23/1/24

Om Shete
THADOMAL SHAHANI TSEC ENGINEERING COLLEGE
TE C32
2103163

# Experiment No: 1

**Aim:** Write an android application to draw basic 2D graphical primitives

**Theory:**

- **Android Studio**

⇒ Android studio is the official Integrated Development Environment (IDE) for android application development.

- Android studio provides more features that enhances our productivity while building Android apps.
- The features of android studio are :-

1) It has a flexible gradle-based build system
2) It has a fast and feature-rich emulator for app testing.
3) It supports C++ and NDK.
4) It provides build-in supports for Google cloud platform.

- **2D primitives**

⇒ In mobile computing, 2D primitives refers to basic graphical elements that are used to create 2D graphics and user interfaces on mobile devices.

- These primitives are essential for creating visually appealing and interactive mobile applications.

- Some of the common 2D primitives are :-

1) Points
⇒ The most basic primitive, representing a single pixel on the screen. Points are the building blocks for other shapes.

2) Lines
⇒ A sequence of connected points forming a straight path.

3) Rectangles
⇒ Defined by four points, representing a four-sided polygon with equal angles at each corner.

4) Circles and Ellipses
⇒ Represented by their center point and radius or two radii.

5) Polygons
⇒ Multi-sided shapes formed by connecting multiple points.

- XML

⇒ XML stands for the extensible Markup Language is a mark up language.
- XML is designed to store and transport the data. XML is designed to be self-descriptive.
- XML is a platform independent and language independent.
- XML tags are not predefined. We must define our own tags.
- XML is designed to carry data, not to display data. XML is not a replacement for HTML.

- Functions

1) Bitmap

⇒ A 'Bitmap' is a representation of graphic image as a pixel array. It is often used to load, manipulate and display images in Android apps.
- Bitmap can be created from various sources, including resources, files or dynamically generated.

2) Canvas

⇒ The 'Canvas' class is used to draw graphics on an Android device. It provides methods to draw various shapes, text and images onto a Bitmap or the screen.

3) ImageView

⇒ An "ImageView" is a UI widget in Android used to display images. It can be used to load and display Bitmaps, drawables or other images sources.

4) Paint

⇒ The 'Paint' class is used to define how to draw graphical elements, such as color, style, stroke width, and text attributes.

5) Color

⇒ The 'Color' class in Android provides methods to work with colors.

- Conclusion :

In this experiment, we learnt about the tool android studio and how to draw 2D primitives in android studio

Gaur A+ 23/1/24

Code:

MainActivity.java

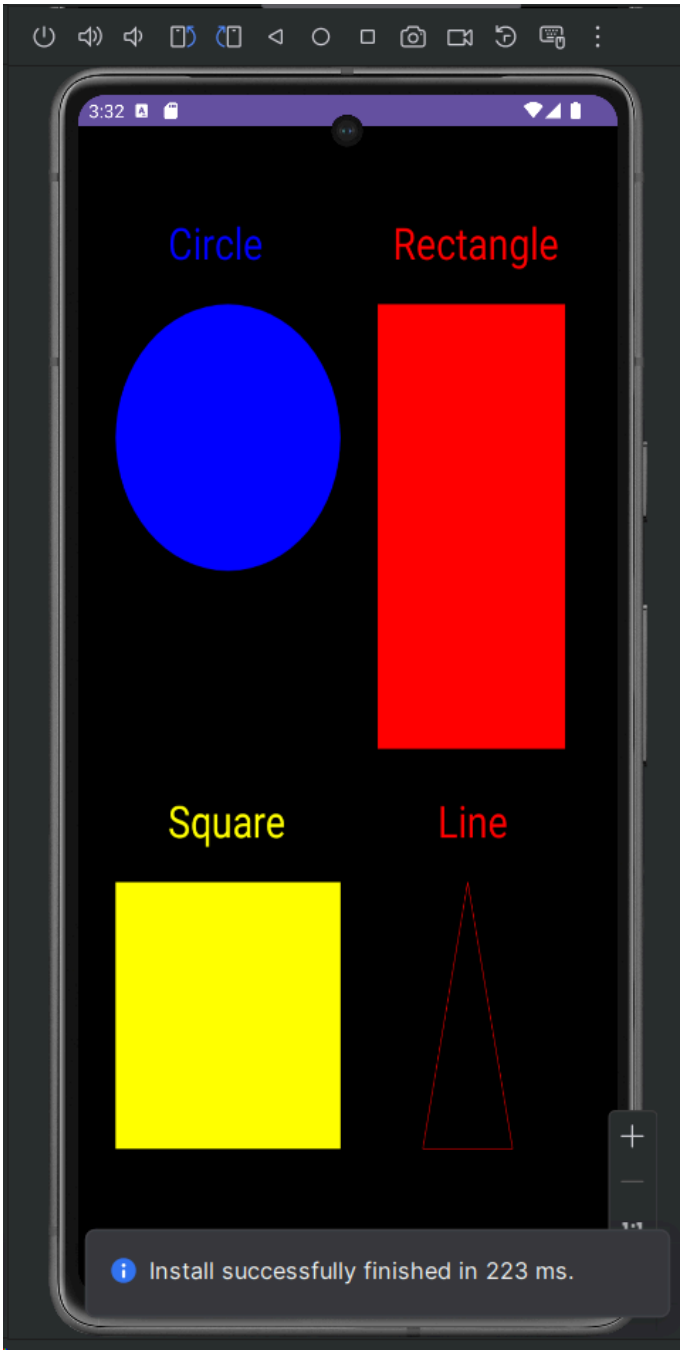
```
package com.example.exp1_mcc;

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.RGB_565);

       //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.RED);
       paint.setTextSize(50);

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

       paint.setColor(Color.BLUE);

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

       paint.setColor(Color.YELLOW);

       //To draw a Square
```

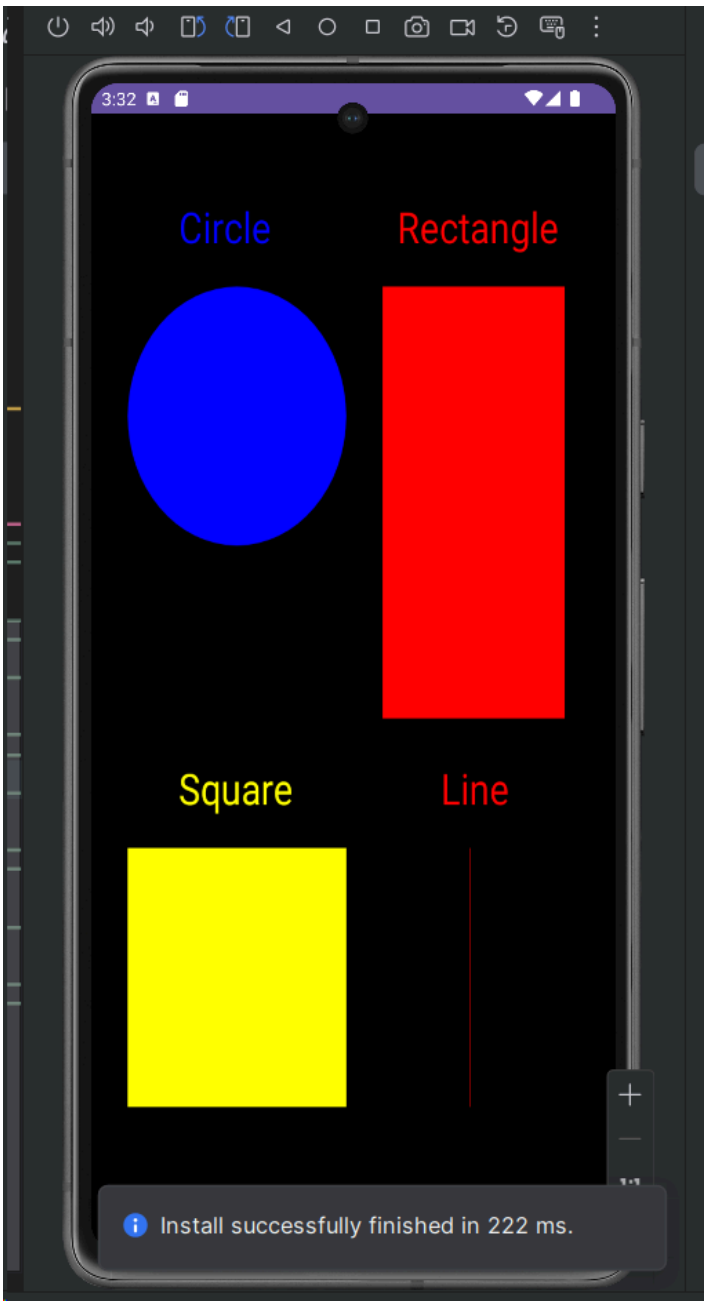
```
        canvas.drawText("Square", 120, 800, paint);
        canvas.drawRect(50, 850, 350, 1150, paint);


        paint.setColor(Color.RED);

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

        //To draw a Line
        canvas.drawText("Line", 480, 800, paint);
        canvas.drawLine(520, 850, 460, 1150, paint);
        canvas.drawLine(520, 850, 580, 1150, paint);
        canvas.drawLine(460, 1150, 580, 1150, paint);

    }
}
```

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>
```

Output:


3:32
Circle
Rectangle
Square
Line
Install successfully finished in 223 ms.


3:32
Circle
Rectangle
Square
Line
Install successfully finished in 222 ms.

# Experiment No: 2

**Aim:** WAA to draw basic graphical 3D primitives.

**Theory:**

- **Parameters used in the Program**

1) **cubeSize**
⇒ Represents the size of the cube. It is used to specify the dimensions when drawing a cube.

2) **cuboidWidth**
⇒ Represents the width of the cuboid. It is used to specify the width when drawing a cuboid.

3) **cubeHeight**
⇒ Represents the height of the cuboid. It is used to specify the height when drawing a cuboid.

- **3D primitives**

1) **Cube**
⇒ Drawn using 'canvas.drawRect()' in a loop. The loop iterates 100 times, drawing multiple rectangles with increasing positions to simulate a 3D effect.

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

2) Cuboid

⇒ Draw using combinations of 'canvas.drawLines()' to create a rectangular prism.

- Lines are drawn to represent the edges of cuboid forming a 3-D shape.

- Layout in XML

⇒ We have used the relative layout in xml file.

- Relative layout defines positions child views relative to each other or the parent.

- Conclusion :

Thus a simple Android application that draws basic Graphical Primitives on the screen is developed and executed successfully.

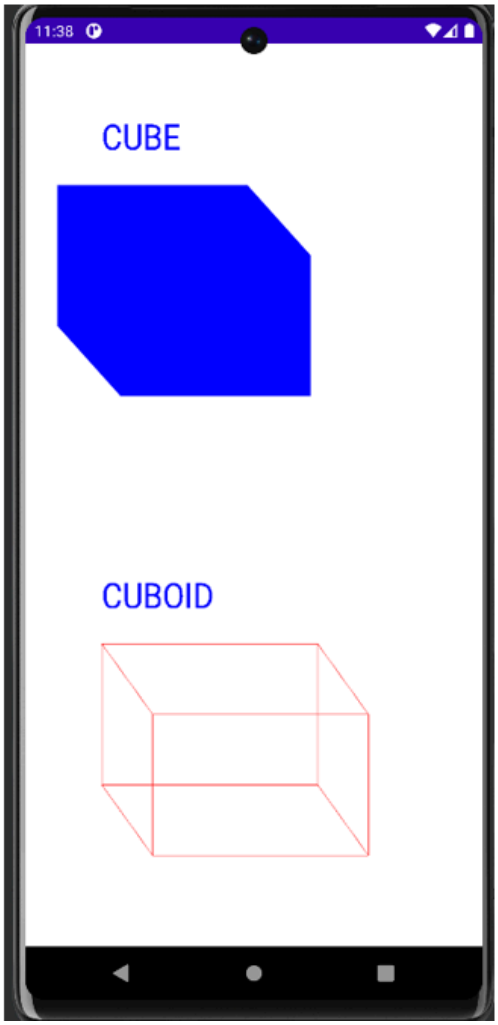
Sign (AT)

## Code:

```
package com.example.mccexp2;

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);

	int cubeSize = 200;
	int cuboidWidth = 300;
	int cuboidHeight = 150;

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

	//Setting the Bitmap as background for the ImageView
	ImageView i = 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);

	Paint paint1 = new Paint();
	paint1.setColor(Color.RED);
	paint1.setTextSize(50);

	canvas.drawText("CUBE", 120, 150, paint);
	for(int it=0; it<100; it++){
		canvas.drawRect(50+it, 200+it, 350+it, 400+it, paint);
		paint1.setColor(Color.RED);
	}

	canvas.drawText("CUBOID", 120, 800, paint);
```

```
        canvas.drawLine(120,850,120,1050, paint1);
        canvas.drawLine(460,850,460,1050, paint1);
        canvas.drawLine(120,850,460,850, paint1);
        canvas.drawLine(120,1050,460,1050, paint1);

        canvas.drawLine(200,950,200,1150, paint1);
        canvas.drawLine(540,950,540,1150, paint1);
        canvas.drawLine(200,950,540,950, paint1);
        canvas.drawLine(200,1150,540,1150, paint1);

        canvas.drawLine(120,850,200,950, paint1);
        canvas.drawLine(460,850,540,950, paint1);
        canvas.drawLine(120,1050,200,1150, paint1);
        canvas.drawLine(460,1050,540,1150, paint1);


        }
}
```

**Output:**



# Experiment No : 3

**Aim :** WAA to design a Form using GUI components

**Theory :**

- A login form is created using the <EditText /> XML element.
- The processing of data is designed in the Java file of the project.
- Clearning of data on pressing the 'Reset' button is done from Java and displaying error message using the built-in error method 'setError()' is done using the Java.android package.
- Forms in XML file as Java :

1) **Logout**

⇒ UI component are defined using XML files in logout directory.

- The logout XML file defines the structure and apperance of the user interface.
- Here the defined EditText fields for name, email, password along with buttons are present in XML logout.

2) **Java code**

⇒ MainActivity class extends AppcompatActivity, which is the base class for activities in Android.

- Initialize Edittext fields and buttons using findViewById() method, which finds the view by the specified Id defined in the XML logout.
- Set click defined listners for proceed and reset buttons using setOnclickListener() method.
- Inside the click listener, we handle the button clicks :-
- For the proceed button, we retreive text from EditText field, process the data as needed, and display it in a toast message using Toast.
- For resed button, we clear all the Edit Text fields.

## 3) Event Handling

⇒ User interaction such as button clicks triggers event that are handled by event listeners.
- SetOnClickListeners() method to set the event listeners for button clicks.
- Inside the event listners, we define actions to be performed when the corresponding button is clicked.

- Conclusion :

The experiment illustrates how to create a simple login form in android studio using event handlers.

**Code:**

(MainActivity.java)

```
package com.example.mcc_exp3;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;



public class MainActivity extends AppCompatActivity {

    private EditText editTextFirstName, editTextLastName,
editTextEmail, editTextPassword;



    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);



        editTextFirstName = findViewById(R.id.editTextFirstName);

        editTextLastName = findViewById(R.id.editTextLastName);

        editTextEmail = findViewById(R.id.editTextEmail);

        editTextPassword = findViewById(R.id.editTextPassword);



        Button buttonProceed = findViewById(R.id.buttonProceed);
```

```
        Button buttonReset = findViewById(R.id.buttonReset);


        buttonProceed.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String firstName =
editTextFirstName.getText().toString();
                String lastName =
editTextLastName.getText().toString();
                String email = editTextEmail.getText().toString();
                String password =
editTextPassword.getText().toString();


                String message = "First Name: " + firstName + "nLast
Name: " + lastName +
                        "nEmail: " + email + "nPassword: " + password;


                Toast.makeText(MainActivity.this, message,
Toast.LENGTH_SHORT).show();
            }
        });


        buttonReset.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
```

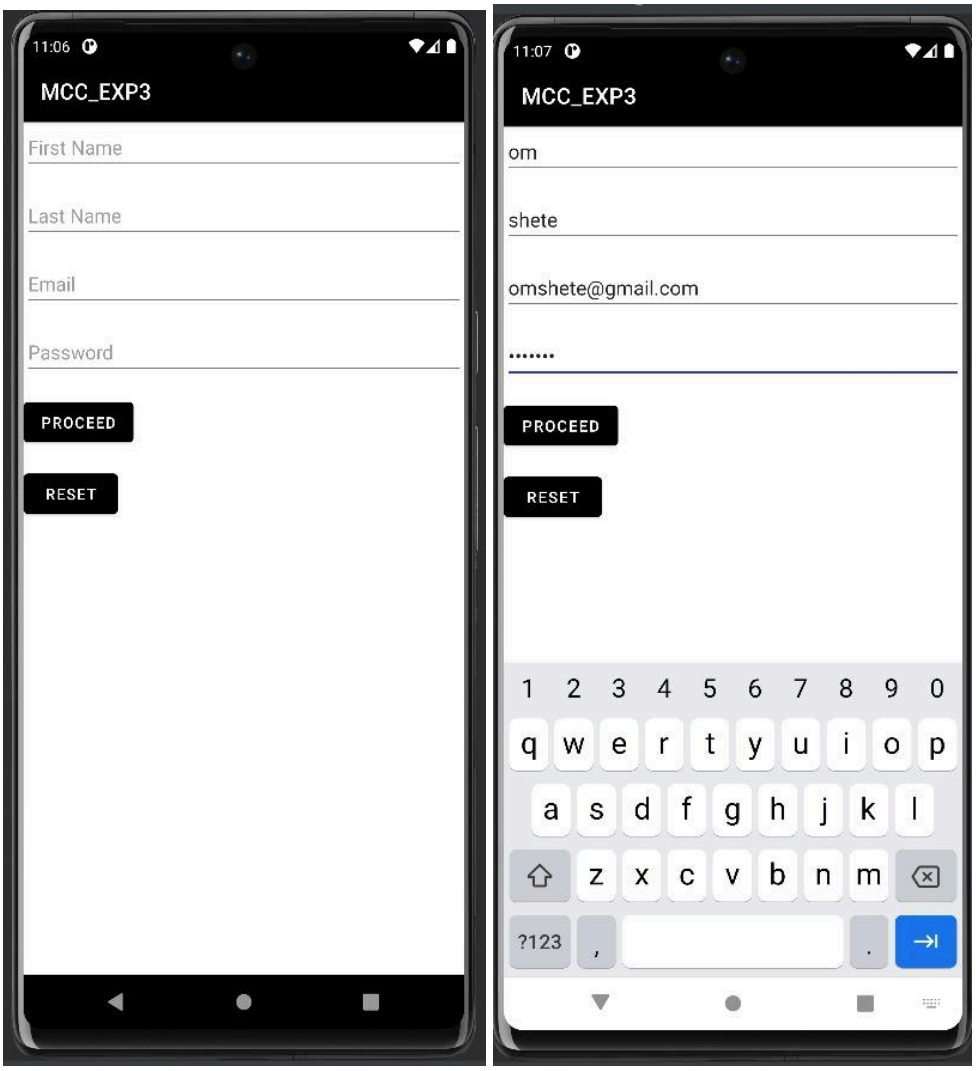
```
                editTextFirstName.setText("");

                editTextLastName.setText("");

                editTextEmail.setText("");

                editTextPassword.setText("");

            }

        });

    }

}
```

(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">

    <EditText
        android:id="@+id/editTextFirstName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="First Name"/>

    <EditText
        android:id="@+id/editTextLastName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/editTextFirstName"
        android:layout_marginTop="16dp"
        android:hint="Last Name"/>

    <EditText
        android:id="@+id/editTextEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```
        android:layout_below="@id/editTextLastName"
        android:layout_marginTop="16dp"
        android:inputType="textEmailAddress"
        android:hint="Email"/>

    <EditText
        android:id="@+id/editTextPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/editTextEmail"
        android:layout_marginTop="16dp"
        android:inputType="textPassword"
        android:hint="Password"/>

    <Button
        android:id="@+id/buttonProceed"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/editTextPassword"
        android:layout_marginTop="16dp"
        android:text="Proceed"/>

    <Button
        android:id="@+id/buttonReset"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/buttonProceed"
        android:layout_marginTop="16dp"
        android:text="Reset"/>
</RelativeLayout>
```

**Output:**


11:06
MCC_EXP3
First Name
Last Name
Email
Password
PROCEED
RESET
11:07
MCC_EXP3
om
shete
omshete@gmail.com
.....
PROCEED
RESET
1 2 3 4 5 6 7 8 9 0
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
?123

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE
Om Shete
TE C92
2103163

# Experiment No. 4

**Aim :** WAA to design GUI component using database.

**Theory :**

**Step 1 :** Create a new project

**Step 2 :** Add permission to access storage in old manifest.xml file.

**Step 3 :** Work on the ActivityMain.xml file <Button> are UI components that uses interaxucial role in the user interface, serving as entry point for various functionalities within an app.

<EditText> is a versatile UI component for capturing user input. It allows users to enter text, numbers, data types making it a fundamental element for forms and data input scenarios.

**Step 4 :** Creating a new java class for performing SQLite operations.

**Step 5 :** Working with the MainActivity.java file.

- Database in Android :

⇒ In android, we make use of a different version of SQL databases. The database we use in native Android application is SQLite.

- In this experiment, we use one java file for insertion and another Java file for retrieval.
- There is also a database helper file to facilitate CRUD operations.
- We define a schema (model) for storing the user details in the database.
- In this app, we take the details of a user via a form and save it in database for persistent storage.
- Once we restard the app, the details remain in the database, are not erased and can be retrieved conveniently.

- Conclusion :

Hence, we had successfully created a form and stored the data in the database.

Rjain (A+)

**Code:**

(activity_main.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    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">
    <ScrollView
        android:id="@+id/sv_registration_form"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_margin="10dp"
            android:orientation="vertical">
            <com.google.android.material.textfield.TextInputLayout
                android:id="@+id/til_name"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_marginTop="20dp">

<com.google.android.material.textfield.TextInputEditText
                    android:id="@+id/tiet_name"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:hint="Name" />
            </com.google.android.material.textfield.TextInputLayout>
            <com.google.android.material.textfield.TextInputLayout
                android:id="@+id/til_email"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_marginTop="20dp">

<com.google.android.material.textfield.TextInputEditText
                    android:id="@+id/tiet_email"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
```

```
                android:hint="Email"
                android:inputType="textEmailAddress"/>
        </com.google.android.material.textfield.TextInputLayout>
        <com.google.android.material.textfield.TextInputLayout
            android:id="@+id/til_contact_no"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_marginTop="20dp">

<com.google.android.material.textfield.TextInputEditText
                android:id="@+id/tiet_contact_no"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:hint="Contact No."
                android:inputType="number"
                android:maxLength="10"/>
        </com.google.android.material.textfield.TextInputLayout>
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Gender"
            android:textSize="18sp"
            android:layout_marginTop="20dp"/>
        <RadioGroup
            android:id="@+id/rg_gender"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" >
            <RadioButton
                android:id="@+id/rb_male"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Male"
                android:textSize="15sp"/>
            24

            <RadioButton
                android:id="@+id/rb_female"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Female"
                android:textSize="15sp" />
        </RadioGroup>
        <Button
```

```
                android:id="@+id/btn_insert_data"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Insert Data"
                android:textAllCaps="false"
                android:layout_gravity="center"
                android:layout_marginTop="20dp"/>
            <Button
                android:id="@+id/btn_view_users"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="View Users"
                android:textAllCaps="false"
                android:layout_gravity="center"
                android:layout_marginTop="20dp"/>
        </LinearLayout>
    </ScrollView>
</LinearLayout>
```

(MainActivity.java)

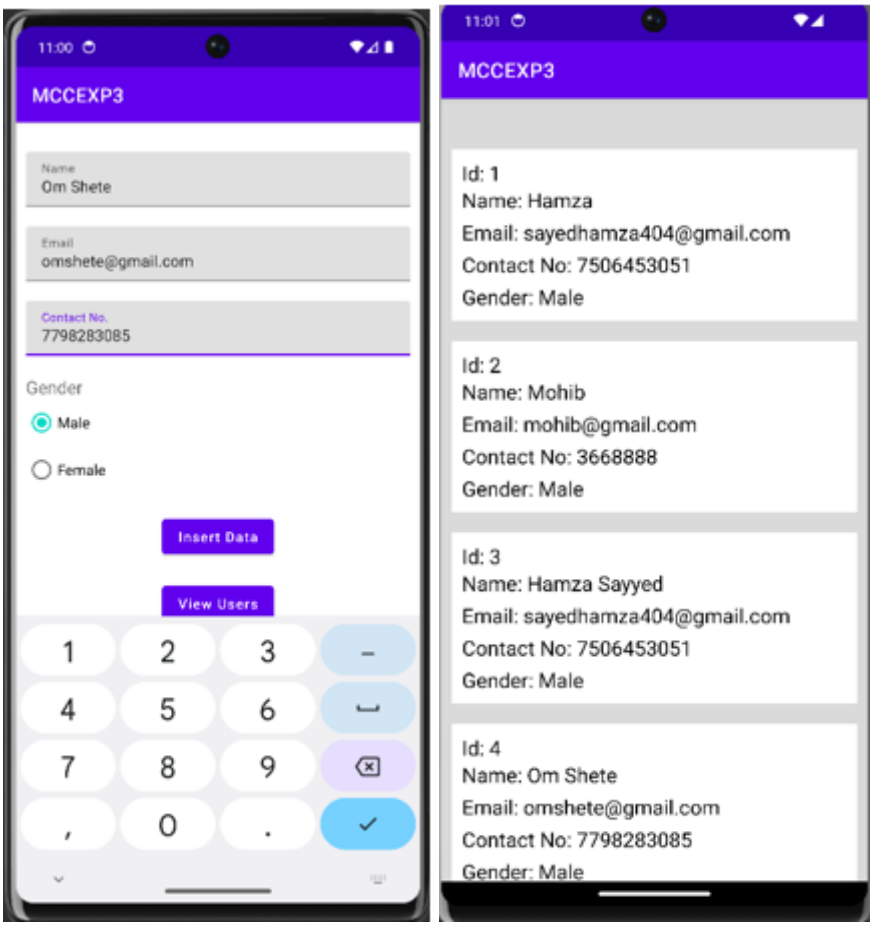
```
package com.example.mccexp3;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.RadioGroup;
import android.widget.ScrollView;
import android.widget.Toast;
import com.google.android.material.textfield.TextInputEditText;
import com.google.android.material.textfield.TextInputLayout;

public class MainActivity extends AppCompatActivity {
    Context context;
    DatabaseHelper databaseHelper;
    ScrollView svRegistrationForm;
    TextInputLayout tilName, tilEmail, tilContactNo;
    TextInputEditText tietName, tietEmail, tietContactNo;
    RadioGroup rgGender;
```

```
    Button btnSignUp, btnViewUsers;
    String name, email, contactNo, gender;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_first);
        context = MainActivity.this;
        databaseHelper = new DatabaseHelper(context);
        svRegistrationForm = findViewById(R.id.sv_registration_form);
        tilName = findViewById(R.id.til_name);
        tilEmail = findViewById(R.id.til_email);
        tilContactNo = findViewById(R.id.til_contact_no);
        tietName = findViewById(R.id.tiet_name);
        tietEmail = findViewById(R.id.tiet_email);
        tietContactNo = findViewById(R.id.tiet_contact_no);
        rgGender = findViewById(R.id.rg_gender);
        btnSignUp = findViewById(R.id.btn_insert_data);
        btnViewUsers = findViewById(R.id.btn_view_users);
        rgGender.setOnCheckedChangeListener(new
RadioGroup.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(RadioGroup group, int
checkedId) {
                if(checkedId == R.id.rb_male) {
                    gender = "Male";
                } else {
                    gender = "Female";
                }
            }
        });
        btnSignUp.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(validateForm()) {
                    name = tietName.getText().toString();
                    email = tietEmail.getText().toString();
                    contactNo = tietContactNo.getText().toString();
                    boolean insertResult =
databaseHelper.insertUserDetails(email, name,
                            contactNo,
                            gender);
                    if(insertResult) {
```

```
                        Toast.makeText(context, "Data inserted
succesfully",
                                Toast.LENGTH_SHORT).show();
                    } else {
                        Toast.makeText(context, "Data not inserted",
                                Toast.LENGTH_SHORT).show();
                    }
                    Intent intent = new Intent(context,
MainActivity.class);
                    startActivity(intent);
                }
            }
        });
        btnViewUsers.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(context,
SecondActivity.class);
                startActivity(intent);
            }
        });
    }
    public boolean validateForm() {
        if(tietName.getText().toString().isEmpty()) {
            tilName.setError("Enter Name");
            focusOnView(tilName);
            return false;
        } else if(tietEmail.getText().toString().isEmpty()) {
            tilEmail.setError("Enter Email");
            focusOnView(tilEmail);
            return false;
        } else if(tietContactNo.getText().toString().isEmpty()) {
            tilContactNo.setError("Enter Contact no");
            focusOnView(tilContactNo);
            return false;
        } else if(rgGender.getCheckedRadioButtonId() == -1) {
            Toast.makeText(context, "Select Gender",
Toast.LENGTH_SHORT).show();
            focusOnView(rgGender);
            return false;
        } else {
            return true;
        }
```

```
    }
    public void focusOnView(final View view) {
        svRegistrationForm.post(new Runnable() {
            @Override
            public void run() {
                svRegistrationForm.smoothScrollTo(0, view.getTop()-50);
            }
        });
    }
}
```

**Output:**



Om Shete
THADOMAL SHAHANI TSEC ENGINEERING COLLEGE
TE C32
2103163

# Experiment No : 5

**Aim :** WAA to develop an EMI calculator application.

**Theory :**

- Developing an EMI (Equated Monthly Installment) Calculator application involves creating a user interface to take input values such as principal amount, interest rate, and tenure and then calculating the EMI based on these inputs.
- Below is a step-by-step guide on how to develop an EMI calculator application using Android and Java :

**Step 1 :** Setup the Android Project

=>
1) Open Android Studio and create a new project.
2) Choose an appropriate template (For e.g., Empty Activity)
3) Setup the project with a suitable package name and save location.

**Step 2 :** Design the UI

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

- To develop EMI calculator, we need to create a UI for input and output, handle user input, perform calculations and display the result.

1) Create the layout
⇒ Design the layout using XML in the 'res/layout' directory. This layout should include input fields for principal amount, interest rate, loan tenure and a button to calculate EMI.
- You can use 'EditText' for input fields and 'Button' for the calculation trigger.

2) Create MainActivity
⇒ In 'MainActivity.java' file, reference the UI components and setup click listener for the calculation button.
- When the button is clicked, retrieve input values, perform the EMI calculation and display the result.

3) Implement EMI calculation
⇒ In the 'calculateEMI' method, use the appropriate formula to calculate the EMI based on the principal amount, interest rate and loan tenure.

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

- The monthly Interest can be calculated as -

$$\text{Monthly Interest} = \frac{\text{interest}}{(12 \times 100)}$$

- The emi can be calculated as

$$\text{emi} = \frac{\text{Principal} \times \text{rate} \times \text{divident}}{(\text{divident} - 1)}$$

Conclusion:

The program to create an android-based EMI calculator was implemented successfully.

Jain (A+)

**Code:**

(activity_main.xml)

```
<androidx.coordinatorlayout.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    tools:context="com.example.mcc_exp5.MainActivity"
    android:layout_height="match_parent"
    android:background="@color/black"
    android:backgroundTint="@color/black">
    <androidx.core.widget.NestedScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_behavior="@string/appbar_scrolling_view_behavior">
        <LinearLayout
            android:layout_width="fill_parent"
            android:layout_height="match_parent"
            android:layout_marginTop="?attr/actionBarSize"
            android:orientation="vertical"
            android:paddingLeft="20dp"
            android:paddingRight="20dp"
            android:paddingTop="10dp">
            <com.google.android.material.textfield.TextInputLayout
                android:id="@+id/input_layout_principal"
                android:layout_width="match_parent"
                android:layout_height="wrap_content">
                <EditText
                    android:id="@+id/principal"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:singleLine="true"
                    android:inputType="number"
                    android:digits="0123456789."
                    android:hint="Principal" />
            </com.google.android.material.textfield.TextInputLayout>
            <com.google.android.material.textfield.TextInputLayout
                android:id="@+id/input_layout_interest"
                android:layout_width="match_parent"
                android:layout_height="wrap_content">
                <EditText android:id="@+id/interest"
```

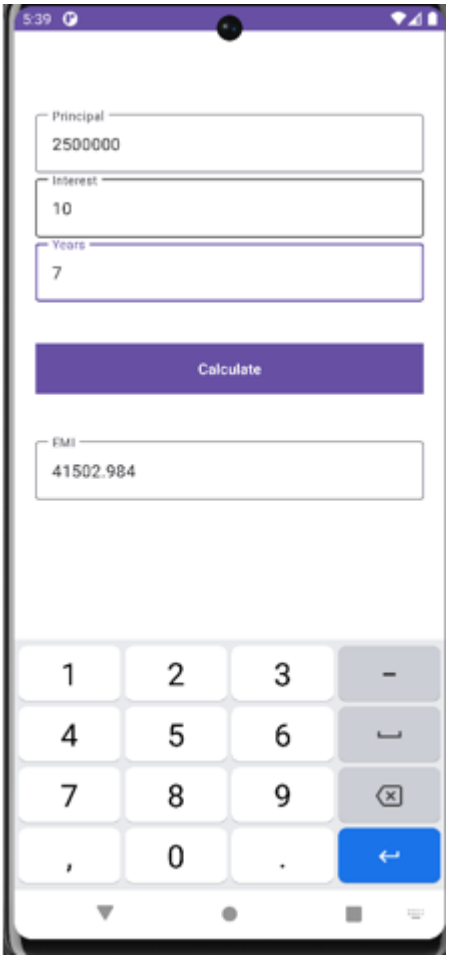
```
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:singleLine="true"
                android:inputType="number"
                android:digits="0123456789."
                android:hint="Interest" />
        </com.google.android.material.textfield.TextInputLayout>
        <com.google.android.material.textfield.TextInputLayout
            android:id="@+id/input_layout_tenure"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">
            <EditText
                android:id="@+id/years"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:inputType="number"
                android:digits="0123456789."
                android:hint="Years" />
        </com.google.android.material.textfield.TextInputLayout>
        <Button android:id="@+id/btn_calculate2"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Calculate"
            android:background="#000000"
            android:layout_marginTop="40dp"
            android:textColor="#FFFFFF"/>
        <com.google.android.material.textfield.TextInputLayout
            android:id="@+id/input_layout_emi"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="40dp">
            <EditText android:id="@+id/emi"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:maxEms="0"
                android:inputType="number"
                android:hint="EMI" />
        </com.google.android.material.textfield.TextInputLayout>
    </LinearLayout>
</androidx.core.widget.NestedScrollView>
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

(MainActivity.java)

```
package com.example.mcc_exp5;
import android.os.Bundle;
import androidx.appcompat.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 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);
                result.setText(String.valueOf(emi));
            }
        });
    }
    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);
    }
}
```

**Output:**


5:39
Principal
2500000
Interest
10
Years
7
Calculate
EMI
41502.984
1
2
3
4
5
6
7
8
9
,
0
.

Om Shete
THADOMAL SHAHANI TSEC ENGINEERING COLLEGE
TE C32
2103163

# Experiment No : 6

**Aim :** To create an app to push notification using Android Studio.

**Theory :**

- **Push Notification**

⇒ Push notifications are a communication channel used by mobile applications to interact with users.

- These notifications are sent by the server to the mobile device and appear in the notification bar, providing users with timely and relavant information.
- Push notification serve as a tool for apps to engage users even when the app is not actively in use.

- **Components of Push Notifications :**

i) **Server**

⇒ The server is responsible for sending push notifications to intended devices.

- It holds the logic to determine when and what notifications should be sent.
- Servers often use cloud services, such as Firebase Cloud Messaging (FCM) or Google Cloud Messaging (GCM), to send push notifications.

2) Client :

⇒ The mobile app needs to register with a push notification service to receive notifications.

- It handles the reception and display of push notifications.
- The app may also include logic to determine the behaviour when a notification is received, such as opening a specific screen or performing a particular action.

3) Push Notification Service :

⇒ This is a platform specific service that facilitates the delivery of notifications from the server to the mobile device.

- Examples include FCM for android and Apple Push Notification Service (APNs) for ios.
- These services ensure that notifications reach the intended devices efficiently.

- Conclusion :

⇒ The program to create an app to push notifications in android was implement successfully.

Pass (A)

**Code:**

(activity_main.xml)

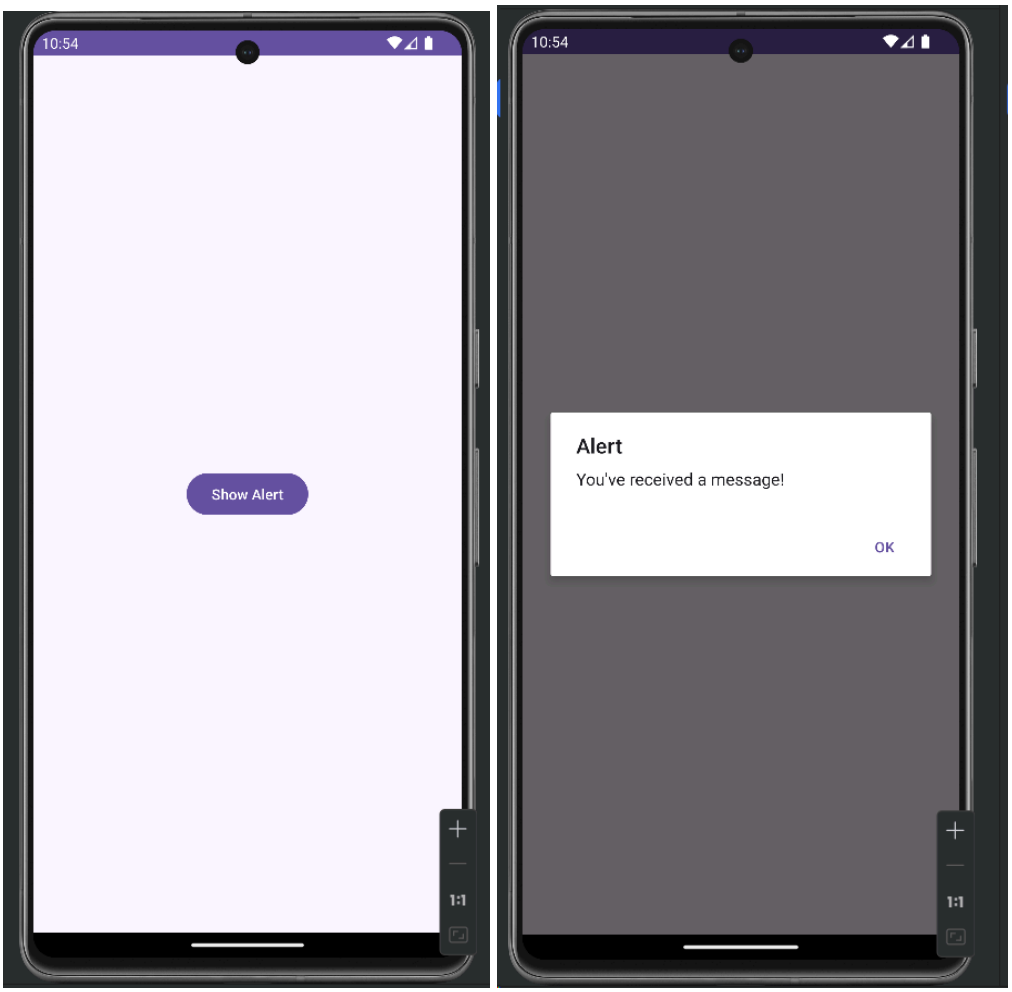
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="16dp"
    android:paddingTop="16dp"
    android:paddingRight="16dp"
    android:paddingBottom="16dp"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/alertButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Show Alert"
        android:layout_centerInParent="true"
        android:onClick="showAlert" />
</RelativeLayout>
```

(MainActivity.java)

```
package com.example.mcc_exp6;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

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

        Button alertButton = findViewById(R.id.alertButton);
        alertButton.setOnClickListener(new View.OnClickListener() {
```

```
            @Override
            public void onClick(View view) {
                showAlert();
            }
        });
    }

    private void showAlert() {
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle("Alert")
                .setMessage("You've received a message!")
                .setPositiveButton("OK", null);

        AlertDialog alertDialog = builder.create();
        alertDialog.show();
    }
}
```

**Output:**



Om Shete
THADOMAL SHAHANI TSEC ENGINEERING COLLEGE
TE COMP 32
2103163

# Experiment No 7

**Aim:** To create a basic calculator using Android studio

**Theory:**

- A calculator is a device or software application that performs arithmetic and mathematical operations.
- In this experiment, the focus is on creating a simple calculator using Android studio.
- The calculator will support basic arithmetic operations such as addition, subtraction, multiplication and division.
- Following functions are used in this experiment:

1) onCreate Method
=) This is a lifecycle method in Android, called when the activity is first created.
- It is where you set up the initial state of your activity, including UI elements and event handlers.

2) attachButtonClickListener(int buttonId) Method
=) A custom method to attach click listeners to buttons.
- It takes a button ID as parameter, finds the button in the layout, and sets a

click listener on it.

3) handleButtonClick ( String buttonText)
⇒ Called only when any digit or operation button is clicked.
- Appends the click button's text to the 'inputStringBuilder', representing the current user input.
- Then, it updates the displays to show the current input.

4) updateDisplay() method
⇒ Updates the 'textViewDisplay' with the current content of the 'inputStringBuilder'.
- Called after each button click to keep the displayed input up to date.

5) handleEqualButtonClick() Method
⇒ Called when the equal button is clicked.
- Tries to evaluate the expression represented by current input string.

- Conclusion
⇒ The program to create a basic calculator in android studio was successfully implemented.

Pain (A+)

**Code:**

(activity_main.xml)

```
<LinearLayout
    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"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/number1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
        android:hint="Enter first number" />

    <EditText
        android:id="@+id/number2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
        android:hint="Enter second number" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <Button
            android:id="@+id/addButton"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Add" />

        <Button
            android:id="@+id/subtractButton"
            android:layout_width="0dp"
```

```
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Subtract" />

        <Button
            android:id="@+id/multiplyButton"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Multiply" />

        <Button
            android:id="@+id/divideButton"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Divide" />
    </LinearLayout>

    <TextView
        android:id="@+id/result"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="24sp"
        android:text="Result will be displayed here" />

</LinearLayout>
```

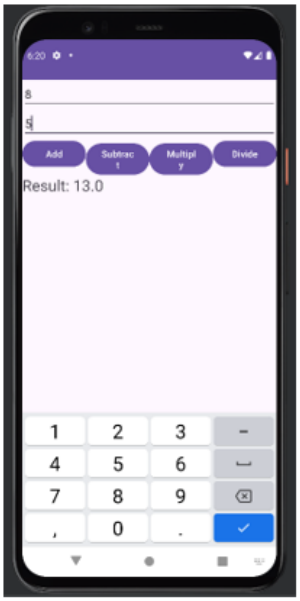
(MainActivity.java)

```
package com.example.mcc_exp5;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    EditText number1, number2;
```

```
    Button addButton, subtractButton, multiplyButton, divideButton;
    TextView result;

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

        number1 = findViewById(R.id.number1);
        number2 = findViewById(R.id.number2);
        addButton = findViewById(R.id.addButton);
        subtractButton = findViewById(R.id.subtractButton);
        multiplyButton = findViewById(R.id.multiplyButton);
        divideButton = findViewById(R.id.divideButton);
        result = findViewById(R.id.result);

        addButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                double num1 =
Double.parseDouble(number1.getText().toString());
                double num2 =
Double.parseDouble(number2.getText().toString());
                double res = num1 + num2;
                result.setText("Result: " + res);
            }
        });

        subtractButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                double num1 =
Double.parseDouble(number1.getText().toString());
                double num2 =
Double.parseDouble(number2.getText().toString());
                double res = num1 - num2;
                result.setText("Result: " + res);
            }
        });

        multiplyButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
```

```
                double num1 =
Double.parseDouble(number1.getText().toString());
                double num2 =
Double.parseDouble(number2.getText().toString());
                double res = num1 * num2;
                result.setText("Result: " + res);
            }
        });

        divideButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                double num1 =
Double.parseDouble(number1.getText().toString());
                double num2 =
Double.parseDouble(number2.getText().toString());
                double res = num1 / num2;
                result.setText("Result: " + res);
            }
        });
    }
}
```

**Output:**



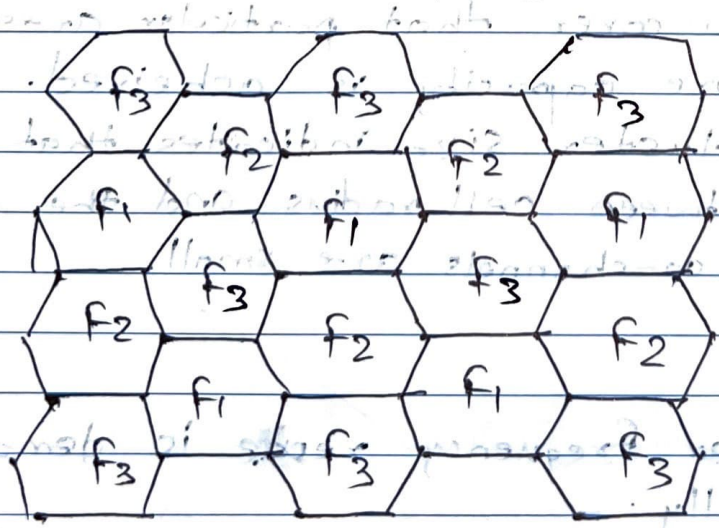
Om Shete
THADOMAL SHAHANI TSEC ENGINEERING COLLEGE
TE C32
2103163

# Experiment No. 8

**Aim:** Write a program to demonstrate Cellular Frequency reuse.

**Theory:**

- Frequency reuse is the technique of using the same radio frequencies on radio transmitter sites within a geographic area that are separated by sufficient distance to cause minimal interference with each other.
- To avoid interference in cellular system, each cell uses a different set of frequencies are compared to its immediate neighbors.
- In other words, no two neighbors use the same set of frequencies as there will be interference.
- A set of several cells are further grouped into clusters. Cells within the same clusters do not use the same frequency sets.




(3 cells cluster)

- Consider a cellular system which has S full duplex channels available for use.
- Assume that the S channels are divided into N number of cells and each cell is allocated a group of K channels. ($K < S$)
- Thus, total number of channels per cell is $K = S/N$
- Therefore, $S = KN$
- The N cells which collectively use the complete set of available frequencies is called cluster.
- The factor N is called the cluster size.
- The frequency reuse factor of cellular system is given by reciprocal of cluster size. i.e. $\frac{1}{N}$.
- If the cluster size N is reduced while the cell size remains constant, more clusters are required to cover that particular area and hence more capacity is achieved.
- A large cluster size indicates that the ratio between cell radius and the distance between co-channels are small.

- Conclusion:
The cellular frequency reuse is demonstrated successfully.

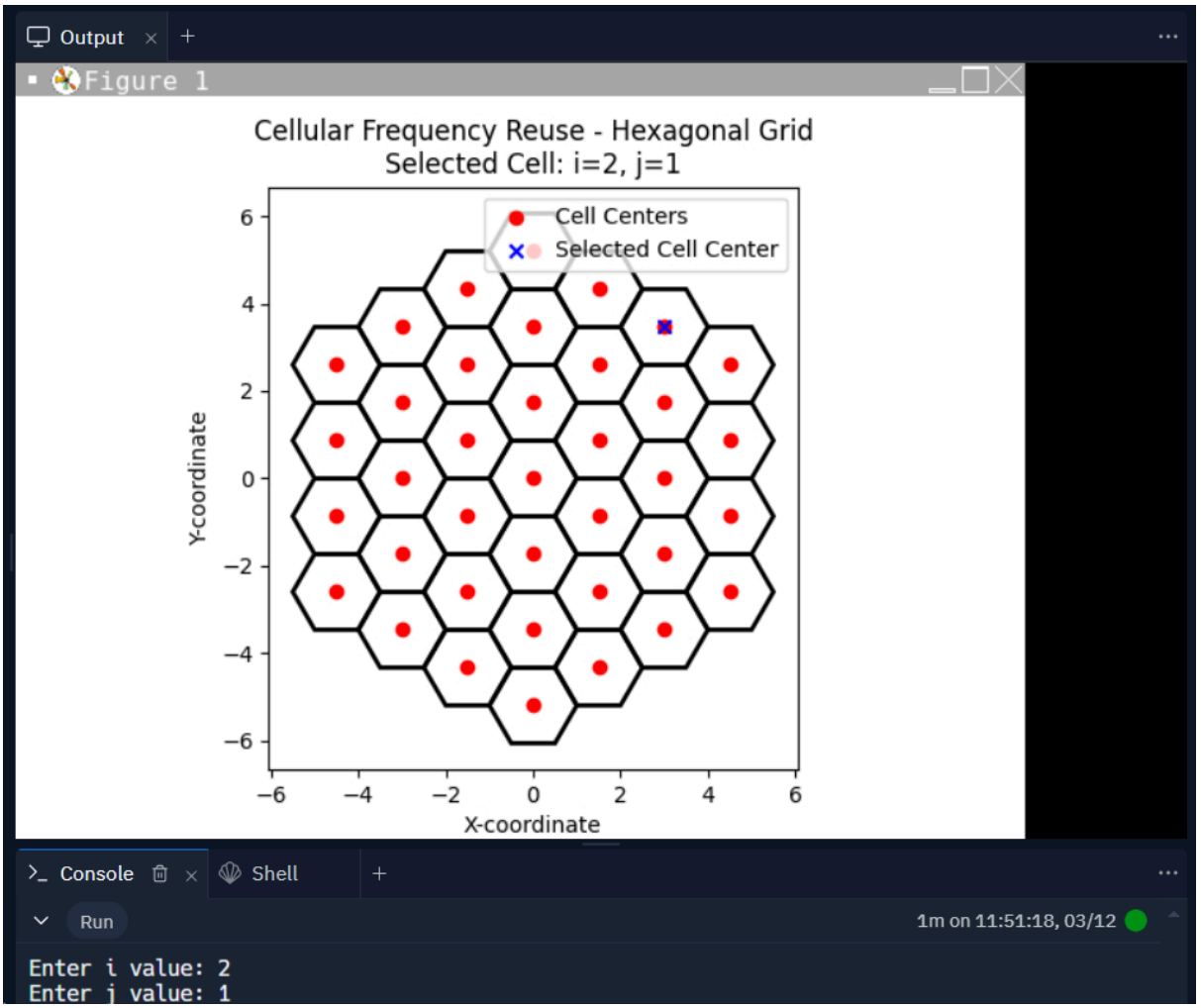
Jain (A+)

**Code:**

```
import matplotlib.pyplot as plt
import numpy as np


def hexagonal_grid(radius, num_cells):
  centers = []
  for i in range(-num_cells, num_cells + 1):
    for j in range(-num_cells, num_cells + 1):
      if abs(i + j) <= num_cells:
        x = 1.5 * radius * i
        y = np.sqrt(3) / 2 * radius * (2 * j + i)
        centers.append((x, y))
  return centers


def plot_cells(centers, radius, selected_i, selected_j):
  fig, ax = plt.subplots()
  ax.set_aspect('equal')

  for center in centers:
    hexagon = plt.Polygon(np.array([[
        np.cos(np.pi / 3 * i) * radius + center[0],
        np.sin(np.pi / 3 * i) * radius + center[1]
    ] for i in range(6)]),
                          edgecolor='black',
                          linewidth=2,
                          fill=None)
    ax.add_patch(hexagon)

  plt.scatter(*zip(*centers), color='red', marker='o', label='Cell
Centers')

  selected_center = (1.5 * radius * selected_i,
                     np.sqrt(3) / 2 * radius * (2 * selected_j +
selected_i))
  plt.scatter(*selected_center,
              color='blue',
              marker='x',
              label='Selected Cell Center')

  plt.title(
```

```
      f'Cellular Frequency Reuse - Hexagonal Grid\nSelected Cell:
i={selected_i}, j={selected_j}'
  )
  plt.xlabel('X-coordinate')
  plt.ylabel('Y-coordinate')
  plt.legend()
  plt.show()


def main():
  radius = 1.0
  num_cells = 3

  selected_i = int(input('Enter i value: '))
  selected_j = int(input('Enter j value: '))

  cell_centers = hexagonal_grid(radius, num_cells)
  plot_cells(cell_centers, radius, selected_i, selected_j)


if __name__ == "__main__":
  main()
```

**Output:**


Output
Figure 1
Cellular Frequency Reuse - Hexagonal Grid
Selected Cell: i=2, j=1
Cell Centers
Selected Cell Center
Y-coordinate
X-coordinate
6
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6
Console
Shell
Run
1m on 11:51:18, 03/12
Enter i value: 2
Enter j value: 1

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE
Om Shete
TE C32
2103163

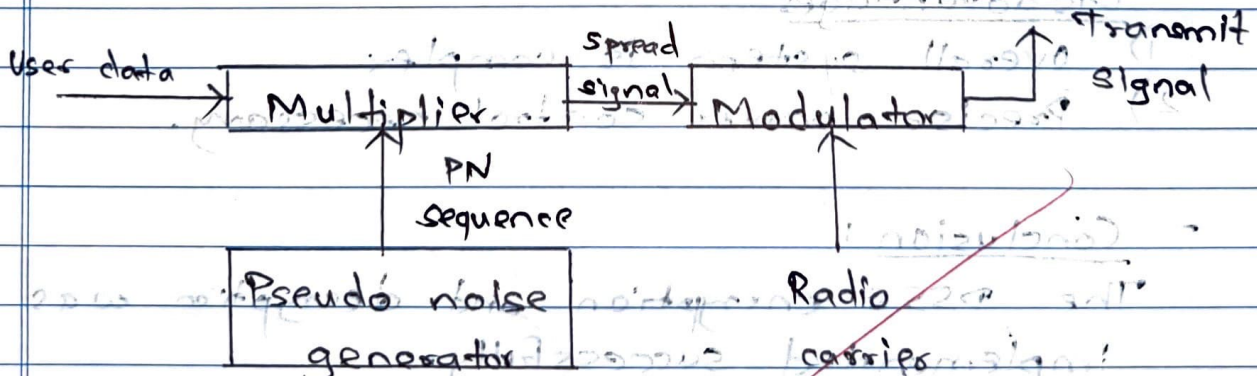
# Experiment No : 9

**Aim :** Write a program to explain the concept of DSSS

**Theory :**

- In telecommunication, direct-sequence spread spectrum (DSSS) is a spread spectrum modulation technique primarily used to reduce overall signal interference.
- The direct-sequence modulation makes the transmitted signal wider in background than the information bandwidth.
- After the despreading or removal of the direct sequencing modulation in the receiver, the info bandwidth is restored, while the unintensional and intensional interference is substantially reduced.
- **DSSS Transmitter**

=>



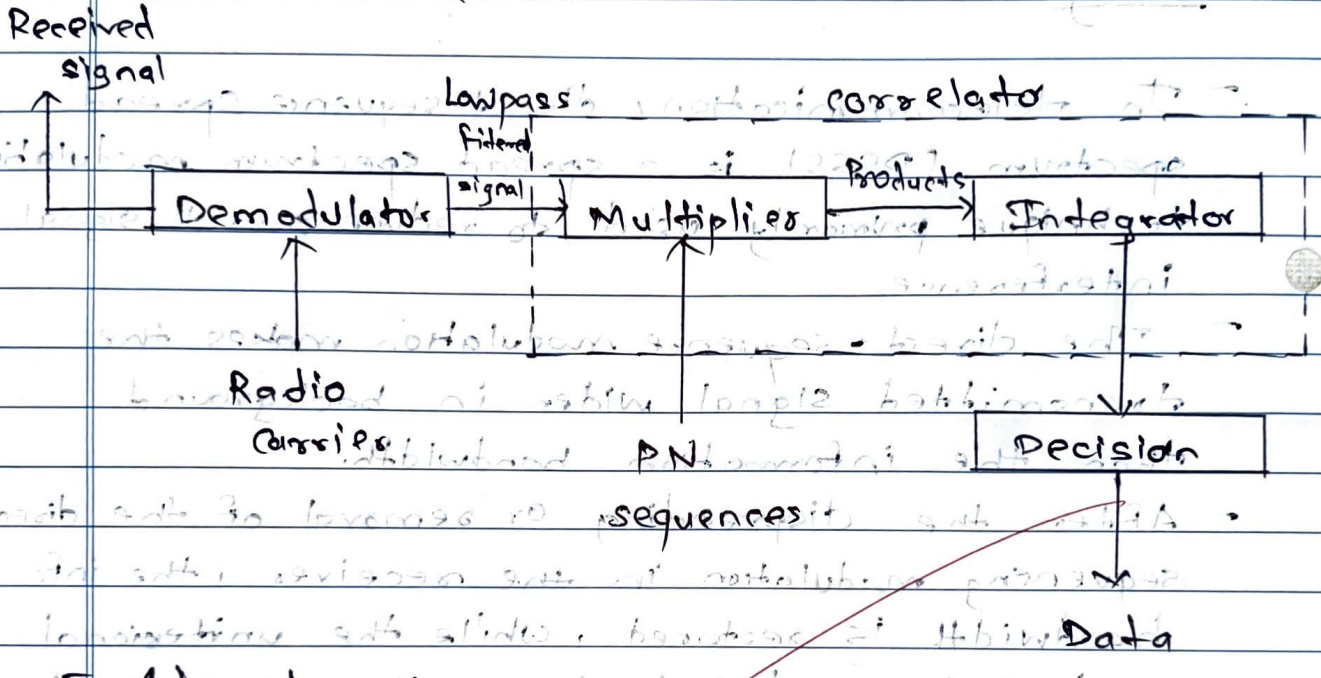

- DSSS transmitter involves two major steps :-
  1) Spreading the signal
  2) Radio modulation

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

- **DSSS Receiver :**

⇒ It also involves three major steps :-

1) Demodulation
2) Correlator
3) Decision making




- **Advantages :**

1) Resistance to Interception
2) Resistance to Fading

- **Disadvantages :**

1) Overall system is complex
2) Precise power control necessary.

- **Conclusion :**

The DSSS encryption and decryption was implemented successfully.

Good (A+)

**Code:**

```
import numpy as np


def generate_spreading_code(length):
  spreading_code = np.random.randint(0, 2, size=length)
  return spreading_code


def string_to_binary(input_string):
  binary_data = ''.join(format(ord(char), '08b') for char in
input_string)
  return np.array(list(map(int, binary_data)))


def binary_to_string(binary_data):
  binary_string = ''.join(map(str, binary_data))
  string_data = ''.join(
      chr(int(binary_string[i:i + 8], 2))
      for i in range(0, len(binary_string), 8))
  return string_data


def dsss_encode(data, spreading_code):
  encoded_data = np.bitwise_xor(data, spreading_code)
  return encoded_data


def dsss_decode(encoded_data, spreading_code):
  decoded_data = np.bitwise_xor(encoded_data, spreading_code)
  return decoded_data


def main():
  user_input = input("Enter a string to transmit: ")

  data = string_to_binary(user_input)
  print("Original Data (Binary):", data)

  spreading_code_length = len(data)

  spreading_code = generate_spreading_code(spreading_code_length)
```

```
  print("Spreading Code:", spreading_code)

  encoded_data = dsss_encode(data, spreading_code)
  print("Encoded Data:", encoded_data)

  decoded_data = dsss_decode(encoded_data, spreading_code)
  print("Decoded Data (Binary):", decoded_data)

  decoded_string = binary_to_string(decoded_data)
  print("Decoded String:", decoded_string)

  if decoded_string == user_input:
    print("Decoding Successful! Original and Decoded data match.")
  else:
    print("Decoding Failed! Original and Decoded data do not match.")


if __name__ == "__main__":
  main()
```

**Output:**

```
Run                                                    4s on 11:40:42, 03/12 ✓

Enter a string to transmit: 110110
Original Data (Binary): [0 0 1 1 0 0 0 1 0 0 1 1 0 0 0 1 0 0 1 1 0 0 0 0 0 0 1 1 0 0 0 1 0 0 1 1 0
 0 0 1 0 0 1 1 0 0 0 0]
Spreading Code: [1 1 0 0 1 1 0 1 1 1 0 0 0 1 1 0 1 0 1 0 0 0 0 1 0 1 1 0 1 0 0 0 0 0 1 0 1
 0 0 0 0 1 0 0 0 1 1 1]
Encoded Data: [1 1 1 1 1 1 0 0 1 1 1 1 0 1 1 1 1 0 0 1 0 0 0 1 0 1 0 1 1 0 0 1 0 0 0 1 1
 0 0 1 0 1 1 1 0 1 1 1]
Decoded Data (Binary): [0 0 1 1 0 0 0 1 0 0 1 1 0 0 0 1 0 0 1 1 0 0 0 0 0 0 1 1 0 0 0 1 0 0 1 1 0
 0 0 1 0 0 1 1 0 0 0 0]
Decoded String: 110110
Decoding Successful! Original and Decoded data match.
```

# Experiment No : 10

**Aim :** Write a program to implement A3/A5/A8 GSM security algorithm.

**Theory :**

- GSM uses three different security algorithms called A3, A5, A8 are generally implemented together.
- An A3/A8 algorithm is implemented in SIM cards and in GSM networks authentication centres.
- It is used to authenticate customers and generate a key for encryption voice and data traffic.
- Development of A3 and A8 algorithms is considered a matter for individual GSM networks operators, although example implementations are available.
- An A5 encryption algorithm scrambles the user's voice and data traffic between the handset and the base station to provide privacy.
- An A5 algorithm is implemented in both the handset and the BSS.

**A3**

⇒

1) Authentication algorithm
2) Calculates SRES Based on the Kikey and RAND send by the MSC

3) Not standardized; can be chosen independently by each operator.

- A8

⇒

1) Key generation algorithm needed to calculate the session key Kc
2) Calculation of Kc depends on ki and RAND.
3) Not standardized; can be choosen independently by each operator

- A5

⇒

1) Stream cipher used to encrypt over the air transmissions.
2) Ciphering is based on Kc and the frame number.
3) Specified at international level to enable roaming.

- Conclusion:

A3/A5/A8 GSM security algorithms are implemented successfully.

Pair (A+)

**Code:**

A3

```
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.Random;

public class Main {
    public static void main(String[] args) throws NoSuchAlgorithmException {
        long k = generateRandomKey();
        long m = generateRandomMessage();

        String sres = a3Algorithm(k, m);

        System.out.println("128-bit Secret Key (K in hexadecimal): " +
Long.toHexString(k));
        System.out.println("128-bit Random Message (M in hexadecimal): " +
Long.toHexString(m));
        System.out.println("RES/SRES (in hexadecimal): " + sres);
    }

    public static long generateRandomKey() {
        Random random = new Random();
        return random.nextLong();
    }

    public static long generateRandomMessage() {
        Random random = new Random();
        return random.nextLong();
    }

    public static String a3Algorithm(long k, long m) throws
NoSuchAlgorithmException {
        byte[] keyBytes = toByteArray(k);
        byte[] messageBytes = toByteArray(m);

        MessageDigest md = MessageDigest.getInstance("MD5");
        byte[] hash = md.digest(concatenateArrays(keyBytes, messageBytes));

        StringBuilder sres = new StringBuilder();
        for (byte b : hash) {
            sres.append(String.format("%02X", b));
        }
        return sres.toString();
    }

    public static byte[] toByteArray(long value) {
        byte[] result = new byte[16];
```

```
        for (int i = 0; i < 8; i++) {
            result[i] = (byte) (value >> (56 - i * 8));
        }
        return result;
    }

    public static byte[] concatenateArrays(byte[] a, byte[] b) {
        byte[] result = new byte[a.length + b.length];
        System.arraycopy(a, 0, result, 0, a.length);
        System.arraycopy(b, 0, result, a.length, b.length);
        return result;
    }
}
```

**Output:**

```
Run                                                    10s on 11:02:58, 03/19 ✓

128-bit Secret Key (K in hexadecimal): 75b0022a0ee84e0d
128-bit Random Message (M in hexadecimal): dc86fd9a7438fcba
RES/SRES (in hexadecimal): 098D641E383AA2B82780E37DD50A731D
```

**Code:**

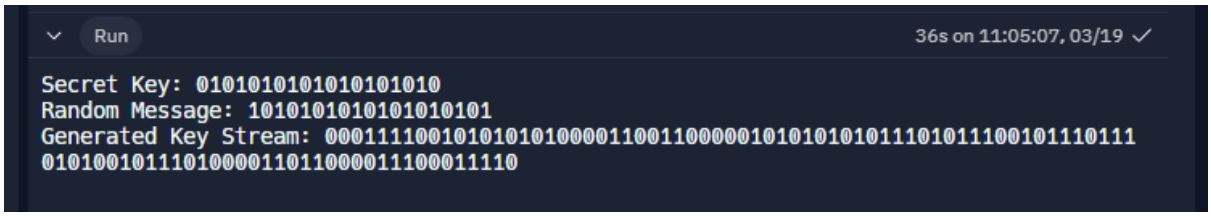
A5

```
public class Main {

  private int[] register1 = new int[19];
  private int[] register2 = new int[22];
  private int[] register3 = new int[23];

  public Main() {
    // Initialize registers with arbitrary values
    for (int i = 0; i < 19; i++) {
      register1[i] = 0;
    }
    for (int i = 0; i < 22; i++) {
      register2[i] = 0;
    }
    for (int i = 0; i < 23; i++) {
      register3[i] = 0;
    }
  }

  public void setKey(String key) {
    // Set the key for register 1
    for (int i = 0; i < 19; i++) {
```

```
      register1[i] = Character.getNumericValue(key.charAt(i % key.length()));
    }

    // Set the key for register 2
    for (int i = 0; i < 22; i++) {
      register2[i] = Character.getNumericValue(key.charAt((i + 19) %
key.length()));
    }

    // Set the key for register 3
    for (int i = 0; i < 23; i++) {
      register3[i] = Character.getNumericValue(key.charAt((i + 41) %
key.length()));
    }
  }

  public void generateKeyStream(int numBits) {
    for (int i = 0; i < numBits; i++) {
      int majority = (register1[8] & register2[10]) ^ (register1[8] &
register3[10]) ^ (register2[10] & register3[10]);
      int newBit = majority ^ register1[18] ^ register2[21] ^ register3[22];

      shiftRegister(register1);
      shiftRegister(register2);
      shiftRegister(register3);

      System.out.print(newBit);
    }
    System.out.println();
  }

  private void shiftRegister(int[] register) {
    int feedback = (register[13] ^ register[16] ^ register[17] ^ register[18])
& 0x01;
    for (int i = register.length - 1; i > 0; i--) {
      register[i] = register[i - 1];
    }
    register[0] = feedback;
  }

  public static void main(String[] args) {
    Main a5 = new Main();
    String key = "01010101010101010"; // Example key
    String randomMessage = "10101010101010101"; // Example random message
    a5.setKey(key);
    System.out.println("Secret Key: " + key);
    System.out.println("Random Message: " + randomMessage);
    System.out.print("Generated Key Stream: ");
```

```
    a5.generateKeyStream(100); // Generate 100 key bits
  }
}
```

**Output:**




**Code:**

A8

```
public class Main {

    // Constants
    private static final int LFSR_LENGTH = 22;
    private static final int[] INIT_STATE = {1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0,
1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0};

    // Variables
    private int[] lfsr = new int[LFSR_LENGTH];

    // Constructor
    public Main() {
        // Initialize LFSR with initial state
        System.arraycopy(INIT_STATE, 0, lfsr, 0, LFSR_LENGTH);
    }

    // Clocking function
    private void clock() {
        int feedback = lfsr[0] ^ lfsr[1] ^ lfsr[2] ^ lfsr[8];
        for (int i = LFSR_LENGTH - 1; i > 0; i--) {
            lfsr[i] = lfsr[i - 1];
        }
        lfsr[0] = feedback;
    }

    // Generate key stream
    public int[] generateKeystream(int numBits) {
        int[] keystream = new int[numBits];
        for (int i = 0; i < numBits; i++) {
            keystream[i] = lfsr[LFSR_LENGTH - 1];
```

```
            clock();
        }
        return keystream;
    }

    public void printInitialState() {
        System.out.println("Initial State of LFSR:");
        for (int bit : lfsr) {
            System.out.print(bit);
        }
        System.out.println();
    }

    public static void main(String[] args) {
        Main a8 = new Main();
        a8.printInitialState();

        int[] keystream = a8.generateKeystream(100); // Generate 100 key bits
        System.out.println("Generated Key Stream:");
        for (int bit : keystream) {
            System.out.print(bit);
        }
    }
}
```

**Output:**

```
Initial State of LFSR:
10100100100100100010
Generated Key Stream:
0100100100100100100101100000111001100010100000011111111101110111000101110011111011000
1001001101011000
```

Om Shete
THADOMAL SHAHANI TSEC ENGINEERING COLLEGE
TE C32
2103163

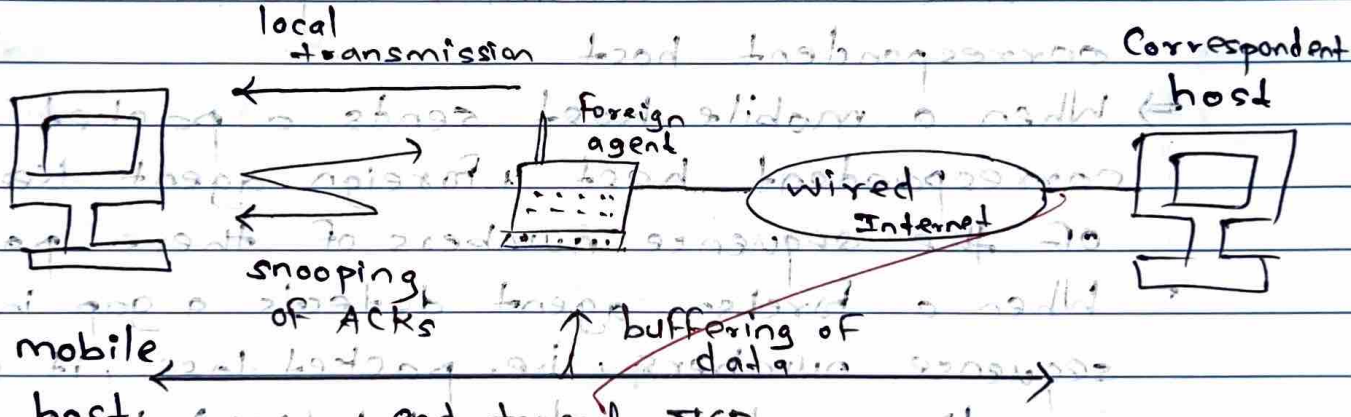
# Assignment No : 1

Q.1 Explain in detail with merits and demerits.

a) Snooping TCP (S-TCP)

⇒

- Snooping TCP works completely transparently and leaves the TCP end-to-end connection intact.
- It overcomes some drawback of Indirect TCP (I-TCP)




- Snooping TCP works as follows :

1) Correspondent host sends a packet to mobile host

⇒ Correspondent host sends a pack to mobile host via a wired TCP connection. The access point buffers the packet sent by correspondent host.

- Access point also snoops on the packet in both directions to recognize acknowledgements

- Once the mobile host receives the packet, it sends an acknowledgment and this ACK also passes through the access point.
- IF the access point doesn't receive any ACK from a mobile host within certain amount of time, then it retransmits the packet from its buffer, performing a much faster retransmission compared to the fixed host

2) Mobile host transmits a packet to a correspondent host

⇒ When a mobile host sends a packet to correspondent host, foreign agent keeps track of the sequence numbers of these packet.
- When a foreign agent detects a gap in the sequence numbers, i.e. packet loss, it sends a negative acknowledgment (NACK) to the mobile host.
- Once the mobile host receives the NACK, it can retransmit the missing packet immediately.

- Merits of Snooping TCP

⇒

1) The end-to-end semantics are preserved
2) Correspondent host need not be changed, most of the enhancements are done in foreign agent.

- Demerits of Snooping TCP

⇒

1) Using NACK between foreign agent and the mobile host assumes additional mechanisms on mobile host. This approach is no longer transparent for arbitrary mobile hosts.
2) If user applies end-to-end encryption, I-TCP fails.

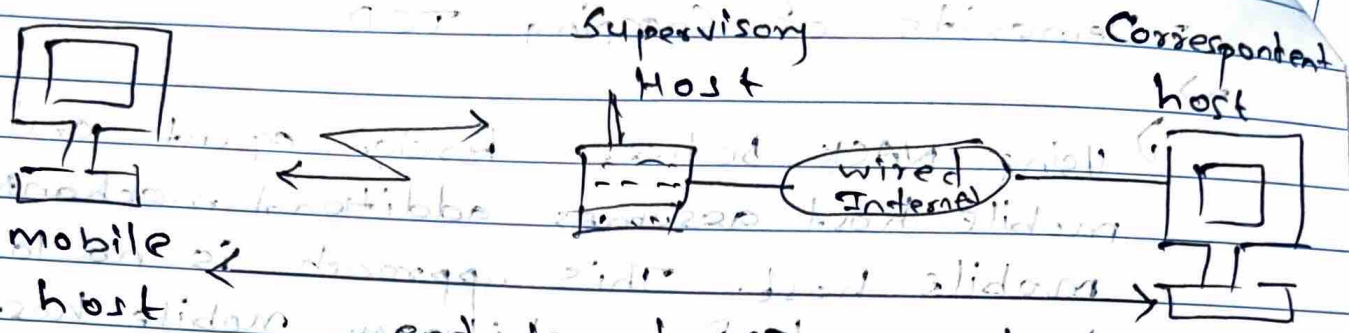
2) Mobile TCP (M-TCP)

⇒ The occurence of lengthy and/or frequent disconnection is the major problem in wireless networks.

- M-TCP aims
  1) To improve overall throughput
  2) To lower the delay
  3) To maintain end-to-end semantics of TCP
  4) To provide a more efficient handover

- Working of M-TCP

⇒ Packets are sent to the mobile host by a correspondent host.
- If any packet is lost on the wireless side, then the original sender retransmits the packet.
- Thus, end-to-end semantics are maintained
- All the packets sent to MH are monitored by the SH and are acknowledged by the MH via ACK packets




- After a set amount of time, if the SH still does not receive any ACK, it assumes that the MH is disconnected.
- SH sets sender's window size to zero and thus chokes the sender.
- Once the window size is set to zero, the sender is forced to go into a persistent mode.
- In the persistent mode, independent of the receiver's period of disconnected state, the state of the sender will not change.
- Once the SH detects the connectivity again, the sender's window size is again set to the old value, enabling the sender to send at full speed.

Merits of M-TCP:

⇒

1) End-to-end semantics are maintained. SH itself doesn't send any ACK, it only forwards ACKs that were received from the MH.
2) It avoids unnecessary retransmissions, if the MH is disconnected.

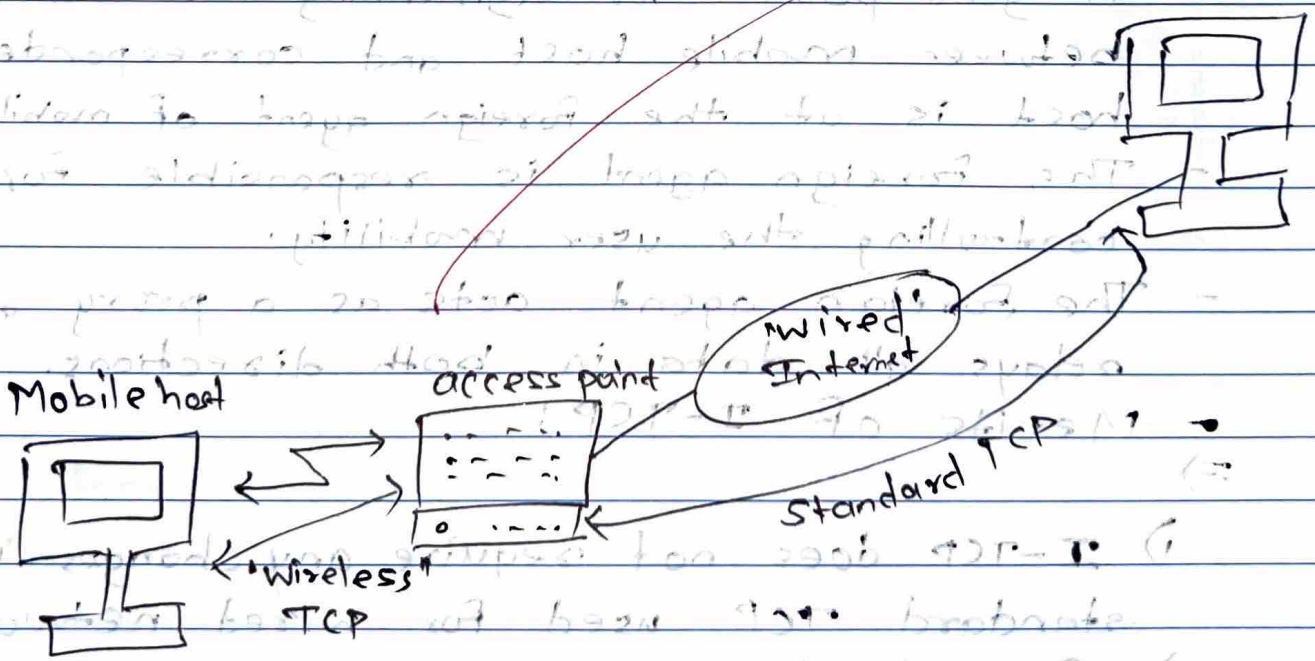
- Demerits of M-TCP

⇒

1) Losses on wireless link are propagated to the wired link. this is because SH does not act as a proxy and does not buffer the packets and is not responsible for local retransmission.
2) It requires new network element like bandwidth manager.

3) Indirect TCP (I-TCP)

⇒ There are two facts : one is that TCP performs poorly together with wireless link and second is that TCP within the fixed network cannot be changed.



- Above figure shows an example with a mobile host connected via a wireless link and an access point to the wired internet where the correspondent node resides. The correspondent node could also use wireless access.
- I-TCP separates a TCP connection into two parts:
  1) Fixed part is between the mobile support router and the fixed host over the fixed network.
  2) Wireless part is between the MH and its access point over the wireless medium.
- Standard TCP is used between the fixed computer and access point.
- A good point for segmenting the connection between mobile host and correspondent host is at the foreign agent of mobile IP.
- The foreign agent is responsible for controlling the user mobility.
- The foreign agent acts as a proxy and relays all data in both directions.
- Merits of I-TCP:

1) I-TCP does not require any changes in the standard TCP used for wired networks.
2) Due to the partioning transmission errors on the wireless link cannot propagate into the fixed networks

Demerits of I-TCP

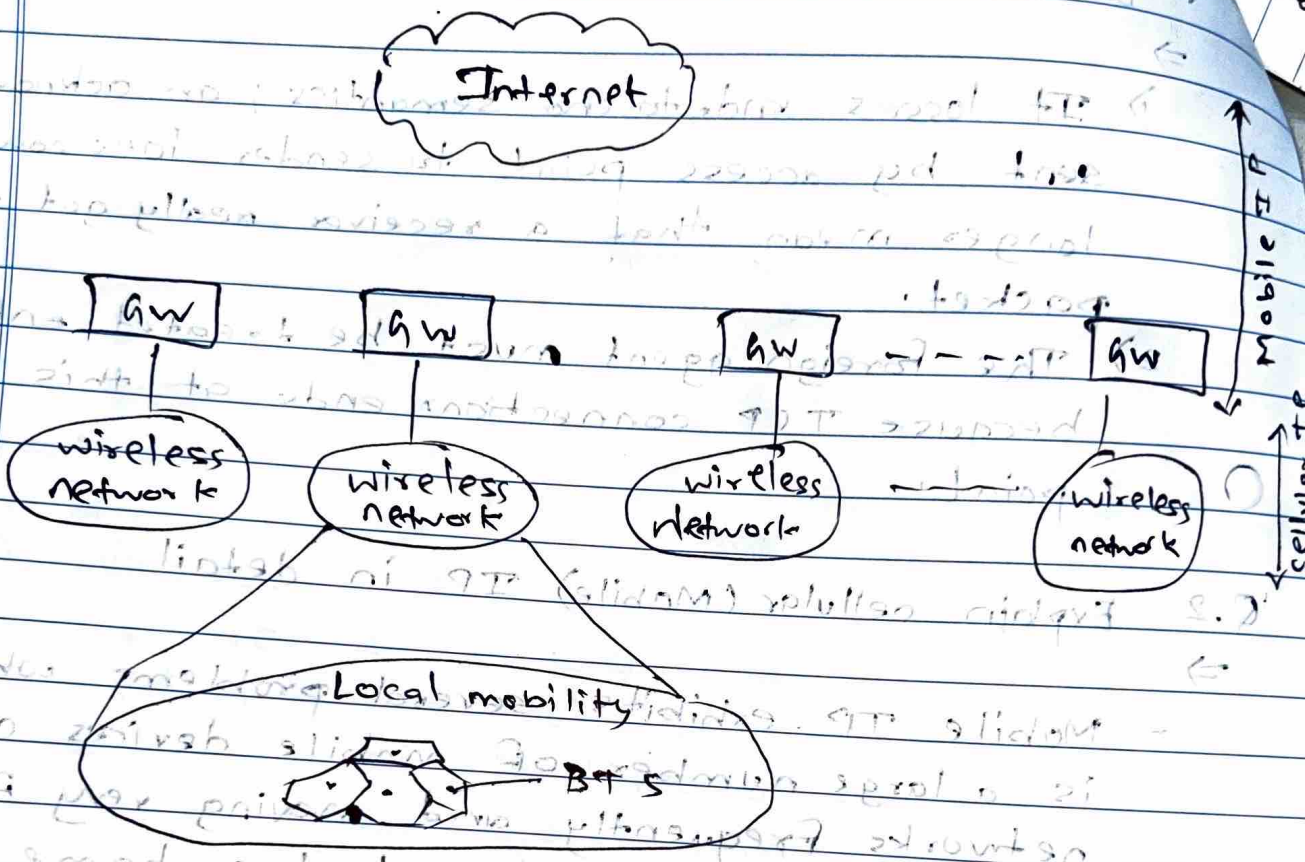
⇒

1) It losses end-to-end semantics; an acknowledgment sent by access point to sender does now no longer mean that a receiver really got a packet.
2) The foreign agent must be treated entity because TCP connections ends at this point.

2 Explain cellular (Mobile) IP in detail

⇒

- Mobile IP exhibits several problems when there is a large number of mobile devices changing networks frequently and moving very fast.
- In such cases, a high load on home agents and on the network is generated by registration and binding update message.
- Mobile IP basically designed for macro level mobility and relatively slow moving hosts.
- Cellular IP (CIP) is a new robust, simple and flexible protocol for highly mobile hosts.
- CIP complements Mobile IP by supporting local mobility.
- It can accomodate large number of users by separating idle hosts from active hosts.




- CIP architecture :

⇒ The architecture of cellular IP is shown in above figure.

- It consists of three major components :-

1) cellular IP gateway (GW)
2) Cellular IP node or base station (BS)
3) Cellular IP mobile host (MH)

Q.3 Explain MIPv6

⇒

- The first IP mobility protocol, Mobile IP was developed for IPv4.
- The mobile IP protocol solves the TCP/IP Layer 3 mobility problem, by assigning a permanent IP address to the mobile node.
- Mobile IP supports for both MIPv4 and MIPv6 but IPv4 has a couple of drawbacks
- The main drawback of IPv4 is address exhaustion, making MIPv4 the future option for mobility protocol in IP networks.
- Mobile IPv6 (MIPv6) is protocol developed as a subset of IPv6 support mobility.
- MIPv6 is an update of the Mobile IP standard designed to authenticate mobile device using IPv6 addresses
- In traditional IP routing, IP addresses represent a topology. Routing mechanisms rely on the assumption that each network node will always have the same point of attachment to the Internet, and that each node's IP address identifies the network link where it is connected.
- In this routing scheme, if you disconnect a mobile device from the internet and want to reconnect through different network, you have to configure the device with a new IP address, and the appropriate netmask

and default router.

- Otherwise, routing protocols have no means of delivering packets because the device's network address doesn't contain the necessary information about the node's network point of attachment to the Internet
- Mobile IPv6 allows a mobile node to transparently maintain connection while moving from one subset to another.

Q.4 Write short note on: HAWAII

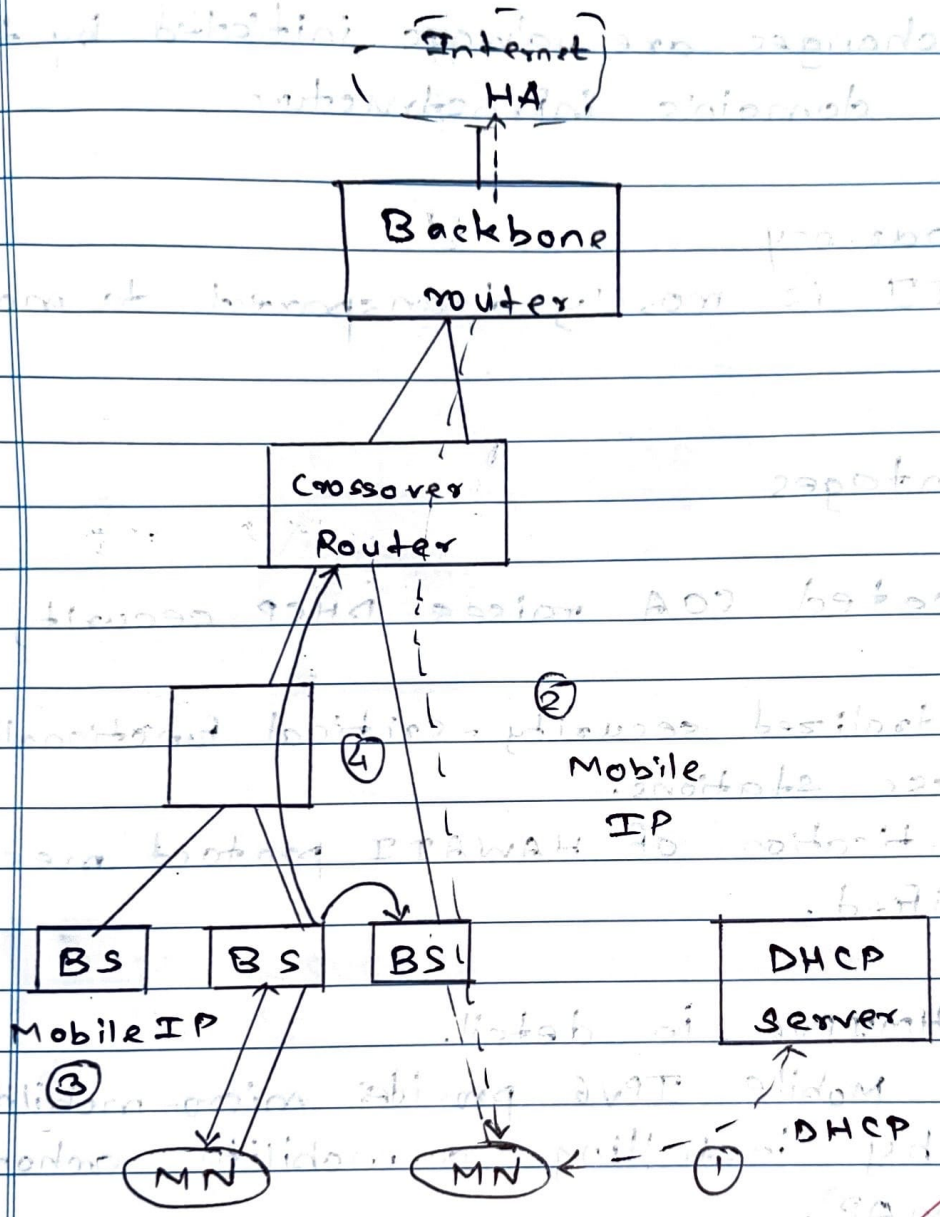
⇒

- HAWAII stands for Handoff-Aware Wireless Access Internet Infrastructure tries to keep micro mobility support as transparent as possible for both home agent and MN.
- Working:

Step 1: On entering an HAWAII domain, a mobile node obtains a co-located COA.

Step 2: MN registers with the HA.

Step 3: When MN moving another cell inside the foreign domain, the MN sends a registration request to the new base station as to a foreign agent.




Step 4: The base station interprets the registration request and sends out a handoff update message, which reconfigure all routers on the paths from the old and new base station to the crossover router.

Advantages:

1) Security: Challenges response extensions are mandatory. In contrast to cellular IP,

routing changes are always initiated by the foreign domain's infrastructure.

2) Transparency

⇒ HAWAII is mostly transparent to mobile nodes.

- Disadvantages

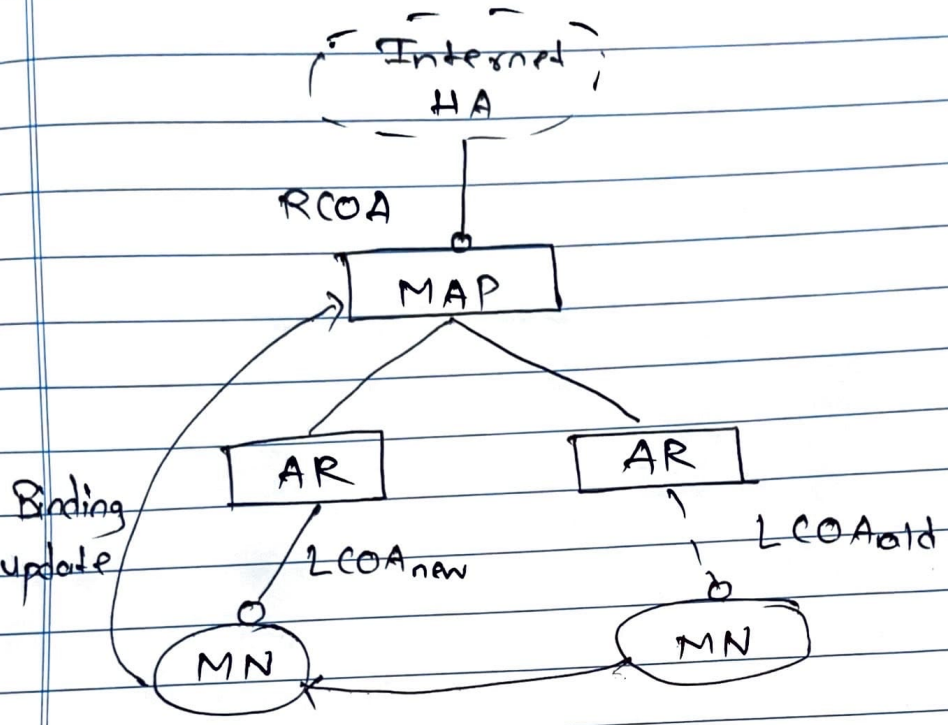
⇒

1) Co-located COA raises DHCP security issue.
2) Decentralized security-critical functionality in base stations.
3) Authentication of HAWAII protocol message unspecified.

Q.] Explain HMIPV6 in detail.

⇒ Hierarchical Mobile IPv6 provides micro-mobility support by installing a mobility anchor point (MAP).

- MAP is an entity which is responsible for a certain domain and acts as a local HA within this domain for visiting MN's
- The following figure shows the basic architecture of HMIPv6
- The MAP receives all packets on behalf of the MN, encapsulates and forwards them directly to the MN's current address LCOA




- Advantages :

1) MNs can have location privacy because LCOAs can be hidden.
2) Direct routing between CN's sharing the same link is possible.

- Disadvantages :

1) Additional infrastructure component (MAP)
2) Routing tables are changed based on messages sent by mobile nodes.

Pain - A+

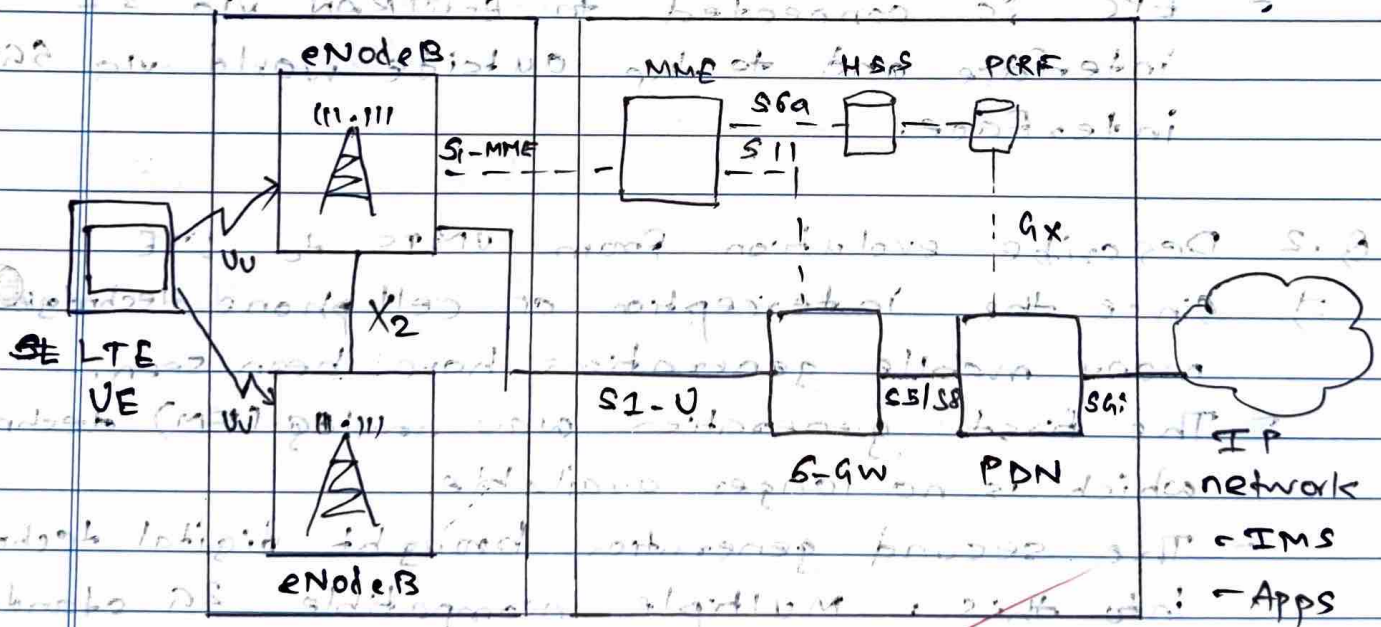
THADOMAL SHAHANI TSEC ENGINEERING COLLEGE
Om Shete
TE C32
2103163

# Assignment 2

Q.1 Draw and explain SAE architecture

⇒ System Architecture Evolution (SAE) is a new network architecture designed to simplify LTE networks.

- It establish a flat architecture similar to other IP based communication protocol.
- SAE uses an eNB and Access Gateway (aGW) and removes the RNC and SGSN from equivalent 3G network architecture. This allows the network to be build with an 'All-IP' based network architecture.
- SAE also includes entities to allow full inter-working with other related wireless technology




- These entities can specifically manage and permit the non-3GPP technologies to interface directly with the network.

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

- The high-level network architecture of LTE is comprised of following three main components:
  1) The User Environment (UE)
  2) The Evolved UMTS Terrestrial Radio Access Network (E-UTRAN)
  3) The Evolved Packet Core (EPC)
- The evolved packet core provides the means to communicate with packet data networks in the outside world such as internet, private corporate networks or the IP multimedia subsystem.
- Between UE and E-UTRAN there is Uu interface.
- EPC is connected to E-UTRAN via S1 interface and to the outside world via SGi interface.

Q.2 Describe evolution from UMTS to LTE

⇒ Since the interception of cell phone technologies many mobile generations have been seen.

- The first generation was analog (FM) technology which is no longer available.
- The second generation brought digital technology into this. Multiple incompatible 2G standards were developed. Only two of them, GSM and IS-95A, CDMA have survived.
- Next, the third generation (3G) standards came into market.

- Again, multiple standards were developed, mainly WCDMA by the 3GPP and CDMA 2000 by Qualcomm. Both have survived and still used today.
- The 3G standards were continually updated into what is known as 3.5G
- WCDMA was updated to HSPA, and CDMA 2000 was expanded with 1xRTT EV-DO release A and B. Both are still widely deployed
- The third generation partnership Project developed widely used UMTS WCDMA/HSPA 3G standards.
- As a 4G successor to WCDMA, 3GPP developed Long-Term Evolution (LTE).
- Thus, LTE was created an upgrade to the 3G standards.
- Release 8 of LTE was complete in 2008, followed by Release 9. Now, release 10 is also available which defines.
- LTE-advanced is also under development.

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

Q.3 Compare mobile generations:

⇒

| Technology Features | 1G | 2G | 3G | 4G | 5G |
|---|---|---|---|---|---|
| Evolution | 1970 | 1980 | 1990 | 2000 | 2020 |
| Deployment | 1984 | 1999 | 2002 | 2010 | 2015 |
| Data rate | 2 kbps | 14.4 - 64 kbps | 2 Mbps | 200 Mbps to 1 Gbps | 10 Gbps to 100 Gbps |
| Famous standards | AMPS | 2G : GSM, CDMA | WCDMA CDMA-2000 | LTA, WiMAX | Not yet defined. |
| Technology behind | Analoge cellular technology | Digital cellular technology | Broad bandwidth CDMA, IP technology | Unified IP and seamless combination of broadband | Unified IP and seamless comb. of broadband. |
| Service | voice | Digital voice SMS | Integrated high quality audio, video and data | Dynamic info access, wearable devices | Dynamic info access, wearable device with AI compatabilities |

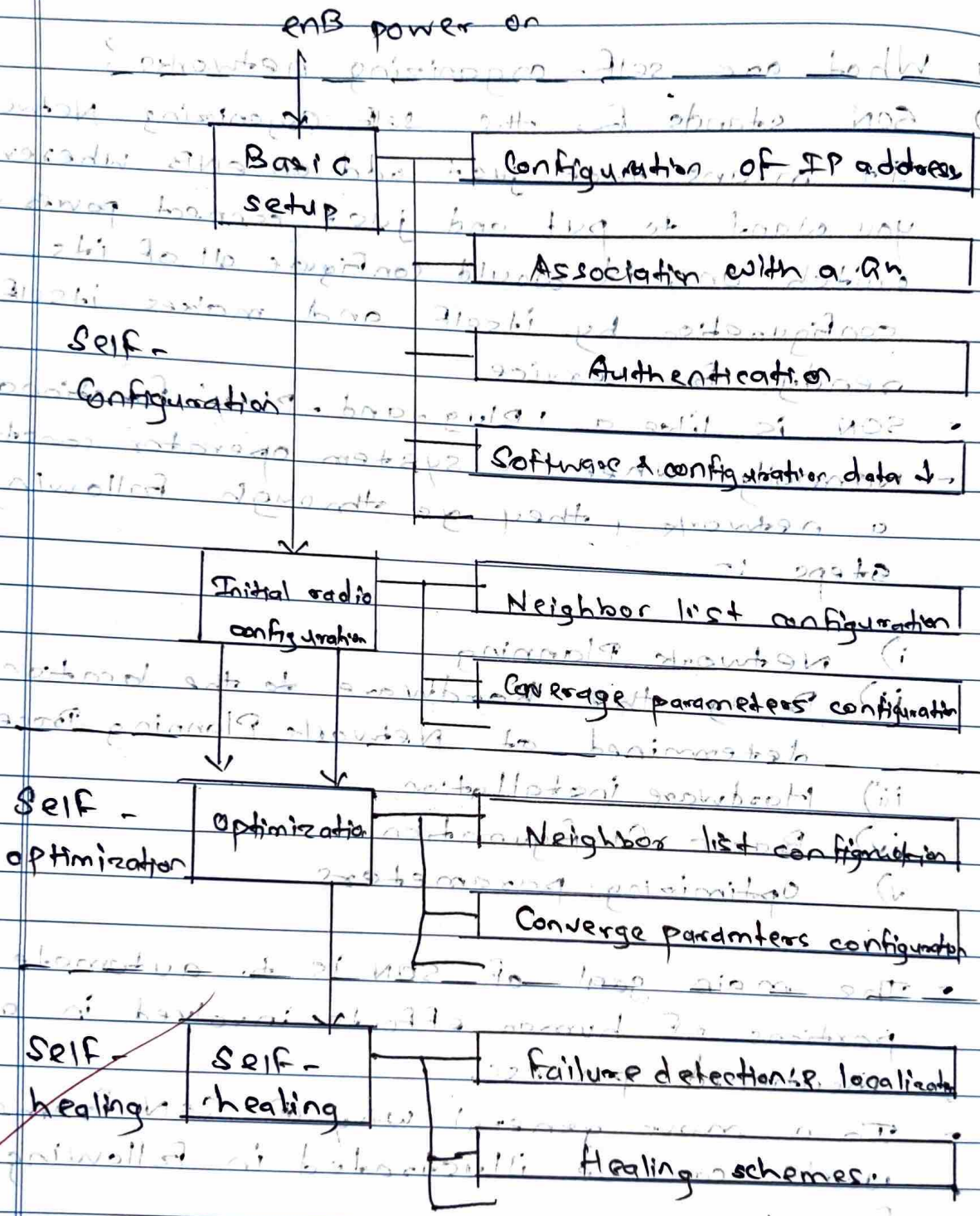
Q.4 What are self-organizing networks?

⇒ SON stands for the self Organizing Network.

- It means that just add an eNB wherever you want to put and just connect power and switch on, it would configure all of its configuration by itself and makes itself ready for service.
- SON is like a 'Plug-and-Play' functionality.
- Normally when a system operator constructs a network, they go through following steps :-

i) Network Planning
ii) Bringing the hardware to the location determined at Network Planning Process
iii) Hardware installation
iv) Basic Configuration
v) Optimizing parameters.

- The main goal of SON is to automate large portions of human efforts involved in above mentioned process.
- In a more general way of SON frame work can be illustrated in following figure :-



Q5 Explain VoLTE in detail

⇒ VoLTE stands for Voice over Long Term Evolution.

- It is a digital packet voice service that is delivered over IP via an LTE access network.
- When 3GPP started designing the LTE system, prime focus was to create a system which can achieve high data throughput with low latency.
- LTE is an all IP network and the ability to carry voice was not given much importance.
- Therefore, for LTE networks to carry traditional circuit switched voice calls, a different solution was required.
- This solution to carry voice over IP in LTE networks is commonly known as "VoLTE".
- Basically VoLTE systems convert voice into data stream, which is then transmitted using the data connection.
- VoLTE is based on the IMS
- IMS is an architectural framework for delivering multimedia communications services such as voice, video and text messaging over IP networks.
- Benefits of VoLTE :-

The implementation of VoLTE offers many benefits, both in terms of cost and operation.

- VoLTE provide following benefits :-

1) Provides a more efficient use of spectrum than traditional voice;
2) Meets the rising demand for richer, more reliable services;
3) Eliminates the need to have voice on one network and data on another;
4) Can be developed in parallel with video calls over LTE and multimedia services, including video share, multimedia messaging.
5) Ensures that video services are fully interoperable across the operator community, just as voice services are.
6) Increase handset battery life by 40%
7) Provide rapid call establishment time.