
{@Override
public void onClick(View v)
{
String message=e1.getText().toString();
try{
File f=new File("/sdcard/myfile.txt");
f.createNewFile();
FileOutputStream fout=new FileOutputStream(f);
fout.write(message.getBytes());
fout.close();
Toast.makeText(getBaseContext(),"Data Written in
SDCARD",Toast.LENGTH_LONG).show();
}
catch (Exception e)
{
Toast.makeText(getBaseContext(),e.getMessage(),Toast.
LENGTH_LONG).show();
}
}
});
read.setOnClickListener(new View.OnClickListener()
{
@Override
public void onClick(View v)
{
String message;
String buf = "";
try
{
File f = new File("/sdcard/myfile.txt");
FileInputStream fin = new FileInputStream(f);
BufferedReader br = new BufferedReader(new
InputStreamReader(fin));
while ((message = br.readLine()) != null)
{
buf += message;
}
e1.setText(buf);
br.close();
fin.close();
Toast.makeText(getBaseContext(),"Data Recived from
SDCARD",Toast.LENGTH_LONG).show();
}
catch (Exception e)
{
Toast.makeText(getBaseContext(), e.getMessage(),
Toast.LENGTH_LONG).show();
}
} });
clear.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v{
e1.setText(""); }
});
}}
```

Output:



CONCLUSION:

Thus Android Application that writes data to the SD Card is developed and executed successfully.

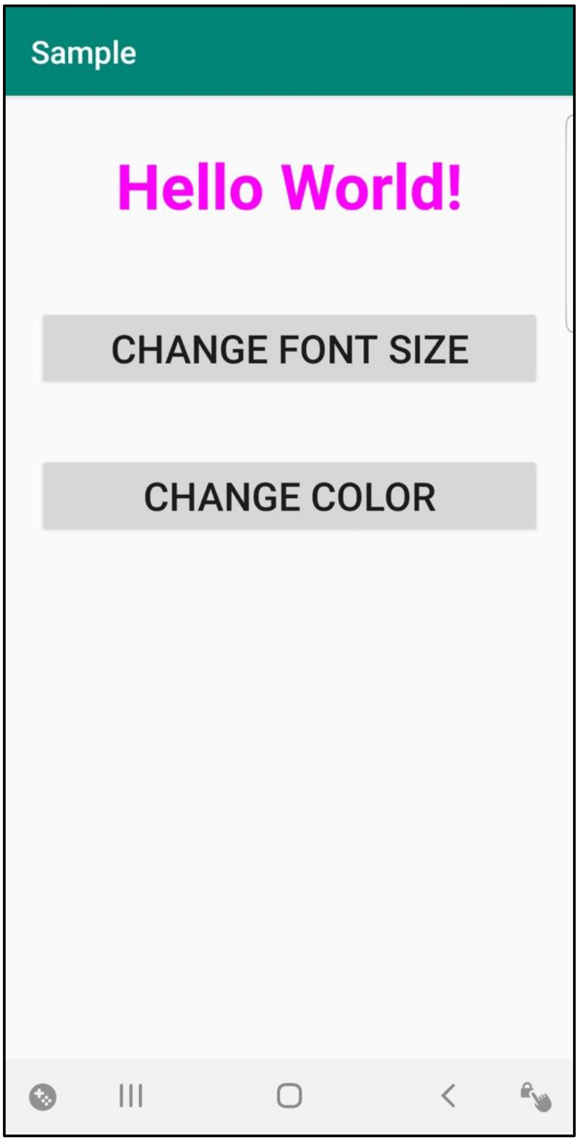
EXPERIMENT NO:5

AIM: Develop an application that uses GUI components.

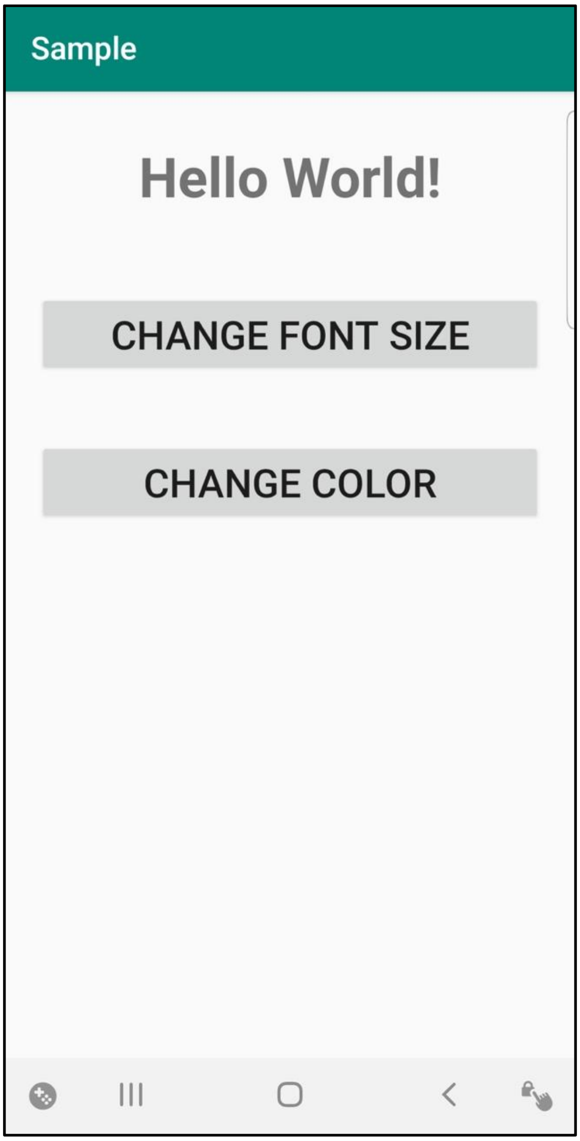
Program:

```
package com.example.sample;
import androidx.appcompat.app.AppCompatActivity;
import android.graphics.Color;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
public class MainActivity extends AppCompatActivity
{
    int ch=1;
    float font=30;
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        final TextView t= (TextView) findViewById(R.id.textView);
        Button b1= (Button) findViewById(R.id.button1);
        b1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                t.setTextSize(font);
                font = font + 5;
                if (font == 50)
                    font = 30;
            }
        });
        Button b2= (Button) findViewById(R.id.button2);
        b2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                switch (ch) {
                    case 1:
                        t.setTextColor(Color.RED);
                        break;
                    case 2:
                        t.setTextColor(Color.GREEN);
                        break;
                    case 3:
                        t.setTextColor(Color.BLUE);
                        break;
                    case 4:
                        t.setTextColor(Color.CYAN);
                        break;
                    case 5:
                        t.setTextColor(Color.YELLOW);
                        break;
                    case 6:
                        t.setTextColor(Color.MAGENTA);
                        break;
                }
                ch++;
                if (ch == 7)
                    ch = 1;
            }
        });
    }
}
```

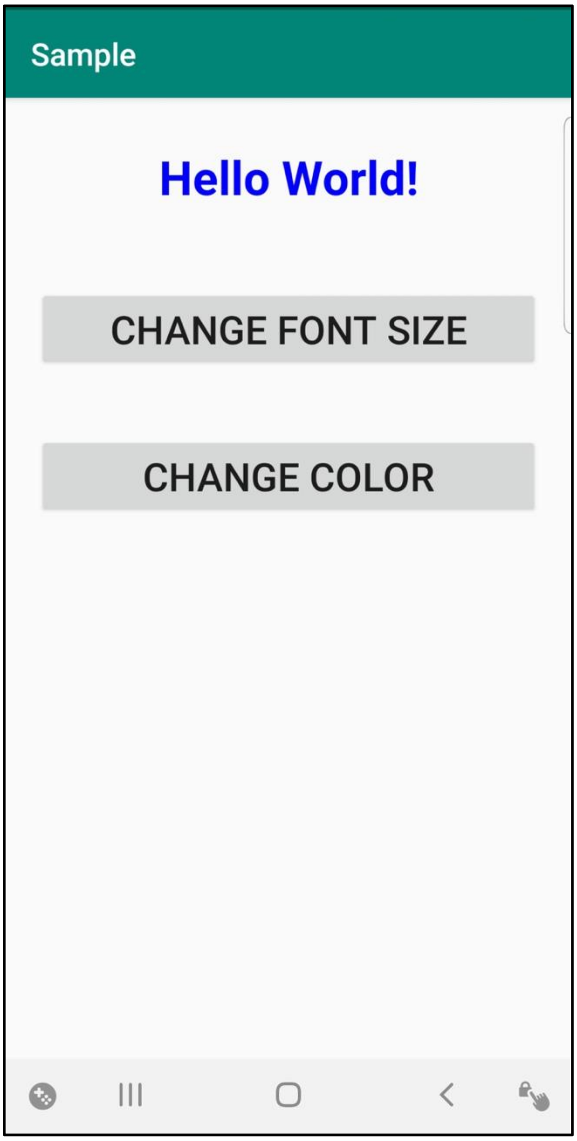
OUTPUT:

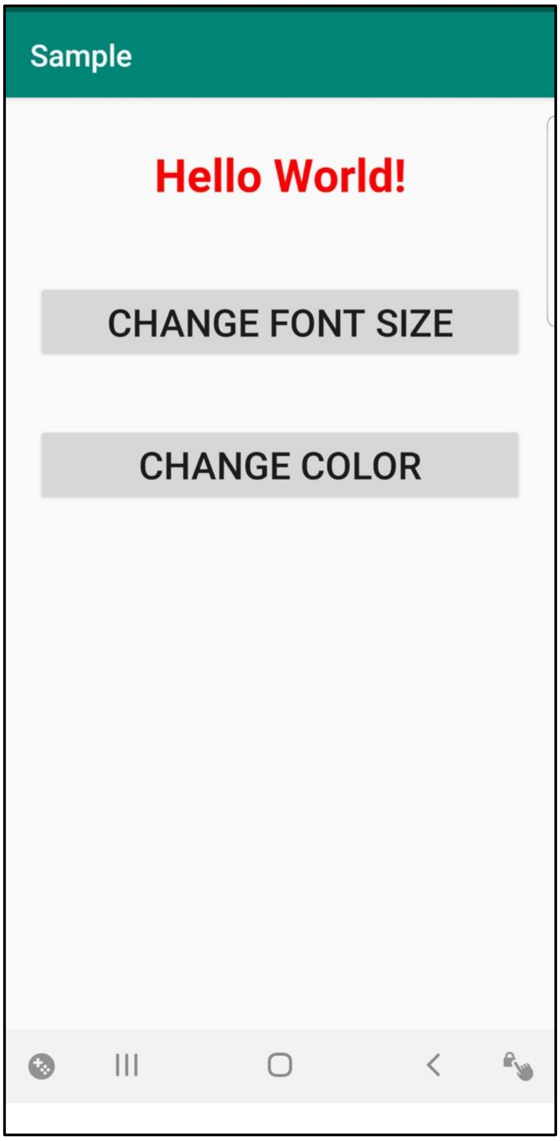

Sample

Hello World!


Sample

Hello World!





CONCLUSION:

Thus, Android Application that uses GUI components is developed and executed successfully.

EXPERIMENT NO:6

AIM: Write an application that draws basic graphical primitives on the screen.

Program:

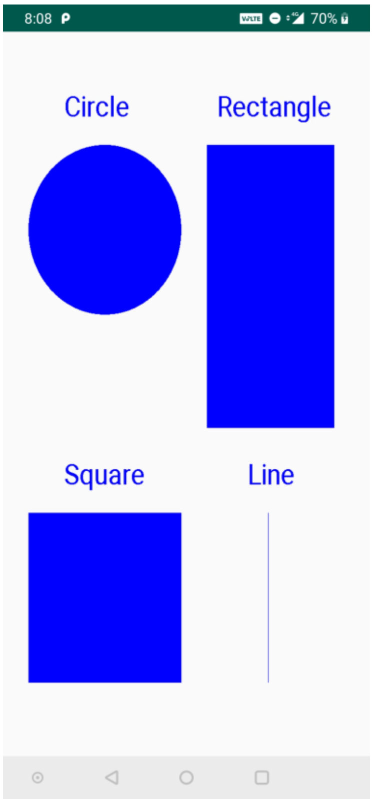
activity_main.xml

```
<?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.exp11;

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);

//To draw a Square
canvas.drawText("Square", 120, 800, paint);
canvas.drawRect(50, 850, 350, 1150, paint);

//To draw a Line
canvas.drawText("Line", 480, 800, paint);
canvas.drawLine(520, 850, 520, 1150, paint);
}
}
```

OUTPUT



CONCLUSION:
Thus, Android Application that draws basic graphical primitives on the screen is developed and executed successfully.

EXPERIMENT NO:7

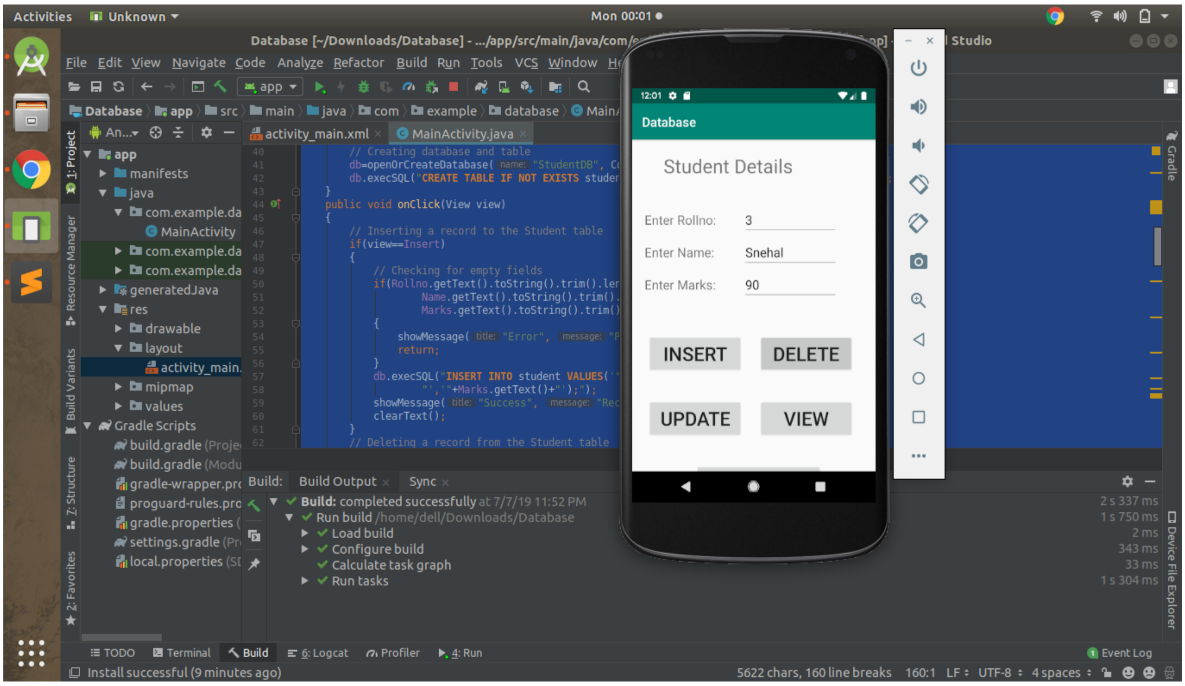
AIM: Develop an application that makes use of database.

Program:

```
package com.example.database;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.app.AlertDialog.Builder;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
public class MainActivity extends AppCompatActivity implements OnClickListener {
    EditText Rollno,Name,Marks;
    Button Insert,Delete,Update,View,ViewAll;
    SQLiteDatabase db;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Rollno=(EditText)findViewById(R.id.Rollno);
        Name=(EditText)findViewById(R.id.Name);
        Marks=(EditText)findViewById(R.id.Marks);
        Insert=(Button)findViewById(R.id.Insert);
        Delete=(Button)findViewById(R.id.Delete);
        Update=(Button)findViewById(R.id.Update);
        View=(Button)findViewById(R.id.View);
        ViewAll=(Button)findViewById(R.id.ViewAll);

        Insert.setOnClickListener(this);
        Delete.setOnClickListener(this);
        Update.setOnClickListener(this);
        View.setOnClickListener(this);
        ViewAll.setOnClickListener(this);

        // Creating database and table
        db=openOrCreateDatabase("StudentDB", Context.MODE_PRIVATE, null);
        db.execSQL("CREATE TABLE IF NOT EXISTS student(rollno VARCHAR,name
VARCHAR,marks VARCHAR);");
    }
    public void onClick(View view)
    {
        // Inserting a record to the Student table
        if(view==Insert)
        {
           // Checking for empty fields
           if(Rollno.getText().toString().trim().length()==0||
                  Name.getText().toString().trim().length()==0||
                  Marks.getText().toString().trim().length()==0)
           {
              showMessage("Error", "Please enter all values");
              return;
           }
           db.execSQL("INSERT INTO student
VALUES('"+Rollno.getText()+"','"+Name.getText()+
                  "','"+Marks.getText()+"');");
           showMessage("Success", "Record added");
           clearText();
        }
        // Deleting a record from the Student table
        if(view==Delete)
        {
           // Checking for empty roll number
           if(Rollno.getText().toString().trim().length()==0)
           {
              showMessage("Error", "Please enter Rollno");
              return;
           }
           Cursor c=db.rawQuery("SELECT * FROM student WHERE
rollno='"+Rollno.getText()+"'", null);
           if(c.moveToFirst())
           {
              db.execSQL("DELETE FROM student WHERE rollno='"+Rollno.getText()+"'");
              showMessage("Success", "Record Deleted");
           }
           else
           {
              showMessage("Error", "Invalid Rollno");
           }
           clearText();
        }
        // Updating a record in the Student table
        if(view==Update)
        {
           // Checking for empty roll number
           if(Rollno.getText().toString().trim().length()==0)
           {
              showMessage("Error", "Please enter Rollno");
              return;
           }
           Cursor c=db.rawQuery("SELECT * FROM student WHERE
rollno='"+Rollno.getText()+"'", null);
           if(c.moveToFirst()) {
              db.execSQL("UPDATE student SET name='" + Name.getText() + "',marks='" +
Marks.getText() +
                  "' WHERE rollno='"+Rollno.getText()+"'");
              showMessage("Success", "Record Modified");
           }
           else {
              showMessage("Error", "Invalid Rollno");
           }
           clearText();
        }
        // Display a record from the Student table
        if(view==View)
        {
           // Checking for empty roll number
           if(Rollno.getText().toString().trim().length()==0)
           {
              showMessage("Error", "Please enter Rollno");
              return;
           }
           Cursor c=db.rawQuery("SELECT * FROM student WHERE
rollno='"+Rollno.getText()+"'", null);
           if(c.moveToFirst())
           {
              Name.setText(c.getString(1));
              Marks.setText(c.getString(2));
           }
           else
           {
              showMessage("Error", "Invalid Rollno");
              clearText();
           }
        }
        // Displaying all the records
        if(view==ViewAll)
        {
           Cursor c=db.rawQuery("SELECT * FROM student", null);
           if(c.getCount()==0)
           {
              showMessage("Error", "No records found");
              return;
           }
           StringBuffer buffer=new StringBuffer();
           while(c.moveToNext())
           {
              buffer.append("Rollno: "+c.getString(0)+"\n");
              buffer.append("Name: "+c.getString(1)+"\n");
              buffer.append("Marks: "+c.getString(2)+"\n\n");
           }
           showMessage("Student Details", buffer.toString());
        }
```

```
    }
    public void showMessage(String title,String message)
    {
        Builder builder=new Builder(this);
        builder.setCancelable(true);
        builder.setTitle(title);
        builder.setMessage(message);
        builder.show();
    }
    public void clearText()
    {
        Rollno.setText("");
        Name.setText("");
        Marks.setText("");
        Rollno.requestFocus();
    }
}
```

OUTPUT:



Database [~/Downloads/Database] - .../app/src/main/java/com/...

```
        // Creating database and table
        db=openOrCreateDatabase( name: "StudentDB", C
        db.execSQL("CREATE TABLE IF NOT EXISTS studer
    }
    public void onClick(View view)
    {
        // Inserting a record to the Student table
        if(view==Insert)
        {
            // Checking for empty fields
            if(Rollno.getText().toString().trim().ler
                    Name.getText().toString().trim().
                    Marks.getText().toString().trim()
            {
                showMessage( title: "Error",  message: "F
                return;
            }
            db.execSQL("INSERT INTO student VALUES('"
                    "','"+Marks.getText()+"');");
            showMessage( title: "Success",  message: "Rec
            clearText();
        }
        // Deleting a record from the Student table
```

```
Build: completed successfully at 7/7/19 11:52 PM    2 s 337 ms
  Run build /home/dell/Downloads/Database           1 s 750 ms
    Load build                                      2 ms
    Configure build                                 343 ms
    Calculate task graph                            33 ms
    Run tasks                                       1 s 304 ms
```

Install successful (9 minutes ago)

5622 chars, 160 line breaks 160:1 LF UTF-8 4 spaces

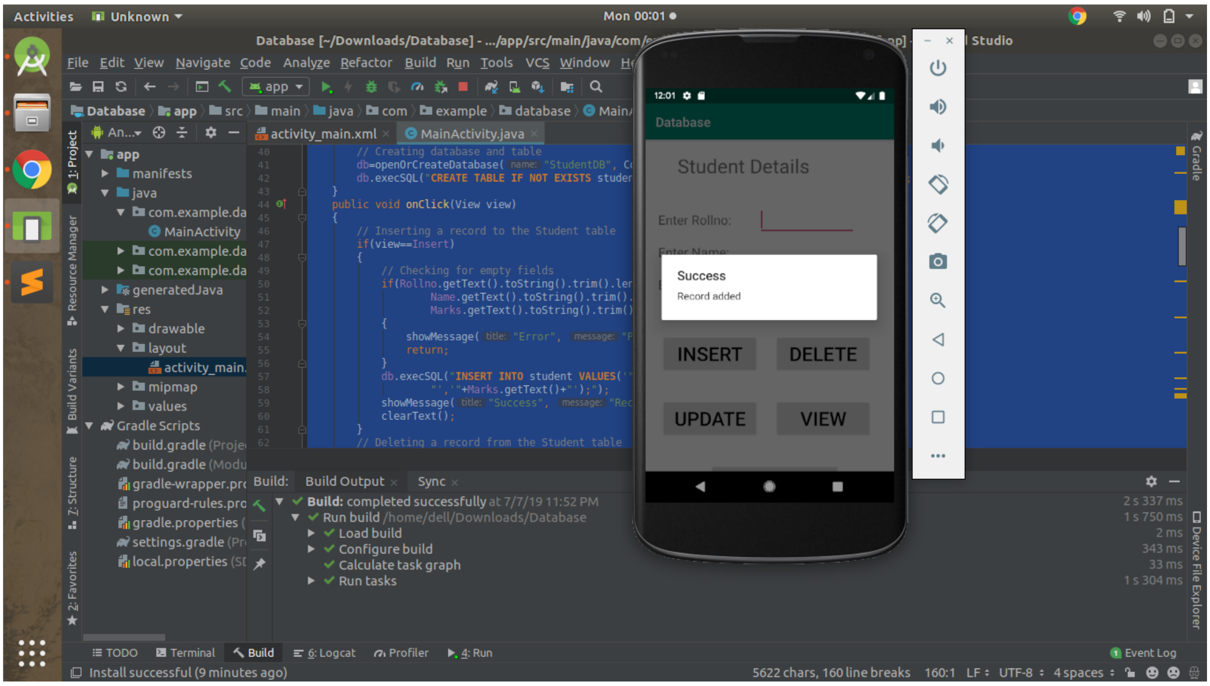
Database

Student Details

Enter Rollno: 3

Enter Name: Snehal

Enter Marks: 90



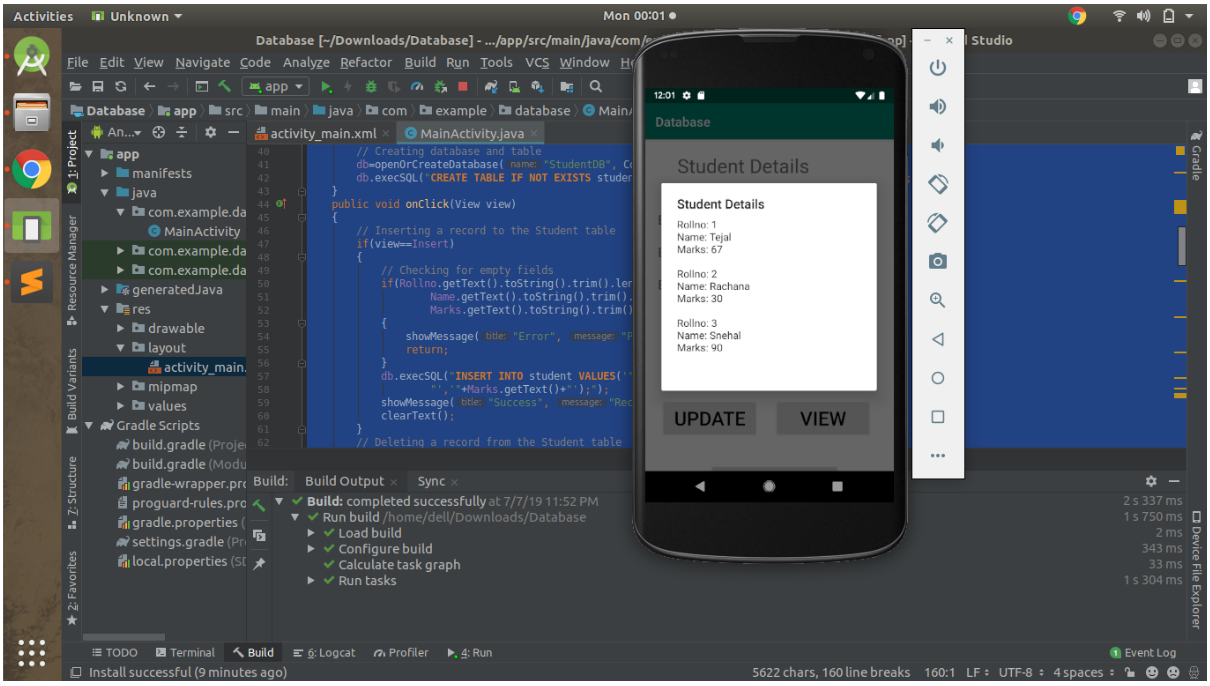
```
// Creating database and table
db=openOrCreateDatabase( name: "StudentDB", C
db.execSQL("CREATE TABLE IF NOT EXISTS studer
}
public void onClick(View view)
{
    // Inserting a record to the Student table
    if(view==Insert)
    {
        // Checking for empty fields
        if(Rollno.getText().toString().trim().ler
                Name.getText().toString().trim().
                Marks.getText().toString().trim()
        {
            showMessage( title: "Error",  message: "P
            return;
        }
        db.execSQL("INSERT INTO student VALUES('
                "','"+Marks.getText()+"');");
        showMessage( title: "Success",  message: "Rec
        clearText();
    }
    // Deleting a record from the Student table
```

```
Build: completed successfully at 7/7/19 11:52 PM          2 s 337 ms
  Run build /home/dell/Downloads/Database                 1 s 750 ms
    Load build                                                  2 ms
    Configure build                                           343 ms
    Calculate task graph                                       33 ms
    Run tasks                                              1 s 304 ms
```

Install successful (9 minutes ago)

Success

Record added



Student Details

Rollno: 1
Name: Tejal
Marks: 67

Rollno: 2
Name: Rachana
Marks: 30

Rollno: 3
Name: Snehal
Marks: 90

CONCLUSION:

Thus, Android Application that makes use of database is developed and executed successfully.

EXPERIMENT NO:8

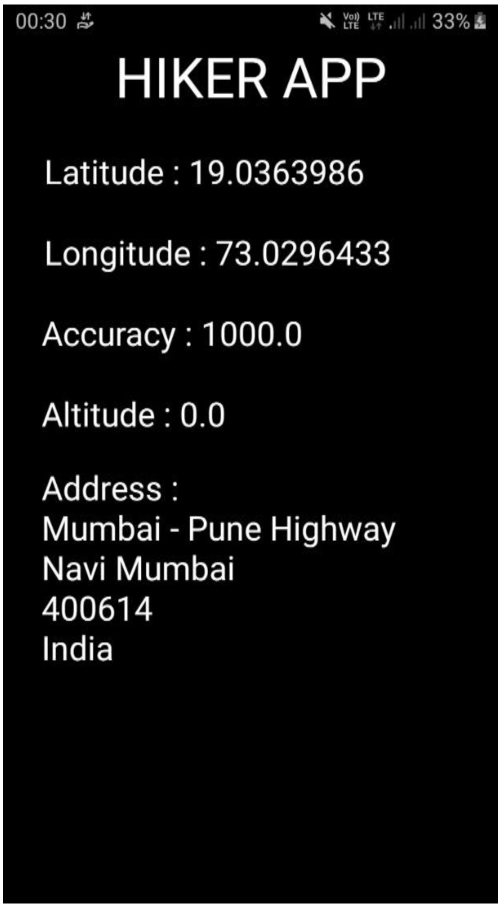
AIM-Develop a native application that uses GPS location information.

Program:

```
import android.Manifest;
import android.content.Context;
import android.content.pm.PackageManager;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.support.annotation.NonNull;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;
import android.widget.Toast;
import java.util.List;
import java.util.Locale;
public class MainActivity extends AppCompatActivity {
    LocationManager locationManager;
    LocationListener locationListener;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        locationManager = (LocationManager)
this.getSystemService(Context.LOCATION_SERVICE);
        locationListener = new LocationListener() {
            @Override
            public void onLocationChanged(Location location) {
                //Toast.makeText(MainActivity.this, location.toString(),
Toast.LENGTH_LONG).show();
                updateLocationDetails(location);
            }
            @Override
            public void onStatusChanged(String provider, int status, Bundle extras) { }
            @Override
            public void onProviderEnabled(String provider) { }
            @Override
            public void onProviderDisabled(String provider) { }
        };

if(ContextCompat.checkSelfPermission(this,Manifest.permission.ACCESS_FINE_LOCATI
ON) != PackageManager.PERMISSION_GRANTED){
            ActivityCompat.requestPermissions(this,new String[]
{Manifest.permission.ACCESS_FINE_LOCATION},1);
```

```
        }else{
           //
locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,0,0,locationL
istener);
locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER,0,0,lo
cationListener);
            Location lastKnownLocation =
locationManager.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
if(lastKnownLocation != null){
                updateLocationDetails(lastKnownLocation);
            }
        }
    }
    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
@NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        if(grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED){
           checkLocPermission();
        }
    }
    public void checkLocPermission(){

if(ContextCompat.checkSelfPermission(this,Manifest.permission.ACCESS_FINE_LOCATI
ON) == PackageManager.PERMISSION_GRANTED){
locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,0,0,locationL
istener);
        }
    }
    public void updateLocationDetails(Location location){
        //Toast.makeText(this, location.toString(), Toast.LENGTH_SHORT).show();
        TextView latitudeView = findViewById(R.id.latitudeView);
        TextView longitudeView =  findViewById(R.id.longitudeView);
        TextView accuracyView =  findViewById(R.id.accuracyView);
        TextView altitudeView = findViewById(R.id.altitudeView);
        TextView addressView = findViewById(R.id.addressView);
       latitudeView.setText("Latitude : "+Double.toString(location.getLatitude()));
        longitudeView.setText("Longitude : "+Double.toString(location.getLongitude()));
        accuracyView.setText("Accuracy : "+Double.toString(location.getAccuracy()));
        altitudeView.setText("Altitude : "+Double.toString(location.getAltitude()));
        String address = "Could not locate address..";
        Geocoder geocoder = new Geocoder(this, Locale.getDefault());
        try {
            List<Address> addressList =
geocoder.getFromLocation(location.getLatitude(),location.getLongitude(),1);
            if(addressList != null && addressList.size() > 0){
                address = "Address :\n";
                if(addressList.get(0).getThoroughfare() != null){
                    address +=  addressList.get(0).getThoroughfare() +"\n";
                }
                if(addressList.get(0).getLocality() != null){
                    address += addressList.get(0).getLocality() + "\n";
                }
                if(addressList.get(0).getPostalCode() != null){
                    address += addressList.get(0).getPostalCode() + "\n";
                }
                if(addressList.get(0).getCountryName() != null){
                    address += addressList.get(0).getCountryName();
                }
            }
        }catch (Exception e){
            e.printStackTrace();
        }
        addressView.setText(address);
    }
}
```

OUTPUT:



```
HIKER APP

Latitude : 19.0363986

Longitude : 73.0296433

Accuracy : 1000.0

Altitude : 0.0

Address :
Mumbai - Pune Highway
Navi Mumbai
400614
India
```

CONCLUSION:

Thus, Android Application that uses GPS location information is developed and executed successfully.

EXPERIMENT NO:9

AIM: -To develop an Android Application that creates an alert upon receiving a message.

PROCEDURE:

1. Open Eclipse IDE.
2. Create the project Ex_No_10.
3. Go to package explorer in the left-hand side. Select the project Ex_No_10.
4. Go to res folder and select layout. Double click the activity_main.xml file.
5. Now you can see the Graphical layout window.
6. This application has no components, because this just generates a notification alone.
7. Again go to package explorer in the left hand side. Select the project Ex_No_10.
8. Go to src folder. Double click the MainActivity.java file.
9. In java file write the activities done by the application such as receiving a message and notify it.
10. Get the following permissions in AndroidManifest.xml file:

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

11. Add Receiver class as receiver in AndroidManifest.xml file.
12. Finally run the android application.

PROGRAMS: activity_main.xml:

```
<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.ex_no_10.MainActivity" >
</RelativeLayout>
```

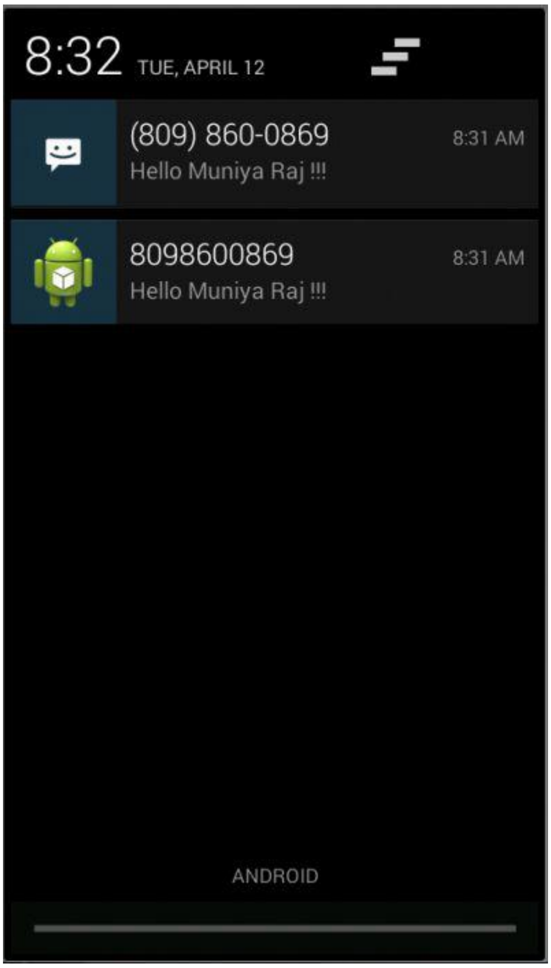
MainActivity.java:

```
package com.example.ex_no_10;
import android.support.v7.app.ActionBarActivity;
import android.app.Notification;
import android.app.NotificationManager;
import android.content.Context;
import android.os.Bundle;
public class MainActivity extends ActionBarActivity {
private static MainActivity inst;
public static MainActivity instance() {
// TODO Auto-generated method stub
return inst;
}p
ublic void onStart()
{
super.onStart();
inst=this;
}N
otificationManager nm;
Notification n;
@SuppressWarnings("deprecation")
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
nm=(NotificationManager)getSystemService(Context.NOTIFICATION_SERVICE);
n=new Notification(R.drawable.ic_launcher,"SMS Alert",System.currentTimeMillis());
}@
SuppressWarnings("deprecation")
public void update_notification(String no, String msg) {
// TODO Auto-generated method stub
n.setLatestEventInfo(getBaseContext(), no, msg, null);
nm.notify(1337, n);
}
}
```

Receiver.java:

```
package com.example.ex_no_10;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.telephony.SmsMessage;
public class Receiver extends BroadcastReceiver {
public static final String SMS_BUNDLE="pdus";
@Override
public void onReceive(Context arg0, Intent arg1) {
// TODO Auto-generated method stub
String no = null,msg = null;
Bundle b=arg1.getExtras();
if(b!=null)
{
Object[] sms=(Object[])b.get(SMS_BUNDLE);
for(int i=0;i<sms.length;++i)
{
SmsMessage sm=SmsMessage.createFromPdu((byte[])sms[i]);
no=sm.getOriginatingAddress();
msg=sm.getMessageBody().toString();
}M
ainActivity inst=MainActivity.instance();
inst.update_notification(no,msg);
}
}
}
```

OUTPUT:



8:32 TUE, APRIL 12

(809) 860-0869 8:31 AM
Hello Muniya Raj !!!

8098600869 8:31 AM
Hello Muniya Raj !!!

ANDROID

CONCLUSION:

Thus Android Application that creates an alert upon receiving a message is developed and executed successfully.

EXPERIMENT NO:10

AIM- Implementation of income tax/loan EMI calculator and deploy the same on real devices.

Program:

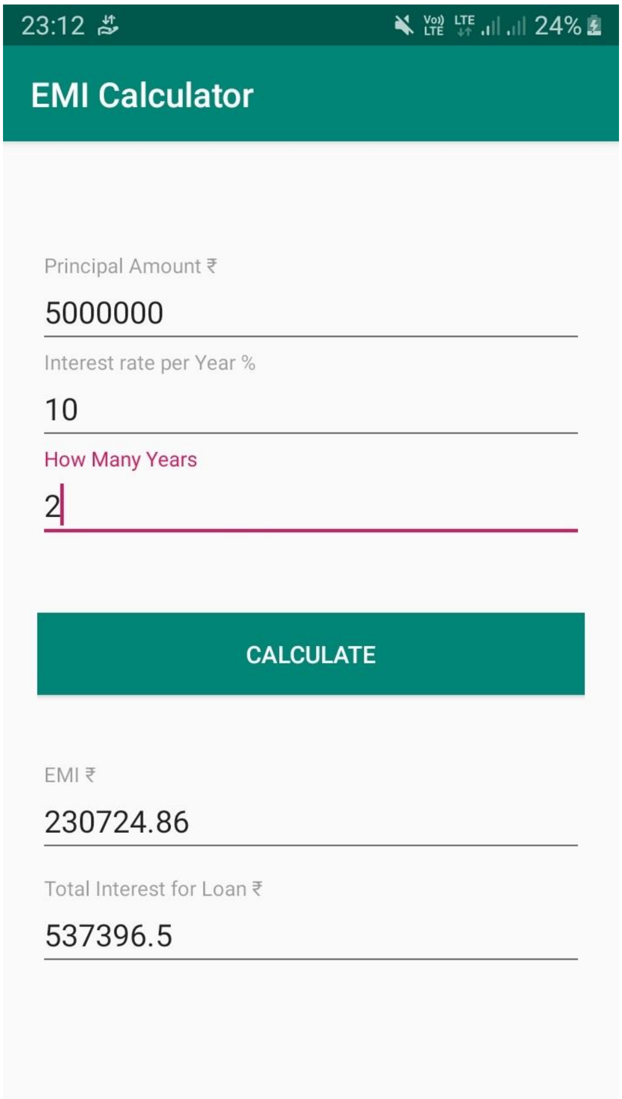
```
<resources>
   <string name="app_name">EMI Calculator</string>
   <string name="hint_principal">Principal Amount ?</string>
   <string name="hint_interest">Interest rate per Year %</string>
   <string name="hint_years">How Many Years</string>
   <string name="hint_emi">EMI ?</string>
   <string name="hint_interest_total">Total Interest for Loan ?</string>
</resources>
```

MainActivity.java

```
package com.example.exp9;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
public class MainActivity extends AppCompatActivity {
   Button emiCalcBtn;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);

      final EditText P = (EditText)findViewById(R.id.principal);
      final EditText I = (EditText)findViewById(R.id.interest);
      final EditText Y = (EditText)findViewById(R.id.years);
      final EditText TI = (EditText)findViewById(R.id.interest_total);
      final EditText result = (EditText)findViewById(R.id.emi) ;
      emiCalcBtn = (Button) findViewById(R.id.btn_calculate2);
      emiCalcBtn.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            String st1 = P.getText().toString();
            String st2 = I.getText().toString();
            String st3 = Y.getText().toString();
            if (TextUtils.isEmpty(st1)) {
               P.setError("Enter Prncipal Amount");
               P.requestFocus();
               return;
            }
            if (TextUtils.isEmpty(st2)) {
               I.setError("Enter Interest Rate");
               I.requestFocus();
```

```
                return;
            }
            if (TextUtils.isEmpty(st3)) {
                Y.setError("Enter Years");
                Y.requestFocus();
                return;
            }
            float p = Float.parseFloat(st1);
            float i = Float.parseFloat(st2);
            float y = Float.parseFloat(st3);
            float Principal = calPric(p);
            float Rate = calInt(i);
            float Months = calMonth(y);
            float Dvdnt = calDvdnt( Rate, Months);
            float FD = calFinalDvdnt (Principal, Rate, Dvdnt);
            float D = calDivider(Dvdnt);
            float emi = calEmi(FD, D);
            float TA = calTa (emi, Months);
            float ti = calTotalInt(TA, Principal);
            result.setText(String.valueOf(emi));
           TI.setText(String.valueOf(ti));
        }
    });
  }
  public  float calPric(float p) {
 return (float) (p);
  }
  public  float calInt(float i) {
     return (float) (i/12/100);
  }
  public  float calMonth(float y) {
     return (float) (y * 12);
  }
  public  float calDvdnt(float Rate, float Months) {
     return (float) (Math.pow(1+Rate, Months));
  }
  public  float calFinalDvdnt(float Principal, float Rate, float Dvdnt) {
     return (float) (Principal * Rate * Dvdnt);
  }
  public  float calDivider(float Dvdnt) {
     return (float) (Dvdnt-1);
  }
  public  float calEmi(float FD, Float D) {
     return (float) (FD/D);
  }

  public  float calTa(float emi, Float Months) {
     return (float) (emi*Months);
  }
  public  float calTotalInt(float TA, float Principal) {
        return (float) (TA - Principal);
    }
}
```

OUTPUT:



EMI Calculator

Principal Amount ₹
5000000

Interest rate per Year %
10

How Many Years
2

EMI ₹
230724.86

Total Interest for Loan ₹
537396.5

CONCLUSION:

Thus Android Application that creates income tax/loan EMI calculator developed and executed successfully.