MAD LAB EX NO. 3 – 6

Ex. No: 3 DRAWING GRAPHICAL PRIMITIVES

AIM

To develop a Simple Android Application that draws basic Graphical Primitives on the screen.

PROCEDURE

1. Creating a New project:
   - Open Android Studio and then click on **File -> New -> New project.**
   - Select the **Empty Activity** and click **Next and Finish**
   - **In MainActivity.java Import graphics package.**
   - Create a Bitmap, Setting the Bitmap as background for the ImageView, Creating the Canvas Object, Creating the Paint Object and set its color & TextSize
   - Draw Square,Rectangle,Circle and line
2. Designing Layout for Main Activity:
   - Click on **app -> res -> layout -> activity_main.xml.**
   - **Create relative layout and Image view.**

SOURCE CODE

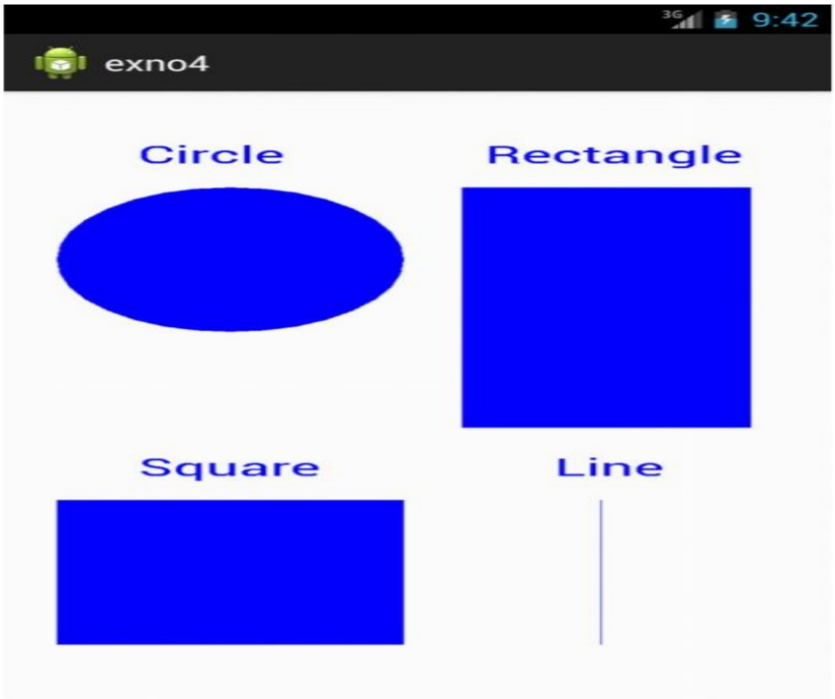
activity_main.xml

```
<?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

```
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



RESULT

Thus a simple android application to draw basic graphical primitives on the screen is implemented successfully.

Ex. No: 4

SIMPLE ANDROID APPLICATION THAT MAKES USE OF DATABASE

AIM

To develop a Simple Android Application that makes use of Database

PROCEDURE

1.Creating a New project:

- Open Android Studio and then click on **File -> New -> New project->Next->type application name->** select the **Empty Activity** and click **Next->Finish.**

Designing layout for the Android Application:

- Click on app -> res -> layout -> activity_main.xml.
- Create absolute layout with text views and buttons for retrieving the data from the database.
- Click on main activity and import android.database.sqlite.SQLiteDatabase.
- Create a method on create and set the content view.
- Create action listeners for buttons insert delete, update and view.
- Create database student and insert the values.
- Use cursors to retrieve rows from the database and display it on the application.
- Use if clause for the usage of buttons on click insert,delete,update and view.

SOURCE CODE

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_x="50dp"
        android:layout_y="20dp"
        android:text="Student Details"
        android:textSize="30sp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_x="20dp"
        android:layout_y="110dp"
        android:text="Enter Rollno:"
        android:textSize="20sp" />
```

```
<EditText
   android:id="@+id/Rollno"
   android:layout_width="150dp"
   android:layout_height="wrap_content"
   android:layout_x="175dp"
   android:layout_y="100dp"
   android:inputType="number"
   android:textSize="20sp" />

<TextView
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:layout_x="20dp"
   android:layout_y="160dp"
   android:text="Enter Name:"
   android:textSize="20sp" />

<EditText
   android:id="@+id/Name"
   android:layout_width="150dp"
   android:layout_height="wrap_content"
   android:layout_x="175dp"
   android:layout_y="150dp"
   android:inputType="text"
   android:textSize="20sp" />

<TextView
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:layout_x="20dp"
   android:layout_y="210dp"
   android:text="Enter Marks:"
   android:textSize="20sp" />

<EditText
   android:id="@+id/Marks"
   android:layout_width="150dp"
   android:layout_height="wrap_content"
   android:layout_x="175dp"
   android:layout_y="200dp"
   android:inputType="number"
   android:textSize="20sp" />

<Button
   android:id="@+id/Insert"
   android:layout_width="150dp"
   android:layout_height="wrap_content"
   android:layout_x="25dp"
   android:layout_y="300dp"
   android:text="Insert"
   android:textSize="30dp" />
```

```
    <Button
        android:id="@+id/Delete"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_x="200dp"
        android:layout_y="300dp"
        android:text="Delete"
        android:textSize="30dp" />


    <Button
        android:id="@+id/Update"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_x="25dp"
        android:layout_y="400dp"
        android:text="Update"
        android:textSize="30dp" />

    <Button
        android:id="@+id/View"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_x="200dp"
        android:layout_y="400dp"
        android:text="View"
        android:textSize="30dp" />

    <Button
        android:id="@+id/ViewAll"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:layout_x="100dp"
        android:layout_y="500dp"
        android:text="View All"
        android:textSize="30dp" />

</AbsoluteLayout>
```

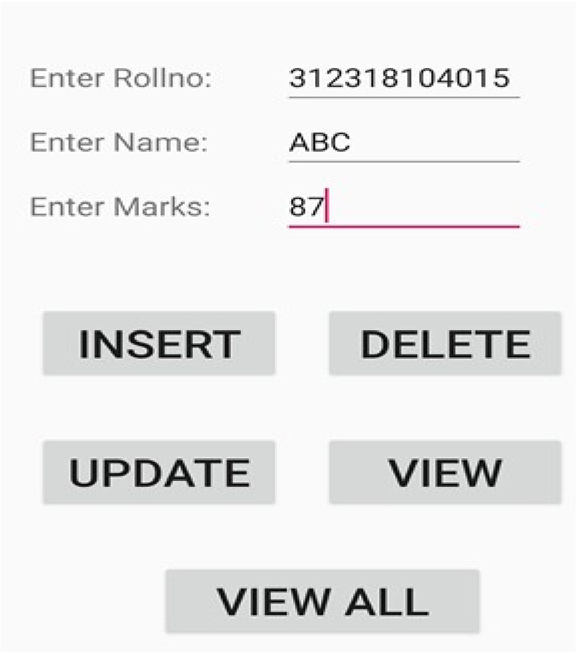
MainActivity.java

```
import android.app.Activity;
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 Activity implements OnClickListener
```

```
{
   EditText Rollno,Name,Marks;
   Button Insert,Delete,Update,View,ViewAll;
   SQLiteDatabase db;
   /** Called when the activity is first created. */
   @Override
   public void onCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);

      Rollno=(EditText)findViewById(R.id.Rollno);
      Name=(EditText)findViewById(R.id.Name);
      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();
    }
}
```

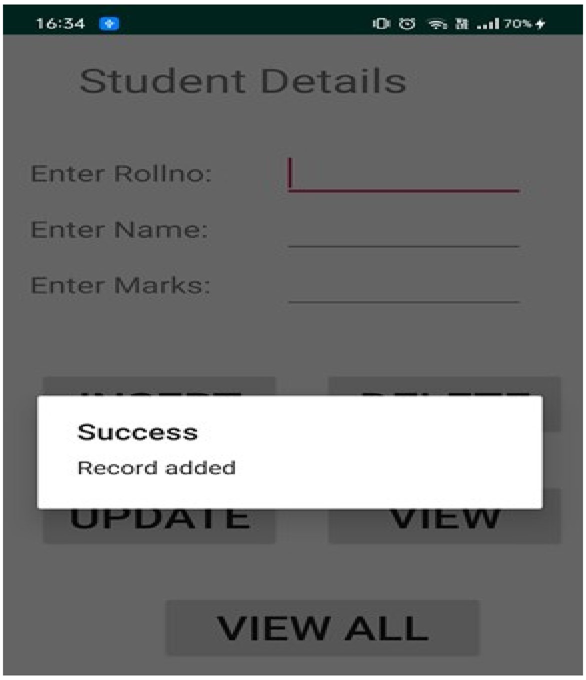
OUTPUT:



Enter Rollno: 312318104015

Enter Name: ABC

Enter Marks: 87



Student Details

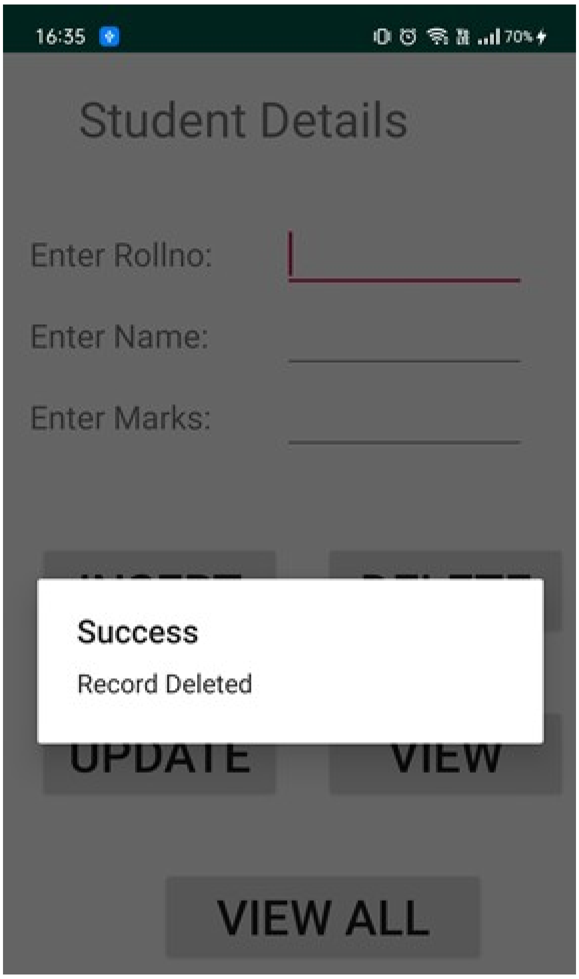
Enter Rollno:

Enter Name:

Enter Marks:

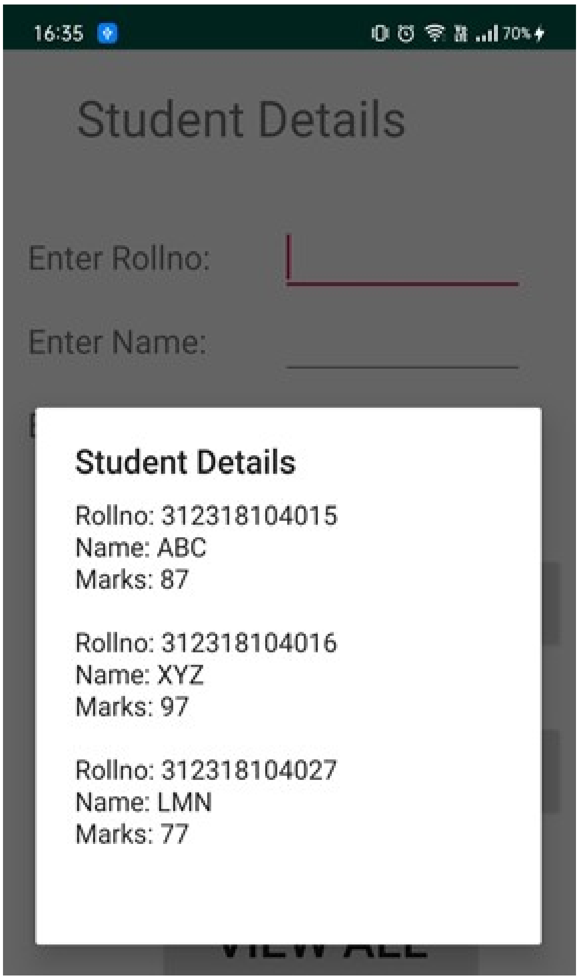
Success

Record added



Success

Record Deleted



Student Details

Rollno: 312318104015
Name: ABC
Marks: 87

Rollno: 312318104016
Name: XYZ
Marks: 97

Rollno: 312318104027
Name: LMN
Marks: 77

RESULT

Thus a simple android application that makes use of database is implemented successfully.

Ex. No: 5 NOTIFICATION MANAGER

AIM

To develop a Simple Android Application for notification using Notification manager.

PROCEDURE

1. Creating a New project:
   - Open Android Stdio and then click on File -> New -> New project.
   - Select the Empty Activity and click Next and Finish.
   - In MainActivity.java import notification manager and notification channel
   - Create button and set on click listener for notification
   - Create two methods makeNotificationChannel() and issueNotification()
   - When the button it clicked notification will be displayed with help of notification channel and manager.

2. Designing Layout for Main Activity:
    - Click on app -> res -> layout -> activity_main.xml.
    - Create Constraint layout and buttons.

SOURCE CODE

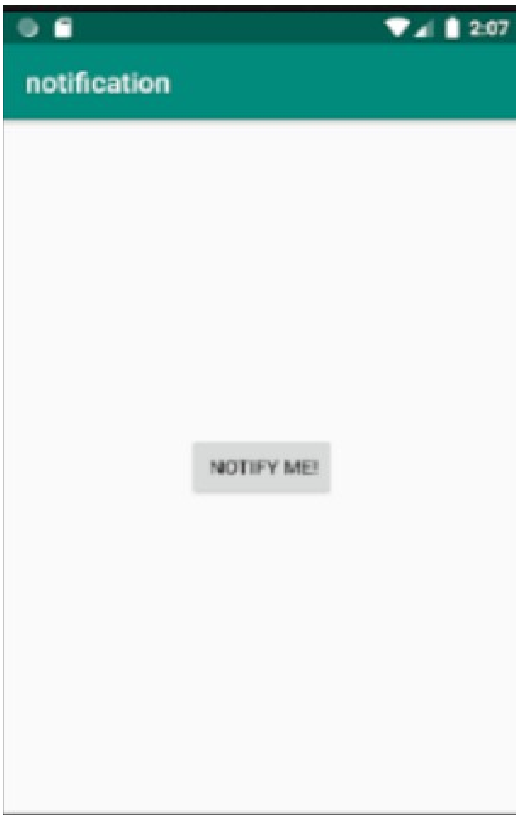
activity_main.xml

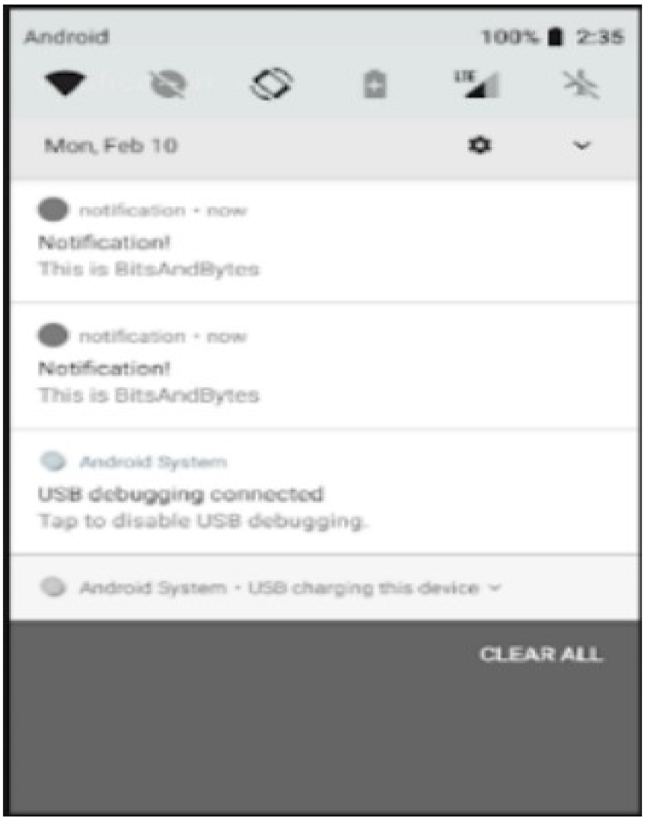
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <Button
        android:id="@+id/notify"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Notify me!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.ex555;
import androidx.appcompat.app.AppCompatActivity;
import androidx.annotation.RequiresApi;
import androidx.core.app.NotificationCompat;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.content.Context;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button notify = findViewById(R.id.notify);
        notify.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                issueNotification();
            }
        });
    }
    @RequiresApi(api = Build.VERSION_CODES.O)
    void makeNotificationChannel(String id, String name, int importance)
    {
        NotificationChannel channel = new NotificationChannel(id, name,
importance);
        channel.setShowBadge(true);
        NotificationManager notificationManager =
                (NotificationManager)getSystemService(NOTIFICATION_SERVICE);
        assert notificationManager != null;
        notificationManager.createNotificationChannel(channel);
    }
    void issueNotification()
    {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            makeNotificationChannel("CHANNEL_1", "Example channel",
NotificationManager.IMPORTANCE_DEFAULT);
        }
        NotificationCompat.Builder notification =
                new NotificationCompat.Builder(this, "CHANNEL_1")
                        .setSmallIcon(R.mipmap.ic_launcher)
                        .setContentTitle("Notification!")
                        .setContentText("This is BitsAndBytes")
                        .setNumber(3);
        NotificationManager notificationManager =
                (NotificationManager)getSystemService(NOTIFICATION_SERVICE);
        assert notificationManager != null;
        notificationManager.notify(1, notification.build());
        notificationManager.notify(2, notification.build());
    }
}
```

OUTPUT





notification • now
Notification!
This is BitsAndBytes

notification • now
Notification!
This is BitsAndBytes

Android System
USB debugging connected
Tap to disable USB debugging.

Android System • USB charging this device

RESULT

Thus a simple android application for notification using notification manager is implemented successfully.

Ex. No: 6 **MULTITHREADING**

AIM

To develop a Simple Android Application that implements Multithreading.

PROCEDURE

Creating a New project:

- Open Android Studio and then click on File -> New -> New project. type the Application name and click Next.
- Select the Empty Activity and click Next.

Designing layout for the Android Application:

- Click on app -> res -> layout -> activity_main.xml

1. Copy the Images and Paste it in “app -> res -> drawable” by pressing “right click mouse button on *drawable*” and selecting the “*Paste*” option.

SOURCE CODE

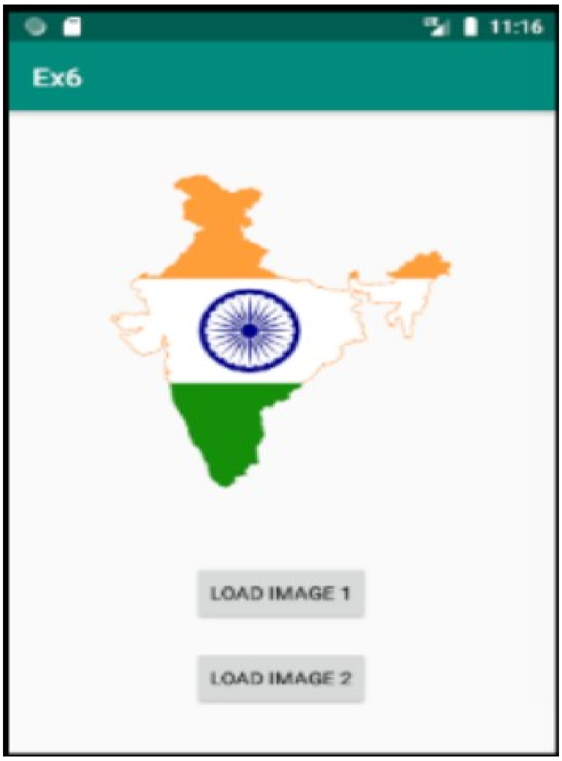
activity_main.xml

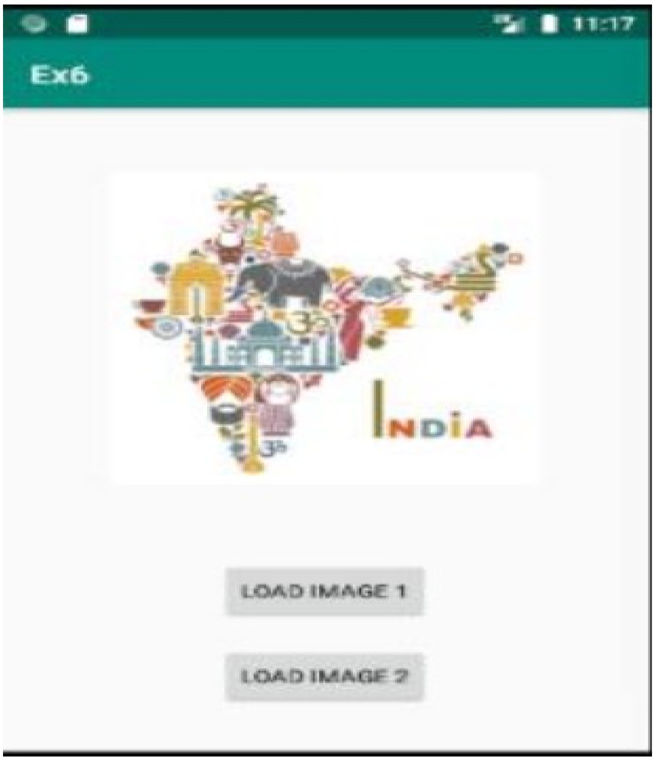
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="250dp"
        android:layout_height="250dp"
        android:layout_margin="50dp"
        android:layout_gravity="center" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_gravity="center"
        android:text="Load Image 1" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_gravity="center"
        android:text="Load image 2" />
 </LinearLayout>
```

MainActivity.java

```
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
public class MainActivity extends AppCompatActivity
{
    ImageView img;
    Button bt1,bt2;
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        bt1 = (Button)findViewById(R.id.button);
        bt2= (Button) findViewById(R.id.button2);
        img = (ImageView)findViewById(R.id.imageView);
        bt1.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                new Thread(new Runnable(){
                    @Override
                    public void run(){
                        img.post(new Runnable(){
                            @Override
                            public void run()
                            {
                                img.setImageResource(R.drawable.india1);
                            }
                        });
                    }
                }).start();
            }
        });
        bt2.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v){
                new Thread(new Runnable()
                {
                    @Override
                    public void run()
                    {
                        img.post(new Runnable()
                        {
                            @Override
                            public void run()
                            {
                                img.setImageResource(R.drawable.india2);
                            }
                        });
                    }
                }).start();
            }
        });}}
```

OUTPUT





RESULT

Thus a simple android application that uses multithreading are implemented successfully.

Ex. No: 7 **GPS LOCATION**

AIM

To develop an Android Application that uses GPS location information.

PROCEDURE

1. Creating a New project:
   - Open Android Studio and then click on File -> New -> New project.
   - Then type the Application name and click Next.
   - Then select the Minimum SDK as shown below and click Next.
   - Then select the Empty Activity and click Next.
   - Finally click Finish.
2. Designing layout for the Android Application:
   - Click on app -> res -> layout -> activity_main.xml
   - Type the Code for Activity_main.xml with button in relative layout.
3. Java Coding for the Android Application:

- Click on app -> java -> MainActivity.
- Create button for show location
- Set Toast for displaying latitude and longitude information.
- Call GPSTrace.java for getting GPS location by creating Location manager by enabling GPS provider and network provider.

4. Adding permissions in Manifest for the Android Application:
   - Click on app -> manifests -> AndroidManifest.xml.
   - Now include the INTERNET permissions in the AndroidManifest.xml file.

Selection of the location

Click three dots above the android and select the location. Now the area is indicated in red color. Click SAVEPOINT and note the latitude n longitude given in the right side. Finally click SET LOCATION and run the code again. Click showlocation in the android. It will display the latitude & longitude.

SOURCE CODE

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <Button
        android:id="@+id/show_Location"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Show Location"
        android:layout_centerVertical="true"
        android:layout_centerHorizontal="true" />
</RelativeLayout>
```

MainActivity.java

```
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.Manifest;
import android.content.Context;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    Button btnShowLocation;
    Context mContext;
    GPSTracker gps;
    @Override
    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mContext = this;
        btnShowLocation=findViewById(R.id.show_Location);
        btnShowLocation.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (ContextCompat.checkSelfPermission(mContext,
                    Manifest.permission.ACCESS_FINE_LOCATION) !=
                PackageManager.PERMISSION_GRANTED &&
                ActivityCompat.checkSelfPermission(mContext,
                Manifest.permission.ACCESS_COARSE_LOCATION) !=
                PackageManager.PERMISSION_GRANTED) {

                    ActivityCompat.requestPermissions(MainActivity.this,new
                String[]{Manifest.permission.ACCESS_FINE_LOCATION}, 1);

                }
            else {
                    gps = new GPSTracker(mContext, MainActivity.this);
                    // Check if GPS enabled
                    if (gps.canGetLocation()) {
                        double latitude = gps.getLatitude();
                        double longitude = gps.getLongitude();

                        // \n is for new line
                        Toast.makeText(getApplicationContext(), "Yoshow_Locationur
Location is - \nLat: " + latitude + "\nLong: " + longitude,
Toast.LENGTH_LONG).show();
                    } else {
                        // Can't get location.
                        // GPS or network is not enabled.
                        // Ask user to enable GPS/network in settings.
                        Toast.makeText(mContext, "Location Services off!",
Toast.LENGTH_SHORT).show();
                    }
                }
            }
        });
    }
    }
```

GPSTracker.java

```
   package com.example.ex7;
   import android.Manifest;
   import android.app.Activity;
   import android.app.Service;
   import android.content.Context;
   import android.content.DialogInterface;
```

```
    import android.content.Intent;
    import android.content.pm.PackageManager;
    import android.location.Location;
    import android.location.LocationListener;
    import android.location.LocationManager;
    import android.os.Bundle;
    import android.os.IBinder;
    import android.provider.Settings;
    import android.util.Log;
    import androidx.appcompat.app.AlertDialog;
    import androidx.core.app.ActivityCompat;
    import androidx.core.content.ContextCompat;
    public class GPSTracker extends Service {
        private Context mContext;
        // Flag for GPS status
        boolean isGPSEnabled = false;
        // Flag for network status
        boolean isNetworkEnabled = false;
        // Flag for GPS status
        boolean canGetLocation = false;
        Location location; // Location
        double latitude; // Latitude
        double longitude; // Longitude
        // The minimum distance to change Updates in meters
        private static final long MIN_DISTANCE_CHANGE_FOR_UPDATES = 1000; //
10 meters
        // The minimum time between updates in milliseconds
        private static final long MIN_TIME_BW_UPDATES = 1000 * 60 * 1; // 1
minute
        // Declaring a Location Manager
        protected LocationManager locationManager;
        Activity activity;
        public GPSTracker(Context context, Activity activity) {
            this.mContext = context;
            this.activity = activity;
            getLocation();
        }
        private boolean checkPermissions() {
            return ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) ==
PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) ==
PackageManager.PERMISSION_GRANTED;
            // If we want background location
            // on Android 10.0 and higher,
            // use:
            // ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_BACKGROUND_LOCATION) ==
PackageManager.PERMISSION_GRANTED
        }
        public Location getLocation() {
```

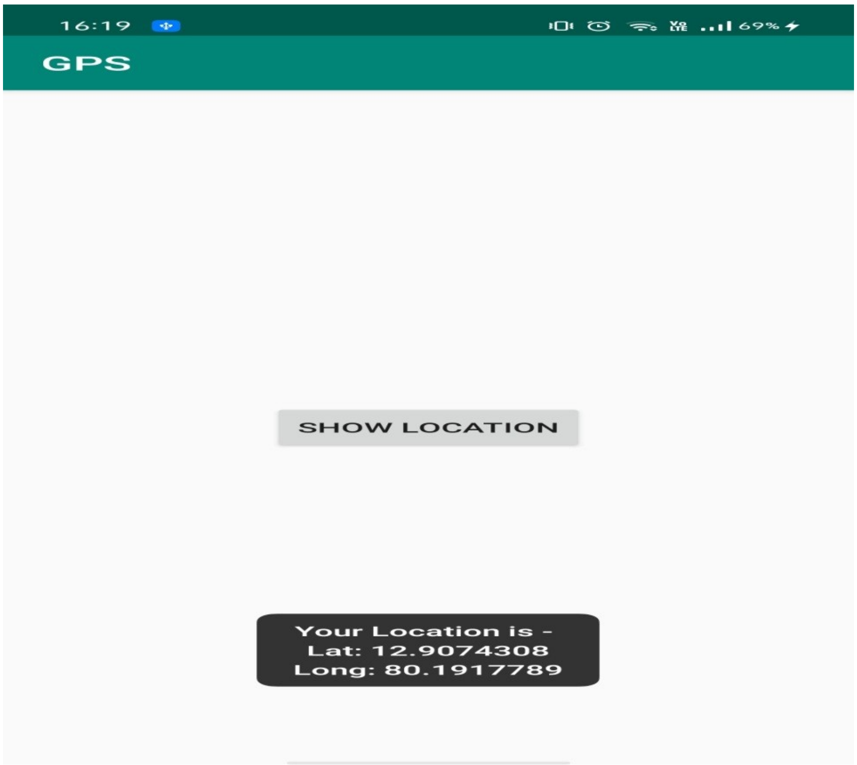
```
        try {
            locationManager = (LocationManager)
mContext.getSystemService(LOCATION_SERVICE);
            // Getting GPS status
            isGPSEnabled = locationManager
                    .isProviderEnabled(LocationManager.GPS_PROVIDER);
            // Getting network status
            isNetworkEnabled = locationManager
                    .isProviderEnabled(LocationManager.NETWORK_PROVIDER);
            if (!isGPSEnabled && !isNetworkEnabled) {
                // No network provider is enabled
            } else {
                this.canGetLocation = true;
                if (isNetworkEnabled) {
                    int requestPermissionsCode = 50;

locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVID
ER, MIN_TIME_BW_UPDATES, MIN_DISTANCE_CHANGE_FOR_UPDATES,
mLocationListener);
                    Log.d("Network", "Network");
                    if (locationManager != null) {
                        location =

locationManager.getLastKnownLocation(LocationManager.NETWORK_PROVIDER
);
                        if (location != null) {
                            latitude = location.getLatitude();
                            longitude = location.getLongitude();
                        }
                    }
                }
            }
            // If GPS enabled, get latitude/longitude using GPS Services
            if (isGPSEnabled) {
                if (location == null) {
                    if (ContextCompat.checkSelfPermission(activity,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(activity,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
                        ActivityCompat.requestPermissions(activity, new String[]
{Manifest.permission.ACCESS_FINE_LOCATION}, 50);
                    } else {
                        locationManager.requestLocationUpdates(
                                LocationManager.GPS_PROVIDER,
                                MIN_TIME_BW_UPDATES,
                                MIN_DISTANCE_CHANGE_FOR_UPDATES,
mLocationListener);
                        Log.d("GPS Enabled", "GPS Enabled");
                        if (locationManager != null) {
                            location = locationManager
```

```
                                    .getLastKnownLocation(LocationManager.GPS_PROVID
ER);
                            if (location != null) {
                                latitude = location.getLatitude();
                                longitude = location.getLongitude();
                            }
                        }
                    }
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return location;
    }
    private final LocationListener mLocationListener = new LocationListener()
{
        @Override
        public void onLocationChanged(final Location location) {
            if (location != null) {
                latitude = location.getLatitude();
                longitude = location.getLongitude();
            }
        }
        @Override
        public void onStatusChanged(String provider, int status, Bundle extras)
{
        }
        @Override
        public void onProviderEnabled(String provider) {
        }
        @Override
        public void onProviderDisabled(String provider) {
        }
    };
    /**
     * Function to get latitude*
     */
    public double getLatitude() {
        if (location != null) {
            latitude = location.getLatitude();
        }
        // return latitude
        return latitude;
    }
    /**
     * Function to get longitude
     */
    public double getLongitude() {
        if (location != null) {
            longitude = location.getLongitude();
        }
        return longitude;
    }
    /**
     * Function to check GPS/Wi-Fi enabled
     *
     * @return boolean
     */
    public boolean canGetLocation() {
        return this.canGetLocation;
    }
    /**
     * Function to show settings alert dialog.
     * On pressing the Settings button it will launch Settings Options.
     */
    @Override
    public IBinder onBind(Intent arg0) {
        return null;
     }
    }
```

Activitymanifest.xml

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

OUTPUT



Your Location is -
Lat: 12.9074308
Long: 80.1917789

RESULT

Thus a simple android application that uses GPS location information is implemented successfully.