

Thadomal Shahani Engineering College

Bandra (W.), Mumbai- 400 050.

© CERTIFICATE ©

Certify that Mr./Miss Parth Sandeep Dabholkar
of Computer Department, Semester VI with
Roll No. 2103032 has completed a course of the necessary
experiments in the subject Mobile Computing under my
supervision in the **Thadomal Shahani Engineering College**
Laboratory in the year 2023 - 2024

Teacher In-Charge

Head of the Department

Date 21/03/24

Principal

CONTENTS

SR. NO.	EXPERIMENTS	PAGE NO.	DATE	TEACHERS SIGN.
1.	Write an Android Application to draw Basic graphical 2D primitives.	1 - 6	17-01-24	A
2.	Write an Android Application to draw Basic 3D graphical primitives.	7 - 12	24-01-24	C
3.	WAA - to design a form with GUI components.	13 - 19	07-02-24	C
4.	WAA - to design GUI components with database.	20 - 27	07-02-24	B
5.	WAA to develop an EMI calculator Application.	28 - 36	21-02-24	B
6.	WAA that creates an alert on receiving a message.	37 - 42	21-02-24	A
7.	WAA - to implement Basic Calculator using Android.	43 - 51	28-02-24	A
8.	Write a program to demonstrate Cellular frequency reuse.	52 - 66	28-02-24	A ⁺
9.	Write a program to explain concept of DSSS.	67 - 70	13-03-24	A ⁺
10.	Write a program to implement A3/A5/A8 GSM Security Algorithm.	71 - 83	13-03-24	A ⁺
11.	Assignment no: 1	84 - 90	21-02-24	A ⁺
12.	Assignment no: 2	91 - 104	15-03-24	A ⁺

(A)
EXPERIMENT NO: 1

AIM: WAA to draw basic 2D graphical primitives.

THEORY: Android studio is the official integrated development environment for android application development. Android studio provides more features that enhance our productivity while building android apps.

Features:

- It has flexible gradle-based build system.
- It has a fast and feature-rich emulator for app testing.
- Android studio has a consolidated environment where we can develop for all android app.
- Apply changes to resource code of our running app without restarting the app.
- It supports C++ and NDK.
- Android studio provides extensive testing tools and framework.

Steps on how to use Android studio:

Step 1: Download Android studio from the official download page.

Step 2: Install the required components. Install SDK, AVD manager and android emulator. SDK consists of libraries and tools for app development. AVD manager creates and manages virtual devices for testing.

Step 3: Configure android studio. Launch Android studio and choose standard configuration.

Step 4: Create a new project. Specify the path where your project must be saved. Select the mobile version for its accessibility.

Step 5: Build your application. write and use tools to design or, write code in Java or KOTLIN and access device features.

Step 6: Test your application. Android studio provides an emulator for testing on virtual devices. Use the run button to launch and test application on emulator.

- Built-in function used in the code:

- a) `Bitmap.createBitmap(720, 1280, Bitmap.Config.ARGB_8888);`
 - It creates a new bitmap with a specified width, height and pixel format (ARGB - 8888).
- b) `Canvas(canvas);`
 - Creates a canvas object associated with given bitmap allowing drawing operations.
- c) `Paint.setColor(Color.parseColor("#000000"));`
 - Set the color of paint object to Hex code given in parseColor.
- d) `Paint.setTextSize(20);`
 - Sets the text size of the paint object to 20.
- e) `canvas.drawText("Rectangle", 420, 180, paint);`
 - Draw the text "Rectangle" on the canvas using paint object.
- f) `canvas.drawRect(420, 200, 600, 700, paint);`
 - Draw Rectangle

Experiment 1

Aim: Write an android application (WAA) to draw basic graphical 2D primitives.

Program:

MainActivity.java

```
package com.example.Exp1;

import androidx.appcompat.app.AppCompatActivity;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.os.Bundle;
import android.widget.ImageView;

public class MainActivity extends AppCompatActivity{
    Bitmap bg;
    ImageView img;

    @Override
    protected void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        bg=Bitmap.createBitmap(750,1180, Bitmap.Config.RGB_565); //create bitmap
```

```
img = findViewById(R.id.imageView);
img.setImageBitmap(bg);
Canvas canvas = new Canvas(bg); //canvas creation

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

Paint paint2 = new Paint();
paint2.setColor(Color.WHITE);
paint2.setTextSize(50);

Paint paint = new Paint();
paint.setColor(Color.BLUE);
paint.setTextSize(50);

Paint paint3 = new Paint();
paint3.setColor(Color.CYAN);
paint3.setTextSize(50);

canvas.drawText("Rectangle",420,150,paint1);
canvas.drawRect(400,200,650,700,paint1);

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

canvas.drawText("Circle",120,150,paint3);
canvas.drawCircle(200,350,150,paint3);

//draw line
canvas.drawText("Triangle",430,800,paint2);
```

```
    canvas.drawLine(520,850,460,1150,paint2);  
    canvas.drawLine(520,850,580,1150,paint2);  
    canvas.drawLine(460,1150,580,1150,paint2);  
}  
}
```

activity_main.xml

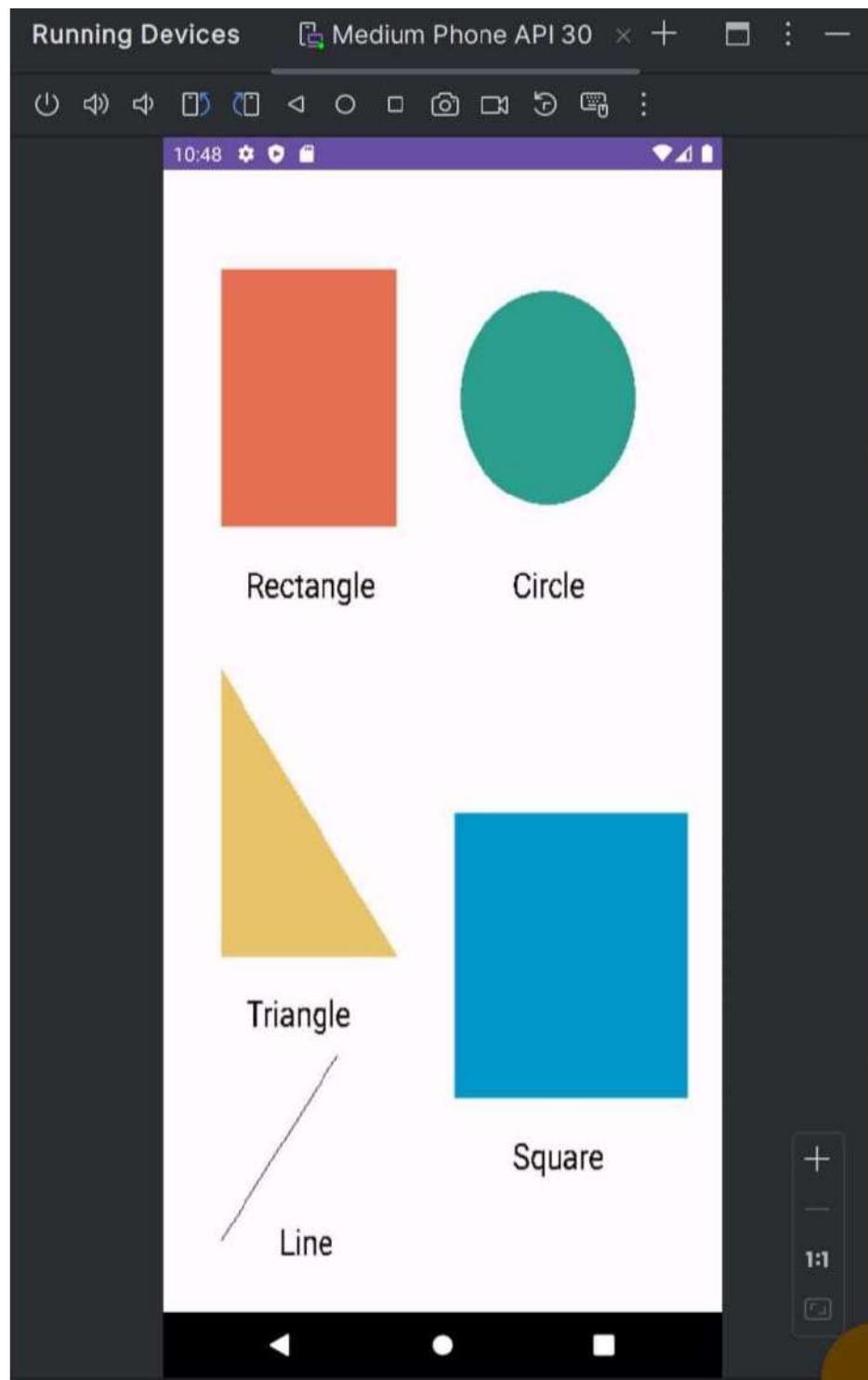
```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    tools:context=".MainActivity">  
  
    <ImageView  
        android:layout_width="match_parent"  
        android:layout_height="match_parent"  
        android:id="@+id/imageView"/>  
  
</LinearLayout>
```

AndroidManifest.xml

```
android:label="Exp 1 Parth Dabholkar C12 2103032" (rest all code remains the same..)
```

Output:-

Parth Dabholkar C12-2103032



EXPERIMENT NO: 2

(C) 

AIM: Write an android application to draw basic 3-D graphical primitive.

THEORY:

Android studio is the official integrated development environment for android application development. Android studio provides more features that enhance our productivity while building android apps.

Built-in functions used in the code:

a) `Bitmap.createBitmap(720, 1280, Bitmap.Config.ARGB_8888);`

It creates a new bitmap with a specified width, height and ARGB_8888 pixel format, suitable to use as a drawing canvas.

b) `Canvas(canvas):`

Creates a canvas object associated with the given bitmap allowing a drawing operation on the Bitmap.

c) `Paint.setColor(Color.Blue):`

Sets the color of the paint object to blue, defining the color for subsequent drawing operations.

d) Paint.set textSize(50):

Sets the text size of the paint object to 50, determining the font size for text drawing.

e) canvas.drawText("Prism", 420, 180, paint):

Draws the text "Prism" on the canvas using the specified paint object.

f) canvas.drawLine(520, 480, 520, 1180):

Draws a line on the canvas from coordinates (x₁, y₁) to (x₂, y₂) using the current paint color.

g) ImageView.setBackgroundDrawable:

It sets the background drawable of the image view to the created Bitmap.

h) canvas.drawArc:

RectF rectF = new RectF(100, 1000, 300, 1200);
canvas.drawArc(rectF, 0, 180, true, paint)

Draws an arc representing a pattern/portion of ellipse

Conclusion: Thus, we understand the concept of drawing 2D shapes using android studio code.

Experiment 2

Aim: Write an android application (WAA) to draw basic graphical 3D primitives.

Program:

MainActivity.java

```
package com.example.Exp2;

import androidx.appcompat.app.AppCompatActivity;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.widget.ImageView;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Bitmap bg=Bitmap.createBitmap(720,1280, Bitmap.Config.ARGB_8888);
        ImageView img=findViewById(R.id.imageView);
        img.setImageBitmap(bg);

        Canvas canvas=new Canvas(bg);
```

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

Paint paint2 = new Paint();
paint2.setColor(Color.BLUE);
paint2.setTextSize(50);

Paint paint = new Paint();
paint.setColor(Color.BLACK);
paint.setTextSize(50);

canvas.drawLine(0,200,300,200,paint);
canvas.drawLine(300,200,300,600,paint);
canvas.drawLine(300,600,0,600,paint);
canvas.drawLine(0,600,0,200,paint);

canvas.drawLine(100,50,400,50,paint);
canvas.drawLine(400,50,400,450,paint);
canvas.drawLine(400,450,100,450,paint);
canvas.drawLine(100,450,100,50,paint);

canvas.drawLine(100,50,0,200,paint);
canvas.drawLine(400,50,300,200,paint);

canvas.drawLine(300,600,400,450,paint);
canvas.drawLine(100,450,0,600,paint);
```

```
        canvas.drawLine(90,900,400,800,paint);
        canvas.drawLine(90,900,50,1100,paint);
        canvas.drawLine(90,900,140,1100,paint);
        canvas.drawLine(140,1100,50,1100,paint);

        canvas.drawLine(140,1100,450,1000,paint);

        canvas.drawLine(450,1000,400,800,paint);

        canvas.drawLine(400,800,360,1000,paint);
        canvas.drawLine(360,1000,450,1000,paint);

        canvas.drawLine(360,1000,50,1100,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">

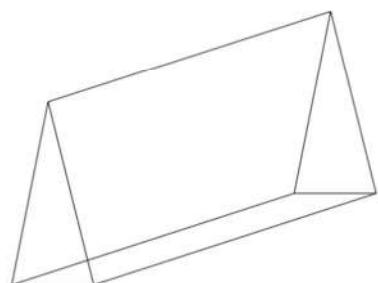
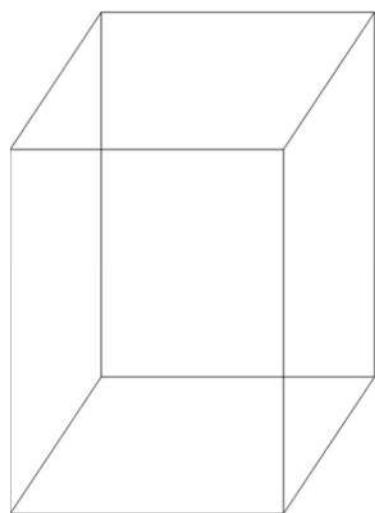
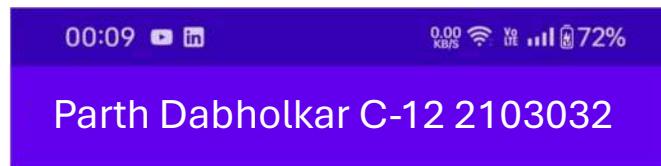
    <ImageView android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/imageView" />

</RelativeLayout>
```

AndroidManifest.xml

```
android:label="Exp 2 Parth Dabholkar C12 2103032"
```

Output:-



EXPERIMENT NO:3

(C) ~~32~~

AIM: write an Android application to design a form with GUI components.

THEORY: In android development, UI are crucial for providing a seamless and intuitive experience. One common aspect is designing forms that allow easy access users to input information. This theory discusses the development of a simple android application using android studio, which creates a form with GUI components.

Built-in Functions:

1) onCreate:

The 'onCreate' method, a vital lifecycle method, initiates app's activity, ensuring seamless connectivity.

2) setOnClickListener(): The setOnClickListener function is pivotal for user interaction. By defining a click listener for the submit button, it orchestrates the behaviour triggered upon button press, creating a responsive and interactive user experience.

3) findViewById: The findViewById function is employed to bridge the gap between the visual and logical aspects of the application. It locates and retrieves references to various GUI components, allowing the code to interact dynamically with UI.

4) getText() and toString(): This function extracts the input from EditTexts, fostering dynamic content handling. Coupled with 'toString', it converts the user's input into a manipulable String, enabling further processing and validation.

5) getCheckedRadioButtonId): With the Radio Group, the getChecked RadioButtonId function becomes instrumental. It identifies the selected Radio Button, facilitating gender selection and enhancing form's adaptability.

6) isChecked(): This function, employed with the Checkbox, serves as crucial input validation tool. It checks whether the user has agreed to the terms, ensuring a very comprehensive and error-free submission.

7) toast.makeText(): This function is judiciously utilized to communicate with the user. It provides concise yet informative feedback, thus enhancing the overall user experience by offering real-time notifications and acknowledgement of successful submissions.

Conclusion:

In this comprehensive exploration of Android Form development, with deeper understanding of built-in functions and processes, we understand how to design intricate and user-centric forms, setting stage for complex mobile applications.

Experiment 3

Aim: Write an android application to design a form with GUI components.

Program:

MainActivity.java

```
package com.example.Exp3;

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

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 + " Last Name: " + lastName +
                " Email: " + email + " Password: " + password;

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

buttonReset.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Handle reset button click
        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"/> 
```

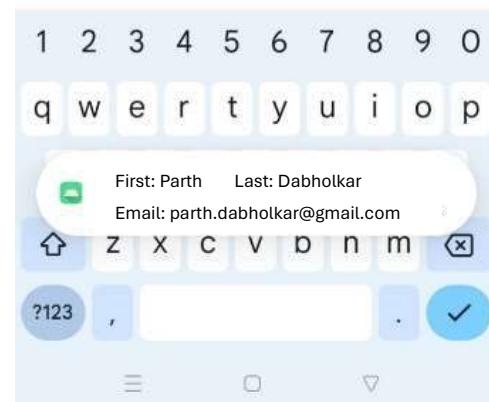
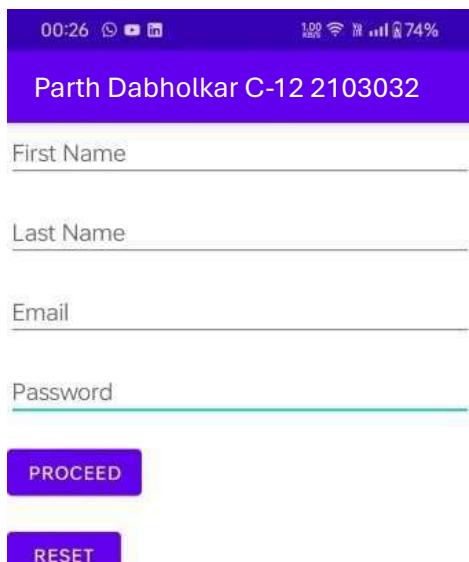
```
<Button  
    android:id="@+id/buttonReset"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_below="@+id/buttonProceed" 
```

```
    android:layout_marginTop="16dp"  
    android:text="Reset"/>  
</RelativeLayout>
```

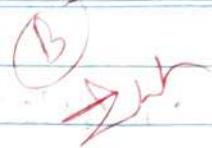
AndroidManifest.xml

```
    android:label="Exp 3 Parth Dabholkar C12 2103032"
```

Output:-



EXPERIMENT NO: 4 :



AIM: WAP to design GUI components using database.

THEORY:

The built in methods are:-

- `setContentView (R.layout.activity_main)`
→ sets the activity content to an explicit view defined by layout resource `activity-main.xml`.
- `findViewById (R.id.Rollno)`
`findViewById (R.id.name)`
`findViewById (R.id.Insert)`
→ finds a view that was identified by the `android:id` XML tag.
- `openOrCreateDatabase (StudentDB, Context.MODE_PRIVATE)`
→ opens or creates a database with the specified name and mode, using the provided cursor factory.
- `execSQL (String sql)`:
→ executes a single SQL statement that is not a `SELECT/INSERT/UPDATE/DELETE` statement.
- `rawQuery (SELECT * FROM STUDENT WHERE Rollno = " " + Rollno.getText() + " ", null)`
→ executes the SQL query and returns a cursor over the result set.

- `moveToFirst()`:
→ Moves the cursor to first row.
- `setText()`
Parameter: `c.getString(1)`.
→ sets the text to be displayed in an EditText or TextView.
- `showMessage(Success, Record Added)`
→ Displays an alert dialog with specified title and message.
- `clearText()`:
→ Clears the text in the EditText Field.
- `onCreate(Bundle savedInstanceState)`
→ Called when the activity is first created. It's where initialization of UI components and database setup occur.

Conclusion : Therefore, we learn to connect a GUI form to a database in android studio code using SQLite for creating a dynamic and data-driven applications. Utilizing this techniques like SQLite ensures efficient data storage and retrieval, enhancing user experience and enabling seamless data management.

Experiment 4

Aim: Write an android application (WAA) to design GUI components with database.

Program:

MainActivity.java

```
package com.example.Exp4;

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 courseNameEdt, courseTracksEdt, courseDurationEdt,
    courseDescriptionEdt;
    private Button addCourseBtn;
    private DBHandler dbHandler;

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

```
courseNameEdt = findViewById(R.id.idEdtCourseName);
courseTracksEdt = findViewById(R.id.idEdtCourseTracks);

courseDurationEdt = findViewById(R.id.idEdtCourseDuration);

courseDescriptionEdt = findViewById(R.id.idEdtCourseDescription);
addCourseBtn = findViewById(R.id.idBtnAddCourse);

dbHandler = new DBHandler(MainActivity.this);

addCourseBtn.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {

        String courseName = courseNameEdt.getText().toString();
        String courseTracks = courseTracksEdt.getText().toString();
        String courseDuration = courseDurationEdt.getText().toString();
        String courseDescription = courseDescriptionEdt.getText().toString();

        if (courseName.isEmpty() && courseTracks.isEmpty() &&
courseDuration.isEmpty() && courseDescription.isEmpty()) {
            Toast.makeText(MainActivity.this, "Please enter all the data..",
Toast.LENGTH_SHORT).show();
            return;
        }

        dbHandler.addNewCourse(courseName, courseDuration, courseDescription,
courseTracks);
```

```
        Toast.makeText(MainActivity.this, "Course has been added.",  
        Toast.LENGTH_SHORT).show();  
  
        courseNameEdt.setText("");  
        courseDurationEdt.setText("");  
        courseTracksEdt.setText("");  
        courseDescriptionEdt.setText("");  
  
    }  
  
});  
  
}  
  
}
```

DBHandler.java

```
package com.example.Exp4;  
  
import android.content.ContentValues;  
import android.content.Context;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
  
  
public class DBHandler extends SQLiteOpenHelper {  
  
  
    private static final String DB_NAME = "parthDB";  
  
  
    private static final int DB_VERSION = 1;  
  
  
    private static final String TABLE_NAME = "mycourses";
```

```
private static final String ID_COL = "id";  
  
private static final String NAME_COL = "name";  
  
private static final String DURATION_COL = "duration";  
  
private static final String DESCRIPTION_COL = "description";  
  
private static final String TRACKS_COL = "tracks";  
  
public DBHandler(Context context) {  
    super(context, DB_NAME, null, DB_VERSION);  
}  
  
@Override  
public void onCreate(SQLiteDatabase db) {  
    String query = "CREATE TABLE " + TABLE_NAME + " ("  
        + ID_COL + " INTEGER PRIMARY KEY AUTOINCREMENT,"  
        + NAME_COL + " TEXT,"  
        + DURATION_COL + " TEXT,"  
        + DESCRIPTION_COL + " TEXT,"  
        + TRACKS_COL + " TEXT)";  
  
    db.execSQL(query);  
}  
  
public void addNewCourse(String courseName, String courseDuration, String  
courseDescription, String courseTracks) {
```

```
SQLiteDatabase db = this.getWritableDatabase();

ContentValues values = new ContentValues();

values.put(NAME_COL, courseName);
values.put(DURATION_COL, courseDuration);
values.put(DESCRIPTION_COL, courseDescription);
values.put(TRACKS_COL, courseTracks);

db.insert(TABLE_NAME, null, values);
db.close();
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
    onCreate(db);
}
}
```

AndroidManifest.xml

```
android:label="Exp 4 Parth Dabholkar C12 2103032"
```

Output:-

12:41 80% Parth Dabholkar C-12 2103032

CSS

4 months

AES, DES

Cryptography

Add Course

12:40 80% Parth Dabholkar C-12 2103032

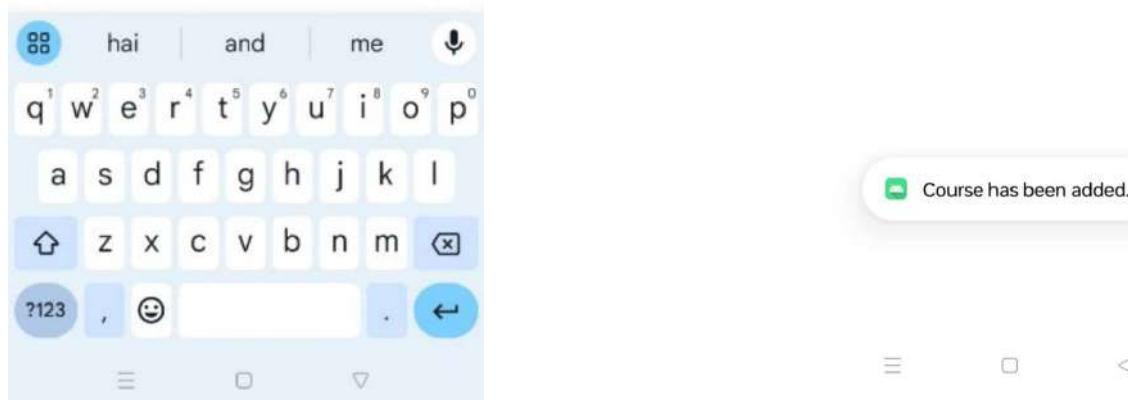
Enter course Name

Enter Course Duration

Enter Course Tracks

Enter Course Description

Add Course



Open File Parth db1.d mycourses

Search tables... Reset Filters Records: 2 Search 2 records...

Tables (3)

mycourses

sqlite_sequence

		id	name	duration	description	tracks
		1	spcc	2hrs	lexical analysis	chp1
		2	CSS	4 months	Cryptography	AES,DES

EXPERIMENT NO: 5

(B) ✓

AIM: Write an android application to develop an EMS calculator application.

THEORY:

This experiment's code implements an EMS (Equated Monthly Instalment) calculator android application. Users input the principal amount, interest rate, and loan period, and the application calculates and displays the monthly EMI. The interface contains EditText fields for user input and a button to trigger calculation. Upon clicking the button, the EMI is shown on a TextView.

Built-In Functions:

- 1) onCreate: This method is called when the activity is first created. Here, we initialize the UI components and set the click listener for the calculate/submit button.
- 2) calculateEMI: This method calculates the EMI based on the provided principal amount, interest rate and loan period. It then sets the result in the 'resultView' text.
- 3) setOnClickListener: This method sets a click Listener on the calculate Button. When the button is clicked, it triggers the 'calculateEMI' method to calculate and display the EMI.

4) RelativeLayout : This is a layout manager that arranges the child views relative to each other or the parent. In this layout, child views are positioned relative to the parent or to each other using attributes such as: 'android:layout_below', 'android:layout_above'.

5) EditText : This tag represents a text field where users can input text. Attributes here are:

- android_id : Unique identifier for view.
- android_layout_width, height : Width and height of the view.
- android_hint : Hint text displayed when field is empty.
- android_input_type : Specifies the type of data the field accepts.

6) Button : Represents a push button widget. Attributes used:

- android_id : Unique identifier.
- android_width : Width of button.
- android_height : Height of button.
- android_text : Text displayed on button.

7) TextView : Displays text to the user. Attributes used are:

- android_id : Unique identifier.
- android_layout_width : Width of TextView.
- android_layout_height : Height of TextView.
- android_text : Text displayed on the screen.

Conclusion : The assignment aimed to develop a simple EM2 calculator android application using android studio. We learned how to create a basic UI using XML layout files and handle user interactions in Java code.

Experiment 5

Aim: Write an android application (WAA) to calculate EMI payment.

Program:

MainActivity.java

```
package com.example.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 = callInt(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 callInt(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);
}

}
```

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.exp5.MainActivity"
    android:layout_height="match_parent"
    android:background="@color/white"
    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">
```

32

```
<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>

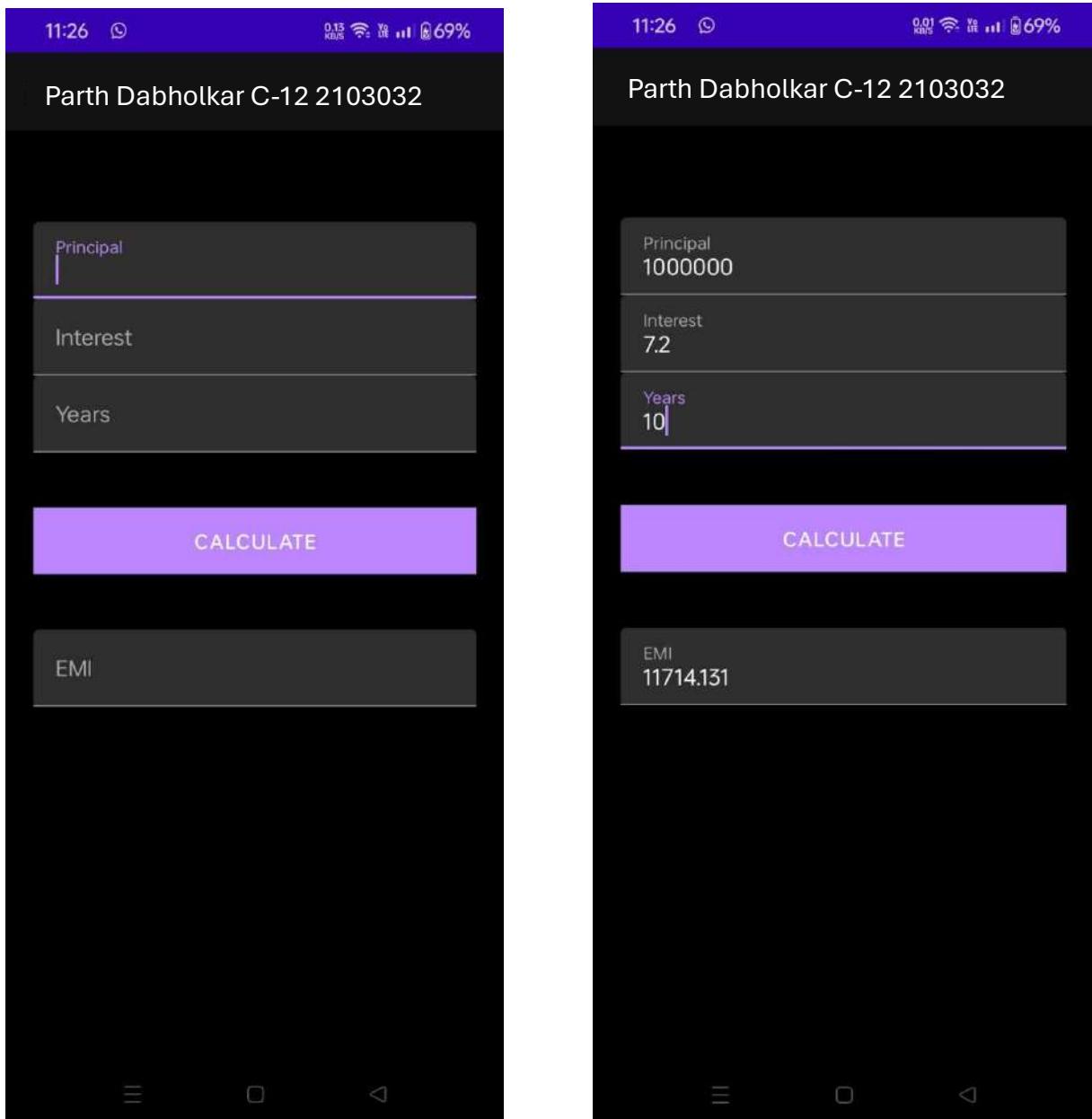
<Button android:id="@+id/btn_calculate2"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Calculate"
        android:background="#000000"
        android:layout_marginTop="40dp"
        android:textColor="#FFFFFF"/>

<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>
```

AndroidManifest.xml

```
android:label="Exp 5 Parth Dabholkar C12 2103032"
```

Output:-



EXPERIMENT NO: 6

(A)
Date

AIM: Write an android application that create an alert on receiving a message.

THEORY:

- Create a basic application:
 - A notification is the most basic and compact element also known as collapsed form - displays an icon, a title, and a small amount of text content. This experiment shows how to create a notification that can tap to launch an activity in your application.
- Declare the runtime permission:
 - Android 13 and higher supports a runtime permission for posting non-exempt notification from an app. The permission that you need to declare in your app's manifest file appear in the following:
- set the notification content:
 - Two get started set two notification content channel using a notification commandBuilder object. The following example shows how to create a notification:

Eg: A small icon is set by setSmallIcon(). This is the only user-visible content that is required.

- The body text set by setContentText().

- Create a channel and set the importance:

Before you can deliver the notification on Android 8.0 and later, register your app's notification channel with system by passing an integer of notification channel to `createNotificationChannel()`.

- Set notification's tap action:

Every notification must respond to a tap, usually to open an activity in your app that corresponds to notification. To do so specify a content intent defined with a pending Intent object and pass it to `setContentIntent()`.

To make the notification appear, call `NotificationManager.notify()` passing it a unique ID for notification.

Conclusion: Hence successfully notification on receiving a message or performing a certain action.

Experiment 6

Aim: Write an android application to push notifications using Android Studio.

Program:

MainActivity.java

```
package com.example.Exp6;

import android.app.AlertDialog;
import android.content.DialogInterface;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
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 showDialogButton = findViewById(R.id.showAlertDialogButton);
        showDialogButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                showAlert();
            }
        });
    }

    private void showAlert() {
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle("Alert Dialog");
        builder.setMessage("This is an alert dialog!");
        builder.setPositiveButton("OK", null);
        builder.setNegativeButton("Cancel", null);
        builder.show();
    }
}
```

```
    }

});

}

private void showAlert() {

    AlertDialog.Builder builder = new AlertDialog.Builder(this);

    builder.setTitle("Alert");

    builder.setMessage("This is an alert dialog!");

    builder.setPositiveButton("OK", new DialogInterface.OnClickListener() {

        public void onClick(DialogInterface dialog, int id) {

        }

    });

    AlertDialog dialog = builder.create();

    dialog.show();

}

}
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:tools="http://schemas.android.com/tools">

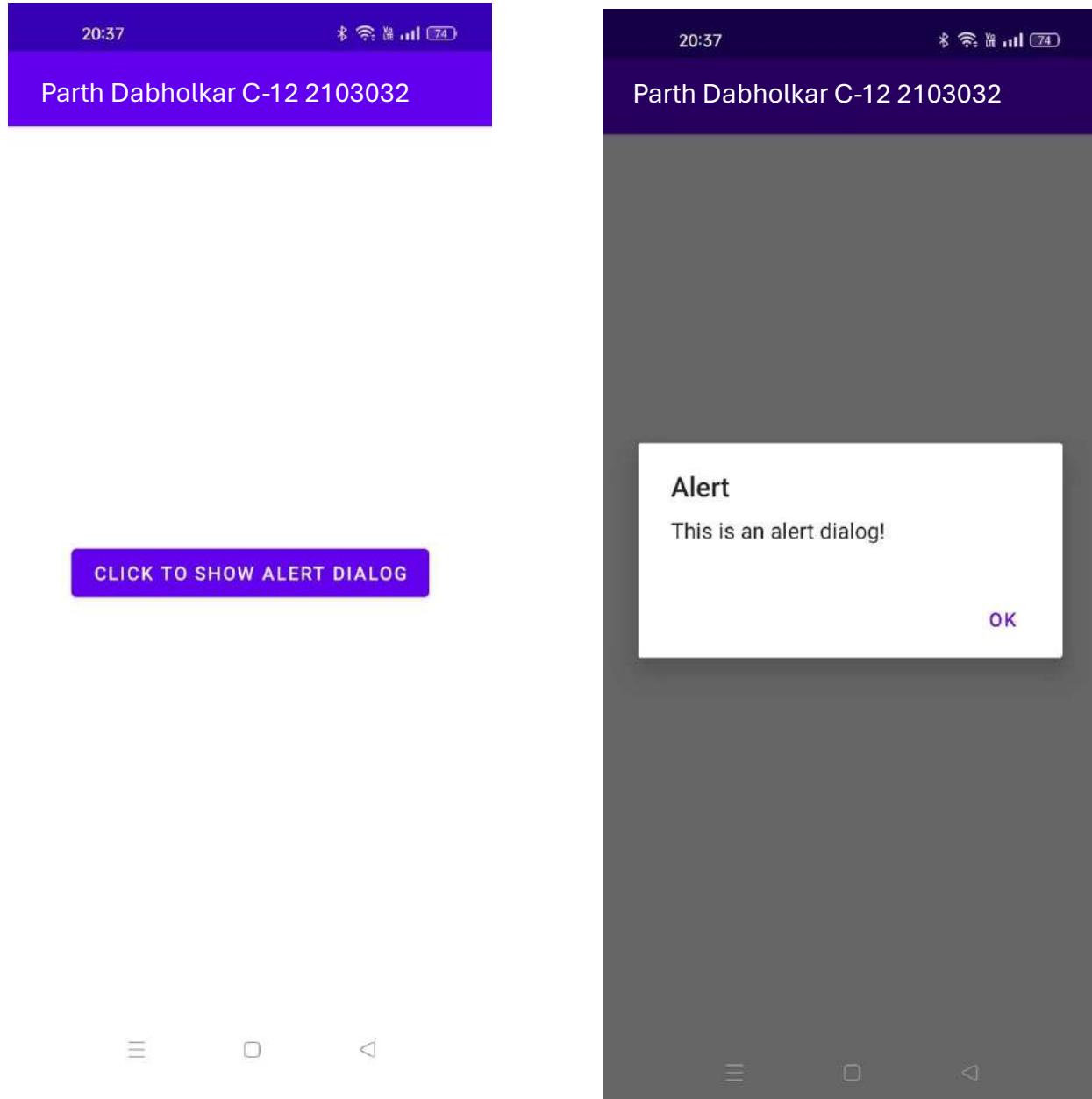
    <application

        android:allowBackup="true"

        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="Exp 6 Parth Dabholkar C12 2103032"
```

```
    android:roundIcon="@mipmap/ic_launcher_round"  
    android:supportsRtl="true"  
    android:theme="@style/Theme.Exp6"  
    tools:targetApi="31">  
  
    <activity  
        android:name=".MainActivity"  
        android:exported="true">  
        <intent-filter>  
            <action android:name="android.intent.action.MAIN" />  
  
            <category android:name="android.intent.category.LAUNCHER" />  
        </intent-filter>  
  
        <meta-data  
            android:name="android.app.lib_name"  
            android:value="" />  
    </activity>  
  
    </application>  
  
</manifest>
```

Output:



EXPERIMENT NO: 7

(A) ~~7~~

AIM: Write an Android Application to develop basic math calculator.

THEORY: This experiment focuses on Android Application on a basic calculator using Java in Android Studio Code. It allows users to perform arithmetic operations (addition, subtraction, multiplication and division) on two input numbers and also calculate the square root of a single input number.

Built-in functions used are:

1. onCreate(): This method is called when the activity is created. It initializes UI elements and sets click listeners for the buttons.
2. performCalculation(char operator): This method performs arithmetic calculations based on operator (+, -, *, /). It retrieves input values from EditText fields, checks for empty inputs, performs the calculation, and displays the result in the TextView.
3. calculateSquareRoot(): This method calculates the square root of a number entered in one of the EditText fields. It retrieves the input value, checks for empty input value, checks for empty input; calculates the square root using Math.sqrt() function and display result in TextView.
4. DecimalFormat: This class is used to format numerical values.

in this case, to display the result with maximum of two decimal places.

5. findViewById(Id): This method is used to find and return a view from the layout based on its ID.
 6. setOnClickListener(): This method is used to set a click listener on the button to perform a specific action when the button is clicked.
 7. Toast.makeText(): This method creates a toast message, a small pop-up message that briefly appears on the screen to provide feedback to the user.
 8. <EditText>: defines an input field where users can enter numeric values.
 9. <Buttons>: defines a clickable button for performing arithmetic operations or other actions.
 10. <TextView>: defines a non-editable text field for displaying results or other information.
 11. ConstraintLayout: Layout manager used to create flexible layouts with constraints.
- # Conclusion: The provided code implements a basic calculator app in Android Studio using Java. It helps in grasping essential concepts in app development.

Experiment 7

Aim: Write an android application for basic operations on calculator.

Program:

MainActivity.java

```
package com.example.Exp7;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {

    EditText etFirstNumber, etSecondNumber;
    TextView tvResult;
    Button btnAdd, btnSubtract, btnMultiply, btnDivide, btnClear;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        etFirstNumber = findViewById(R.id.et_first_number);
        etSecondNumber = findViewById(R.id.et_second_number);
        tvResult = findViewById(R.id.tv_result);
        btnAdd = findViewById(R.id.btn_add);
        btnSubtract = findViewById(R.id.btn_subtract);
```

```
btnMultiply = findViewById(R.id.btn_multiply);
btnDivide = findViewById(R.id.btn_divide);
btnClear = findViewById(R.id.btn_clear);
btnAdd.setOnClickListener(v -> {
    int firstNumber = Integer.parseInt(etFirstNumber.getText().toString());
    int secondNumber =
        Integer.parseInt(etSecondNumber.getText().toString());
    int result = firstNumber + secondNumber;
    tvResult.setText(result+" ");
});
btnSubtract.setOnClickListener(v -> {
    int firstNumber = Integer.parseInt(etFirstNumber.getText().toString());
    int secondNumber =
        Integer.parseInt(etSecondNumber.getText().toString());
    int result = firstNumber - secondNumber;
    tvResult.setText(result+" ");
});
btnMultiply.setOnClickListener(v -> {
    int firstNumber = Integer.parseInt(etFirstNumber.getText().toString());
    int secondNumber =
        Integer.parseInt(etSecondNumber.getText().toString());
    int result = firstNumber * secondNumber;
    tvResult.setText(result+" ");
});
btnDivide.setOnClickListener(v -> {
    int firstNumber = Integer.parseInt(etFirstNumber.getText().toString());
```

```
        int secondNumber =  
            Integer.parseInt(etSecondNumber.getText().toString());  
  
        int result = firstNumber / secondNumber;  
  
        tvResult.setText(result + " ");  
    }  
  
    btnClear.setOnClickListener(v -> {  
  
        etFirstNumber.setText("");  
  
        etSecondNumber.setText("");  
  
        tvResult.setText("");  
    });  
}  
}
```

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">  
  
    <EditText  
        android:id="@+id/et_first_number"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:ems="10"  
        android:inputType="number"  
        android:layout_marginTop="50dp"  
        android:hint="First Number"/>
```

```
<EditText  
    android:id="@+id/et_second_number"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:ems="10"  
    android:inputType="number"  
    android:layout_marginTop="50dp"  
    android:hint="Second Number"/>  
  
<TextView  
    android:id="@+id/tv_result"  
    android:layout_width="match_parent"  
    android:layout_height="35dp"  
    android:textSize="30sp"  
    android:textColor="#000000"  
    android:text="Result: "  
    android:layout_marginTop="50dp"  
    android:paddingLeft="20dp"/>  
  
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="50dp"  
    android:layout_marginRight="50dp"  
  
    android:layout_marginLeft="50dp">  
  
<Button  
    android:id="@+id/btn_add"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"
```

```
    android:text="+"
    android:layout_marginLeft="30dp"
    android:layout_marginRight="20dp"/>
<Button
    android:id="@+id	btn_subtract"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=""

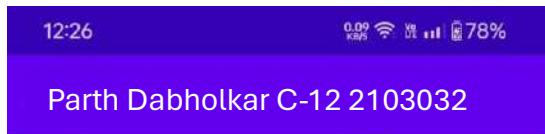
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="50dp"
    android:layout_marginRight="50dp"
    android:layout_marginLeft="50dp">
    <Button
        android:id="@+id	btn_multiply"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="*"
        android:layout_marginLeft="30dp"
        android:layout_marginRight="20dp"/>
    <Button
```

```
    android:id="@+id	btn_divide"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="/"   
    android:layout_marginRight="20dp"  
    android:layout_marginLeft="20dp"/> >  
</LinearLayout>  
  
<Button  
    android:id="@+id	btn_clear"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="clear"  
    android:layout_marginTop="50dp"  
    android:layout_gravity="center"/> >  
</LinearLayout>
```

AndroidManifest.xml

```
    android:label="Exp 7 Parth Dabholkar C12 2103032"
```

Output:-



9

5

3

3

3

15

+ -

+ -

* /

* /

CLEAR

CLEAR

☰ ☐ ⌂

☰ ☐ ⌂

EXPERIMENT NO: 8

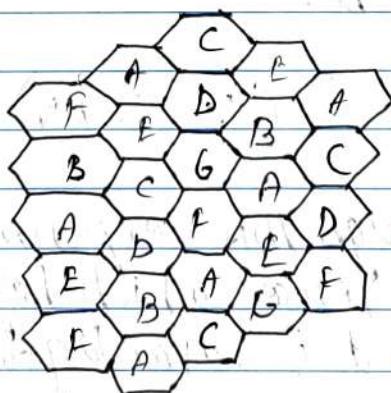
(A+)
Date

AIM: Write a program to demonstrate cellular frequency reuse.

THEORY: Frequency reuse is the scheme in which allocation and reuse of channels throughout a coverage region is done. Each cellular base station is allocated a group of radio channels or frequency sub-bands to be used within a small geographical area known as cells. The shape of cell is hexagonal.

The process of selecting and allocating the frequency sub-bands for all the cellular base station within a system is called frequency reuse or frequency Planning.

- Advantages:
- i) It improves the quality of service (QoS).
 - ii) In frequency reuse scheme, total bandwidth is divided into different sub-bandwidths that are used by the cells.
 - iii) Frequency reuse scheme allow the operators to reuse the same frequencies at different cell sites.



Cell with same letters use same set of channels groups or frequency sub-band.

To find the total number of channels allocated to a cell:

$S = \text{total number of duplex channels to use available}$

$R = \text{channels allocated to the each cell } (k < s)$

$N = \text{total no. of cells or cluster size.}$

Thus, $S = KN$

frequency reuse factor $= 1/N$.

The value of N is calculated by the formula:

$$N = I^2 + J^2 + \cancel{S} I * J$$

where: I, J : Positive integer indicating position of cell.

N : total number of cells / size of cluster.

If a cluster is repeated or replicated M times, then capacity C will be:

$$C = MKN = MS \quad (\because S = KN)$$

Conclusion: Through this experiment, we learn the concept of Frequency reuse which helps in quality of service (QoS) and is advantageous for using same frequency bands.

Code:**Exp8FrequencyReuse.py :**

```
#!/usr/bin/python

from math import *

# import everything from Tkinter module
from tkinter import *

# Base class for Hexagon shape
class Hexagon(object):

    def __init__(self, parent, x, y, length, color, tags):
        self.parent = parent
        self.x = x
        self.y = y
        self.length = length
        self.color = color
        self.size = None
        self.tags = tags
        self.draw_hex()

    # draw one hexagon
```

```
def draw_hex(self):
    start_x = self.x
    start_y = self.y
    angle = 60
    coords = []
    for i in range(6):
        end_x = start_x + self.length * cos(radians(angle * i))
        end_y = start_y + self.length * sin(radians(angle * i))
        coords.append([start_x, start_y])
        start_x = end_x
        start_y = end_y
    self.parent.create_polygon(coords[0][0],
                               coords[0][1],
                               coords[1][0],
                               coords[1][1],
                               coords[2][0],
                               coords[2][1],
                               coords[3][0],
                               coords[3][1],
                               coords[4][0],
                               coords[4][1],
                               coords[5][0],
                               coords[5][1],
                               fill=self.color,
                               outline="black",
```

```
tags=self.tags)

# class holds frequency reuse logic and related methods
class FrequencyReuse(Tk):

    CANVAS_WIDTH = 800
    CANVAS_HEIGHT = 650
    TOP_LEFT = (20, 20)
    BOTTOM_LEFT = (790, 560)
    TOP_RIGHT = (780, 20)
    BOTTOM_RIGHT = (780, 560)

    def __init__(self, cluster_size, columns=16, rows=10, edge_len=30):
        Tk.__init__(self)
        self.textbox = None
        self.curr_angle = 330
        self.first_click = True
        self.reset = False
        self.edge_len = edge_len
        self.cluster_size = cluster_size
        self.reuse_list = []
        self.all_selected = False
        self.curr_count = 0
        self.hexagons = []
        self.co_cell_endp = []
        self.reuse_xy = []
```

```

self.canvas = Canvas(self,
                     width=self.CANVAS_WIDTH,
                     height=self.CANVAS_HEIGHT,
                     bg="#4dd0e1")

self.canvas.bind("<Button-1>", self.call_back)

self.canvas.focus_set()

self.canvas.bind('<Shift-R>', self.resets)

self.canvas.pack()

self.title("Frequency reuse and co-channel selection")

self.create_grid(16, 10)

self.create_textbox()

self.cluster_reuse_calc()

# show lines joining all co-channel cells

def show_lines(self):

    # center(x,y) of first hexagon

    approx_center = self.co_cell_endp[0]

    self.line_ids = []

    for k in range(1, len(self.co_cell_endp)):

        end_xx = (self.co_cell_endp[k])[0]
        end_yy = (self.co_cell_endp[k])[1]

        # move i^th steps

```

```

    l_id = self.canvas.create_line(approx_center[0],
approx_center[1],
                                         end_xx,
end_yy)

    if j == 0:

        self.line_ids.append(l_id)

        dist = 0

    elif i >= j and j != 0:

        self.line_ids.append(l_id)

        dist = j

        # rotate counter-clockwise and move j^th step

        l_id = self.canvas.create_line(
            end_xx, end_yy, end_xx + self.center_dist *
dist *

            cos(radians(self.curr_angle - 60)),
            end_yy + self.center_dist * dist *

            sin(radians(self.curr_angle - 60)))

        self.line_ids.append(l_id)

        self.curr_angle -= 60


def create_textbox(self):

    txt = Text(self.canvas,
               width=80,
               height=1,
               font=("Helvetica", 12),
               padx=10,

```

```
        pady=10)

    txt.tag_configure("center", justify="center")
    txt.insert("1.0", "Select a Hexagon")
    txt.tag_add("center", "1.0", "end")
    self.canvas.create_window((0, 600), anchor='w', window=txt)
    txt.config(state=DISABLED)
    self.textbox = txt

def resets(self, event):
    if event.char == 'R':
        self.reset_grid()

# clear hexagonal grid for new i/p
def reset_grid(self, button_reset=False):
    self.first_click = True
    self.curr_angle = 330
    self.curr_count = 0
    self.co_cell_endp = []
    self.reuse_list = []
    for i in self.hexagons:
        self.canvas.itemconfigure(i.tags, fill=i.color)

    try:
        self.line_ids
    except AttributeError:
```

```

    pass

else:
    for i in self.line_ids:
        self.canvas.after(0, self.canvas.delete, i)
    self.line_ids = []

if button_reset:
    self.write_text("Select a Hexagon")

# create a grid of Hexagons

def create_grid(self, cols, rows):
    size = self.edge_len
    for c in range(cols):
        if c % 2 == 0:
            offset = 0
        else:
            offset = size * sqrt(3) / 2
        for r in range(rows):
            x = c * (self.edge_len * 1.5) + 50
            y = (r * (self.edge_len * sqrt(3))) + offset + 15
            hx = Hexagon(self.canvas, x, y, self.edge_len,
"#fafafa",
                        "{}{},{}".format(r, c))
            self.hexagons.append(hx)

```

```

# calculate reuse distance, center distance and radius of the
hexagon

def cluster_reuse_calc(self):
    self.hex_radius = sqrt(3) / 2 * self.edge_len
    self.center_dist = sqrt(3) * self.hex_radius
    self.reuse_dist = self.hex_radius * sqrt(3 * self.cluster_size)

def write_text(self, text):
    self.textbox.config(state=NORMAL)
    self.textbox.delete('1.0', END)
    self.textbox.insert('1.0', text, "center")
    self.textbox.config(state=DISABLED)

#check if the co-channels are within visible canvas

def is_within_bound(self, coords):
    if self.TOP_LEFT[0] < coords[0] < self.BOTTOM_RIGHT[0] \
    and self.TOP_RIGHT[1] < coords[1] < self.BOTTOM_RIGHT[1]:
        return True
    return False

#gets called when user selects a hexagon
#This function applies frequency reuse logic in order to
#figure out the positions of the co-channels
def call_back(self, evt):

```

```

selected_hex_id = self.canvas.find_closest(evt.x, evt.y)[0]
hexagon = self.hexagons[int(selected_hex_id - 1)]
s_x, s_y = hexagon.x, hexagon.y
approx_center = (s_x + 15, s_y + 25)

if self.first_click:
    self.first_click = False
    self.write_text(
        """Now, select another hexagon such
        that it should be a co-cell of
        the original hexagon."""
    )
    self.co_cell_endp.append(approx_center)
    self.canvas.itemconfigure(hexagon.tags, fill="green")

for _ in range(6):
    end_xx = approx_center[0] + self.center_dist * i *
    cos(
        radians(self.curr_angle))
    end_yy = approx_center[1] + self.center_dist * i *
    sin(
        radians(self.curr_angle))

    reuse_x = end_xx + (self.center_dist * j) * cos(
        radians(self.curr_angle - 60))

```

```

reuse_y = end_yy + (self.center_dist * j) * sin(
    radians(self.curr_angle - 60))

if not self.is_within_bound((reuse_x, reuse_y)):
    self.write_text(
        """co-cells are exceeding canvas
boundary.

Select cell in the center"""

    )
    self.reset_grid()
    break

if j == 0:
    self.reuse_list.append(
        self.canvas.find_closest(end_xx,
end_yy)[0])
elif i >= j and j != 0:
    self.reuse_list.append(
        self.canvas.find_closest(reuse_x,
reuse_y)[0])

self.co_cell_endp.append((end_xx, end_yy))
self.curr_angle -= 60

else:
    curr = self.canvas.find_closest(s_x, s_y)[0]

```

```

        if curr in self.reuse_list:
            self.canvas.itemconfigure(hexagon.tags,
fill="green")
            self.write_text("Correct! Cell {} is a co-
cell.".format(
                hexagon.tags))
            if self.curr_count == len(self.reuse_list) - 1:
                self.write_text("Great! Press Shift-R to
restart")
                self.show_lines()
            self.curr_count += 1

    else:
        self.write_text("Incorrect! Cell {} is not a co-
cell.".format(
            hexagon.tags))
        self.canvas.itemconfigure(hexagon.tags, fill="red")

if __name__ == '__main__':
    print(
        """Enter i & j values. common (i,j) values are:
(1,0), (1,1), (2,0), (2,1), (3,0), (2,2)"""
    )
    i = int(input("Enter i: "))
    j = int(input("Enter j: "))
    if i == 0 and j == 0:

```

```
raise ValueError("i & j both cannot be zero")

elif j > i:
    raise ValueError("value of j cannot be greater than i")

else:
    N = (i**2 + i * j + j**2)
    print("N is {}".format(N))

freqreuse = FrequencyReuse(cluster_size=N)
freqreuse.mainloop()
```

Output:

(base) PS C:\Users\ParthDabholkar\OneDrive\Desktop\MCC Exp Documents>python Exp8code.py

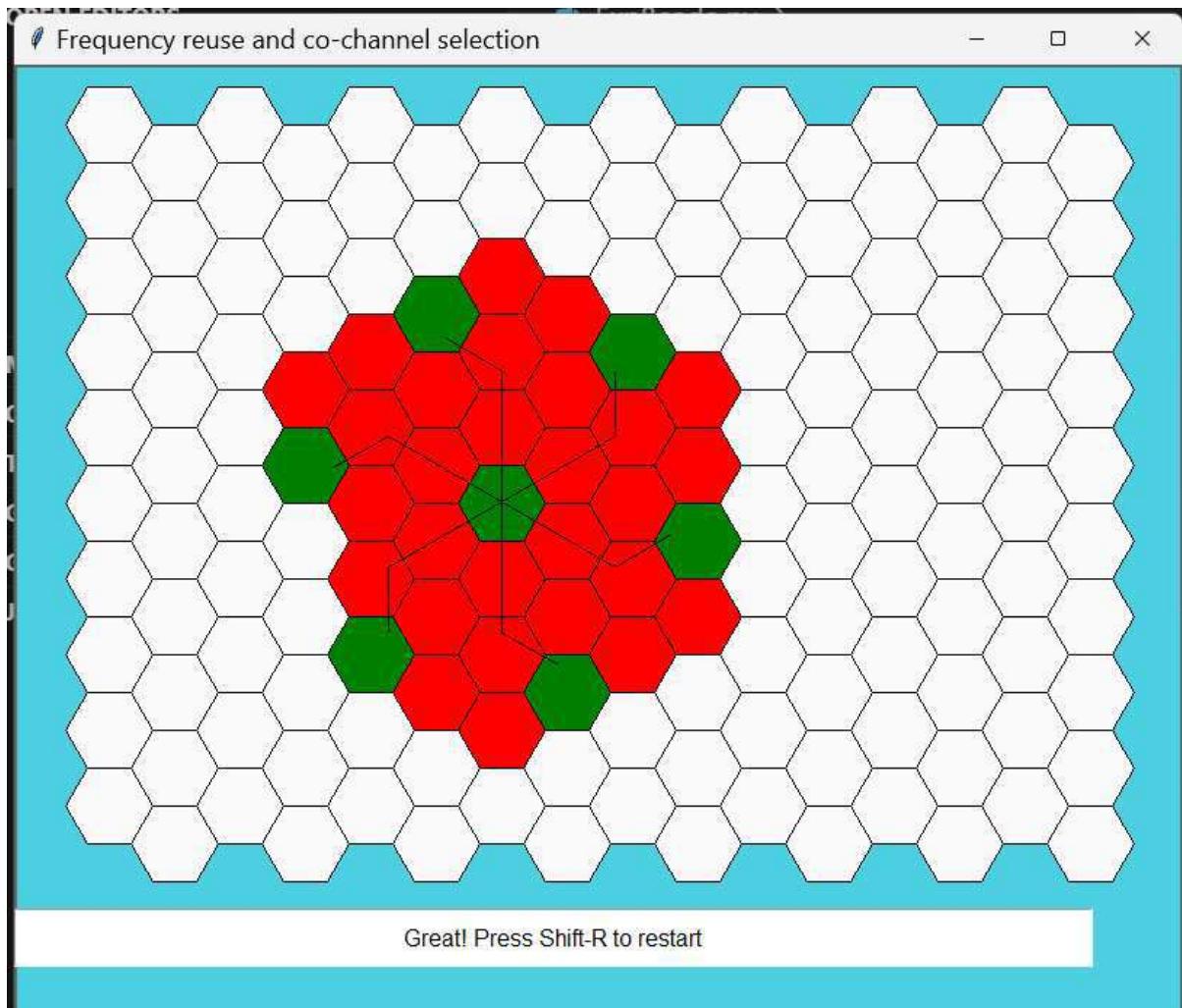
Enter i & j values. common (i,j) values are:

(1,0), (1,1), (2,0), (2,1), (3,0), (2,2)

Enter i: 2

Enter j: 1

N is 7



EXPERIMENT NO: 9

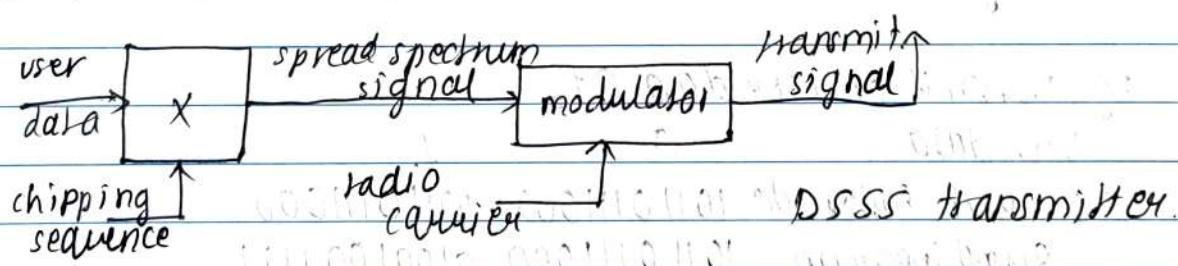
(A) ✓

AIM: write a program to explain the concept of DSSS.

THEORY: Spread Spectrum includes techniques involving spreading bandwidth needed to transmit data, reducing narrowband interference. Direct Sequence Spread Spectrum (DSSS) systems take a user bit stream and perform an (XOR) with so called chipping sequence. Wireless systems use the sequence 10110111000, called as Barker Code. These Barker code exhibit a good robustness against interference and insensitivity to multi-path propagation.

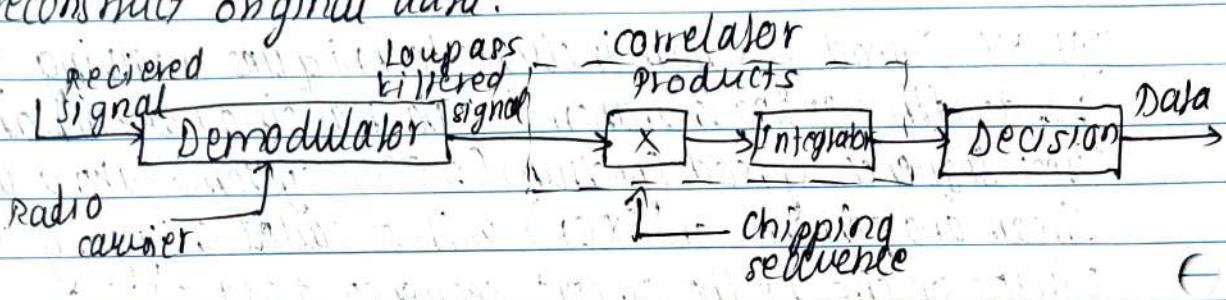
The first step in DSSS transmitter, is the spreading of user data with chipping sequence (digital modulation).

The spread signal is then modulated with a radio carrier (radio modulation).



Assuming a user signal with a bandwidth of 1 MHz, spreading it with 11-chip Barker code would result in a signal with 11 MHz bandwidth. The radio carrier then shifts it to carrier frequency. Eg: 2.4 GHz. This signal is then transmitted further into the system.

DSSS Receiver: The receiver has to perform inverse functions of two transmitters modulation steps. However, noise and multipath propagation require additional mechanisms to reconstruct original data.



First step in receiver DSSS is to demodulate the received signal. This results in a signal with approximately same bandwidth as original spread spectrum signal.

The receiver has to know the original chipping sequence, sequences have to be precisely synchronized since receiver calculates product of chip with incoming signal.

This comprises XOR operation. An integrator integrates all products if integrator sums to less than h take 0 or more than h take 1.

Eg: transmit OR user data 01.

User data	0	1	.	.	.
XOR with Barker code	10110111000	10110111000			
Spread spectrum	10110111000	01001000111			

Received signal : 10110111000 01001000111

XOR Barker code : 10110111000 10110111000

Result : 00000000000 11111111111

Data recovered

Conclusion: Thus, understood the concept of Direct sequence spread spectrum (DSSS) and its significance in mobile computing.

```

import java.util.Arrays;

public class Exp9DSSS {
    public static void main(String[] args) {
        int[] dataSignal = { 0, 1 };
        int[] spreadingCode = { 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0 };

        int[] spreadSignal = spreadDSSS(dataSignal, spreadingCode);

        System.out.println("Original Data Signal: " + Arrays.toString(dataSignal));
        System.out.println("Spreading Code (PN Sequence): " +
                           Arrays.toString(spreadingCode));
        System.out.println("Spread Signal: " + Arrays.toString(spreadSignal));

        int[] recoveredSignal = despreadDSSS(spreadSignal, spreadingCode);

        System.out.println("Recovered Signal: " + Arrays.toString(recoveredSignal));
    }

    private static int[] spreadDSSS(int[] dataSignal, int[] spreadingCode) {
        int[] spreadSignal = new int[dataSignal.length * spreadingCode.length];
        for (int i = 0; i < dataSignal.length; i++) {
            for (int j = 0; j < spreadingCode.length; j++) {
                spreadSignal[i * spreadingCode.length + j] = dataSignal[i] ^
                    spreadingCode[j];
            }
        }
        return spreadSignal;
    }
}

```

```
}
```

```
private static int[] despreadDSSS(int[] spreadSignal, int[] spreadingCode) {
    int length = spreadSignal.length / spreadingCode.length;
    int[] recoveredSignal = new int[length];
    for (int i = 0; i < length; i++) {
        int sum = 0;
        for (int j = 0; j < spreadingCode.length; j++) {
            sum += spreadSignal[i * spreadingCode.length + j] ^ spreadingCode[j];
        }
        System.out.print("Addition of " + " bit " + (i + 1) + " : " + sum);
        recoveredSignal[i] = (sum > 7) ? 1 : 0;
        if (sum > 7) {
            System.out.println(", Since sum is more than 7, it is converted to 1");
        } else {
            System.out.println(", Since sum is less than 4, it is converted to 0");
        }
    }
    return recoveredSignal;
}
```

```
PROBLEMS 46 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\parth\JavaProgram> cd "c:\Users\parth\JavaProgram\" ; if ($?) { javac Exp9DSSS.java } ; if ($?) { java Exp9DSSS }
Original Data Signal: [0, 1]
Spreading Code (PN Sequence): [1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0]
Spread Signal: [1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1]
Addition of bit 1 : 0, Since sum is less than 4, it is converted to 0
Addition of bit 2 : 11, Since sum is more than 7, it is converted to 1
Recovered Signal: [0, 1]
PS C:\Users\parth\JavaProgram>
```

EXPERIMENT NO: 10.

(A) ~~D~~

AIM: Write a program to implement A3/A5/A8 ASM security Algorithm.

THEORY: ASM uses 3 different security mechanisms algorithms called A3, A5, A8. In practice, A3 and A8 are generally implemented together known as A3/A8.

- An A3/A8 algo is implemented in SIM card and in ASM network Authentication centres. It's used to authenticate the customer and generate a key for accepting encrypting voice and data traffic.

Developer of A3 and A8 algo is considered a matter for individual GSM networks operations although implementations are available.

- An A8 encryption algo scrambles the user's voice and data traffic between the handset and the base station to provide privacy. An A3 algo is implemented in both A3 algo. is implemented in both handset and base station subsystem(BSS).

A3 → Authentication algorithm
→ calculates based on key and RAO sent by MSC.
→ not standardized, can be chosen independently by each operator.

A8: → key generation algorithm between
to calculate the session key.

→ calculation of key depends
on RAND.

→ Not standardized, can be
chosen independently by
each operator.

A5: → stream cipher used to encrypt
over the air transmission

→ Ciphering is based on the call
and frame numbers.

→ Specified at international level
to enable roaming.

Conclusion: In conclusion, the implementation of A3, A5, A8
algorithms offers a robust approach to security in mobile
communications. Through their cryptographic mechanisms,
these algorithms provide confidentiality, integrity
and authentication, essential for ensuring secure data
transmission in GSM networks.

Code:**Exp10A3Algo.java :**

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

public class Exp10A3Algo {

    public static void main(String[] args) {
        // Simulate generating a random Ki (secret key) and RAND (challenge)
        String ki = generateRandomHexString(32); // 128 bits (16 bytes) key
        String rand = generateRandomHexString(32); // 128 bits (16 bytes)
        challenge

        // Display the generated Ki and RAND
        System.out.println("Ki (Secret Key): " + ki);
        System.out.println("RAND (Challenge): " + rand);

        // Calculate the expected response (SRES) using the A3 algorithm
        String sres = calculateSRES(ki, rand);

        // Display the calculated SRES
    }
}
```

```
        System.out.println("SRES (Expected Response): " + sres);

    }

private static String generateRandomHexString(int length) {

    Random random = new Random();

    StringBuilder randomHex = new StringBuilder();

    for (int i = 0; i < length; i++) {

        int randomInt = random.nextInt(16); // 0-15

        randomHex.append(Integer.toHexString(randomInt));

    }

    return randomHex.toString();

}

private static String calculateSRES(String ki, String rand) {

    try {

        // Concatenate Ki and RAND

        String input = ki + rand;

        // Use SHA-1 hash function to calculate SRES

        MessageDigest sha1 = MessageDigest.getInstance("SHA-1");

        byte[] hashBytes = sha1.digest(hexStringToByteArray(input));

        // Convert the hash to a hexadecimal string

    }

}
```

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

return sres.toString();
} catch (NoSuchAlgorithmException e) {
    e.printStackTrace();
    return null;
}

}

private static byte[] hexStringToByteArray(String hexString) {
    int len = hexString.length();
    byte[] data = new byte[len / 2];

    for (int i = 0; i < len; i += 2) {
        data[i / 2] = (byte) ((Character.digit(hexString.charAt(i), 16) << 4)
            + Character.digit(hexString.charAt(i + 1), 16));
    }

    return data;
}
```

Output:

```
(base) PS C:\Users\ParthDabholkar\OneDrive\Desktop\MCC Exp Documents> cd "c:\Users\Rishab\OneDrive\Desktop\MCC Exp Documents\" ; if ($?) { javac Exp10A3Algo.java } ; if ($?) { java Exp10A3Algo }
```

Ki (Secret Key): 75b3cd449c491cb7af27683de9dba3f8

RAND (Challenge): 5ed4d5250e87177fd4c4f4e9f9238cf4

SRES (Expected Response):

8A69046163903D54366D9AF1E410B40E56872AC6

Exp10A5Algo.java :

```
import java.lang.Math;

public class A5 {
    static int[] GenerateBits() {
        int[] a = new int[16];
        for (int i = 0; i < 16; i++) {
            double rand = Math.random();
            if (rand >= 0.5) {
                a[i] = 1;
            } else {
                a[i] = 0;
            }
        }
        return a;
    }
}
```

```

static int[] XOR(int[] a, int[] b) {
    int[] temp = new int[16];
    for (int i = 0; i < 16; i++) {
        if (a[i] == 1 && b[i] == 1 || a[i] == 0 && b[i] == 0) {
            temp[i] = 0;
        } else {
            temp[i] = 1;
        }
        System.out.print(temp[i]);
    }
    return temp;
}

static int[] AND(int[] a, int[] b) {
    int[] temp = new int[16];
    for (int i = 0; i < 16; i++) {
        if (a[i] == 1 && b[i] == 1) {
            temp[i] = 1;
        } else {
            temp[i] = 0;
        }
        System.out.print(temp[i]);
    }
    return temp;
}

public static void main(String[] args) {
    int[] a;
    System.out.println("Generating the 1st key identification number");
}

```

```
a = GenerateBits();
for (int i = 0; i < 16; i++) {
    System.out.print(a[i]);
}
int[] b;
System.out.println("\n\nGenerating the 2nd key identification number");
b = GenerateBits();
for (int i = 0; i < 16; i++) {
    System.out.print(b[i]);
}
int[] c;
System.out.println("\n\nGenerating the random number");
c = GenerateBits();
for (int i = 0; i < 16; i++) {
    System.out.print(c[i]);
}
int[] d;
System.out.println("\n\nGenerating the barker code");
d = GenerateBits();
for (int i = 0; i < 16; i++) {
    System.out.print(d[i]);
}
int[] z;
System.out.println("\n\nAND of 1st key and 2nd key");
z = AND(a, b);
int[] p;
System.out.println("\n\nXOR of random number and the AND of 1st key and 2nd key");
p = XOR(z, c);
System.out.println("\n\nXOR of the above number and barker code");
p = XOR(p, d);
```

```

int[] q;
System.out.println("\n\nXOR of random number and the AND of 1st key and 2nd key");
q = XOR(z, c);
System.out.println("\n\nXOR of the above number and barker code");
q = XOR(q, d);
int flag = 0;
for (int i = 0; i < 16; i++) {
    if (p[i] != q[i]) {
        flag = 1;
        break;
    }
}
if (flag == 1) {
    System.out.print("\n\nEncryption Failed");
} else {
    System.out.print("\n\nEncryption Passed");
}
}
}

```

Output:

(base) PS C:\Users\ParthDabholkar\OneDrive\Desktop\MCC Exp Documents> cd "c:\Users\Rishab\OneDrive\Desktop\MCC Exp Documents\" ; if (\$?) { javac Exp10A5Algo.java } ; if (\$?) { java Exp10A5Algo }

Generating the 1st key identification number

0111001001011110

Generating the 2nd key identification number

1111110111000001

Generating the random number

1011000011011100

Generating the barker code

0001011100001111

AND of 1st key and 2nd key

0111000001000000

XOR of random number and the AND of 1st key and 2nd key

1100000010011100

XOR of the above number and barker code

1101011110010011

XOR of random number and the AND of 1st key and 2nd key

1100000010011100

XOR of the above number and barker code

1101011110010011

Encryption Passed

Keystream bit = $0 \wedge 0 \wedge 0 = 0$

Keystream bit = $0 \wedge 0 \wedge 0 = 0$

Keystream bit = $1 \wedge 1 \wedge 0 = 0$

Keystream bit = $0 \wedge 1 \wedge 1 = 0$

Keystream bit = $1 \wedge 1 \wedge 1 = 1$

Keystream bit = $0 \wedge 0 \wedge 1 = 1$

Exp10A8Algo.java :

```
public class A8Algorithm {  
    // Secret key (Ki)  
    private static final int[] KI = { 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1 };  
  
    // A8 Algorithm  
    public static int[] generateKeyStream(int[] rand) {  
        int[] keyStream = new int[rand.length];
```

```

// Generate key stream

for (int i = 0; i < rand.length; i++) {
    keyStream[i] = rand[i] ^ KI[i % KI.length];
}

return keyStream;
}

// Example usage

public static void main(String[] args) {
    // Example random number (RAND)
    int[] rand = { 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0 };

    // Generate key stream using A8 algorithm
    int[] keyStream = generateKeyStream(rand);

    // Print key stream
    System.out.println("Key Stream:");
    for (int keyBit : keyStream) {
        System.out.print(keyBit);
    }
}

```

Output:

```
(base) PS C:\Users\ParthDabholkar\OneDrive\Desktop\MCC Exp
Documents> cd "c:\Users\Shirish\OneDrive\Desktop\MCC Exp
Documents\" ; if (?) { javac Exp10A8Algo.java } ; if (?) { java
Exp10A8Algo }
```

Step 1: RAND = 1, KI = 0, Key Bit = RAND ^ KI = 1 ^ 0 = 1

Step 2: RAND = 0, KI = 1, Key Bit = RAND \wedge KI = 0 \wedge 1 = 1

Step 3: RAND = 1, KI = 0, Key Bit = RAND \wedge KI = 1 \wedge 0 = 1

Step 4: RAND = 0, KI = 1, Key Bit = RAND \wedge KI = 0 \wedge 1 = 1

Step 5: RAND = 1, KI = 0, Key Bit = RAND \wedge KI = 1 \wedge 0 = 1

Step 6: RAND = 0, KI = 1, Key Bit = RAND \wedge KI = 0 \wedge 1 = 1

Step 7: RAND = 1, KI = 0, Key Bit = RAND \wedge KI = 1 \wedge 0 = 1

Step 8: RAND = 0, KI = 1, Key Bit = RAND \wedge KI = 0 \wedge 1 = 1

Step 9: RAND = 1, KI = 0, Key Bit = RAND \wedge KI = 1 \wedge 0 = 1

Step 10: RAND = 0, KI = 1, Key Bit = RAND \wedge KI = 0 \wedge 1 = 1

Step 11: RAND = 1, KI = 0, Key Bit = RAND \wedge KI = 1 \wedge 0 = 1

Step 12: RAND = 0, KI = 1, Key Bit = RAND \wedge KI = 0 \wedge 1 = 1

Step 13: RAND = 1, KI = 0, Key Bit = RAND \wedge KI = 1 \wedge 0 = 1

Step 14: RAND = 0, KI = 1, Key Bit = RAND \wedge KI = 0 \wedge 1 = 1

Step 15: RAND = 1, KI = 0, Key Bit = RAND \wedge KI = 1 \wedge 0 = 1

Step 16: RAND = 0, KI = 1, Key Bit = RAND \wedge KI = 0 \wedge 1 = 1

Key Stream:

1111111111111111

(A+)

→ W

ASSIGNMENT NO: 1

- Q1] Write a short note on antenna.

An antenna is a device that converts electromagnetic radiation in space into electrical currents in conductors or vice-versa, depending on whether it is being used for receiving or for transmitting respectively.

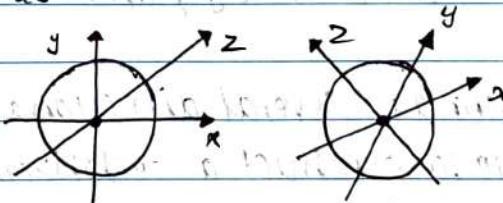
The radiation pattern of an antenna describes the relative strength of the radiated field in various directions from the antenna, at a constant distance.

In reality the radiation pattern is three-dimensional, but usually the measured radiation patterns are two-dimensional slice of the three dimensional pattern, in the horizontal or vertical planes.

The types of antennas are:

1. Isotropic Antenna: An isotropic antenna is a theoretical antenna that radiates its power uniformly in all direction.

In other words, a theoretical isotropic antenna has a perfect 360 degree spherical radiation pattern. Radiation pattern of isotropic antenna is shown as:



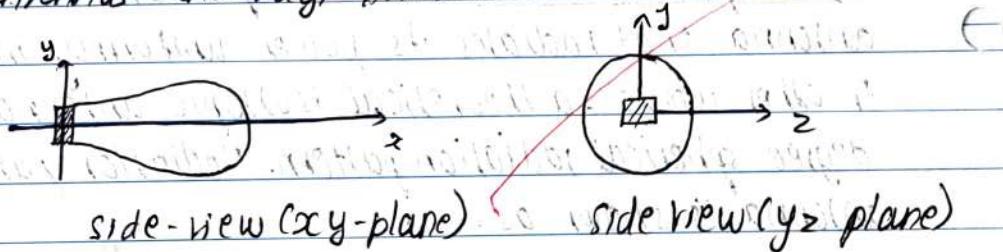
Isotropic antenna is an ideal antenna which radiates equally in all directions and has a gain of 1 (0 dB), ie 0 gain 0 loss.

It is used to compare the power level of given antenna to theoretical isotropic antenna.

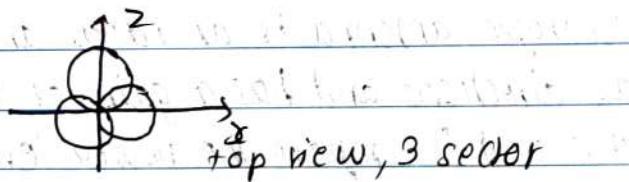
2. Omnidirectional Antennas: Unlike isotropic antennas, dipole antennas are real antennas. The dipole radiation pattern is 360 degrees in the horizontal plane and approximately 75 degrees in the vertical plane.

It is also called as "non-directional" antenna because it does not favor any particular direction. Dipole antennas are said to have a gain of 2.14 dB, which is in comparison to an isotropic antenna. The higher the gain of antennas, the smaller the vertical beam width is. This type of antenna is useful for broadcasting a signal to all points or the compass or when listening for signals from all points.

3. Directional antenna: A directional antenna or beam antenna is an antenna which radiates or receives greater power in specific directions. This allows increased performance and reduced interference from unwanted sources. Unlike omnidirectional antennas, directional antennas must be aimed in direction of transmitter or receiver. Eg: Parabolic antennas and Yagi antennas.



4. Sectorized Antenna: Several directional antennas can be combined on a single pole to construct a sectorized antenna. They are widely used in cellular telephony infrastructure.



Q2] What are various issues in signal propagation?

→ The various issues in signal propagation are listed below:

1. Free Path Loss / Free Space Loss: In free space, the signal follows a straight line. If such a straight line exists between the sender and receiver, it is called the line of sight (LOS). The signal experiences free path loss even if no objects are present between sender and transmitter because the receiver power is inversely proportional to distance. $P_r \propto 1/r^2$
2. Blocking / shadowing: The signals with higher frequency behave like a straight line. These signals are blocked by even small objects like wall, truck, car etc. This is called as blocking/shadowing.
3. Reflection: When a signal encounters a surface that is relatively to the wavelength of the signal, a phenomenon is called reflection. The reflected signal is not as strong as original, as object absorbs some of signal's power.
4. Refraction: The effect occurs because velocity of electromagnetic waves depends on the density of the medium through which it travels.
5. Scattering: If the object size is in the order of the wavelength of the signal or less, then the signal can be scattered into many small signals. Scattered signals are weaker than original signal.
6. Diffraction: Diffraction occurs at the edge of an impenetrable body that is large as compared to the wavelength of a radio wave.

(Q2) What are various issues in signal propagation?

→ The various issues in signal propagation are listed below:

1. Free Path Loss / Free Space Loss: In free space, the signal follows a straight line. If such a straight line exists between the sender and receiver, it is called the line of sight (LOS). The signal experiences free path loss even if no objects are present between sender and transmitter because the receiver power is inversely proportional to distance. $P_r \propto 1/r^2$.
2. Blocking / shadowing: The signals with higher frequency behave like a straight line. These signals are blocked by even small objects like wall, truck, car etc. This is called as blocking/shadowing.
3. Reflection: When a signal encounters a surface that is relatively to the wavelength of the signal, a phenomenon is called reflection. The reflected signal is not as strong as original, as object absorbs some of signal's power.
4. Refraction: The effect occurs because velocity of electromagnetic waves depends on the density of the medium through which it travels.
5. Scattering: If the object size is in the order of the wavelength of the signal or less, then the signal can be scattered into many small signals. Scattered signals are weaker than original signal.
6. Diffraction: Diffraction occurs at the edge of an impenetrable body that is large as compared to the wavelength of a radio wave.

Q3- Explain different applications of mobile computing.

→ Mobile Computing has various applications across different domains.

1. Communication: Mobile devices enable voice calls, text messaging, video conferencing, facilitating communication between individual and groups.

2. Internet Access: Mobile computing allows users to access the internet on the go, enabling browsing, emailing and social networking, and online shopping.

3. Navigational: GPS-enabled mobile devices provide real-time navigation services, helping users find directions, and locate nearby services and plan routes.

4. Entertainment: Mobile devices offer a wide range of entertainment options including streaming videos, music, gaming, e-books and multimedia content consumption.

5. Productivity: Mobile computing enhances productivity by providing access to productivity tools such as calendars, task managers, office suites, and collaboration platforms.

6. Healthcare: Mobile applications facilitate remote patient monitoring, telemedicine consultations, medication reminders, and health tracking, improving access to healthcare services.

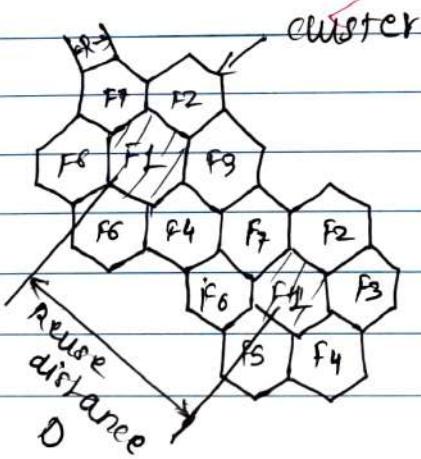
7. Education: Mobile devices support e-learning initiatives by providing access to educational resources, online courses, interactive learning materials, and educational apps.

8. Location Based services: Mobile computing enables location-based services such as geotagging, location-based advertising, location-based gaming, and location-based social networking.

Q) Explain concept of frequency reuse with clustering.

Ans- Frequency reuse is a core principle in cellular networks to optimise spectrum use. Clustering enhances this concept by grouping neighbouring cells and assigning each cluster a unique set of frequencies. This allows efficient frequency reuse within a cluster, minimizing interference while maximizing capacity.

In this approach, neighbour clusters use different frequency sets, preventing interference. Through strategic cluster design and frequency alignments, cellular networks achieve high spectral efficiency, enhancing overall performance and user experience. This balance ensures effective spectrum utilization and improved network capacity.



For hexagonal cells, the reuse distance is given by:

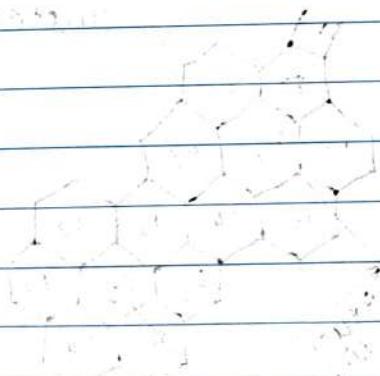
$$D = \sqrt{3}N \times R$$

where, R is cell Radius,
 N is reuse pattern (The cluster size or no. of clusters).

Reuse factor (D/R) can be thus given as:

$$D = \sqrt{3}N$$

The diagram shows a hexagonal lattice. It has 6 cells per side. The reuse pattern is 3. The reuse distance is $\sqrt{3}N \times R$. The reuse factor is $\sqrt{3}N$.



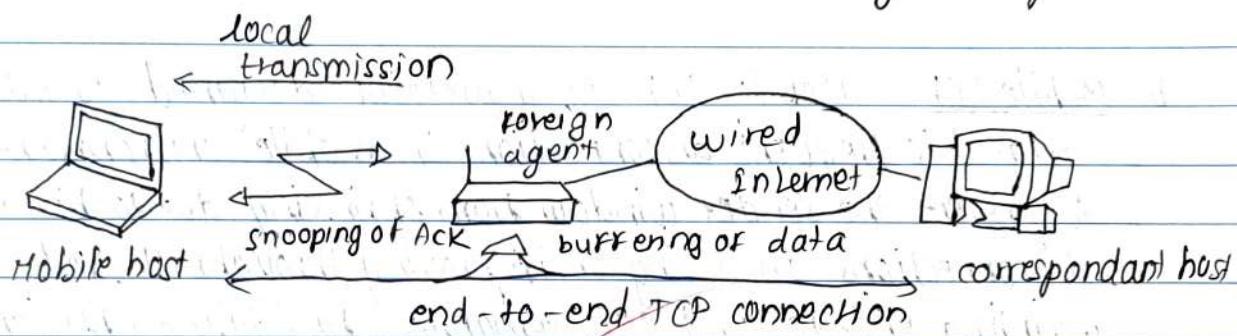
(A+)
→ H

ASSIGNMENT NO: 2

Q1] Explain in detail with merits and demerits.

a) Snooping TCP:

Snooping TCP is one of the classical TCP improvement approaches. This approach is designed to solve the end-to-end semantics loss in I-TCP. The basic concept is to buffer packets close to the mobile node and retransmit them locally if a packet is lost.



Until it receives an acknowledgement from the mobile node, the foreign agent buffers the packet. A foreign agent snoops the packet flow and acknowledgement in both directions. If foreign agent does not receive acknowledgement from mobile node, or receives duplicate acknowledgements, it believes that packet or acknowledgement has been lost. It maintains a timer for retransmission and sends NACKs to the mobile node or missing packets.

Advantages:

1. the end-to-end TCP semantic is preserved: The packet is not acknowledged by the FA. And if the Foreign agent (FA) or base station (BS) fails, the solutions revert to standard TCP.
2. No Modifications at Fixed Host: The fixed computer TCP does not need any changes.

SNR packet Disadvantages:

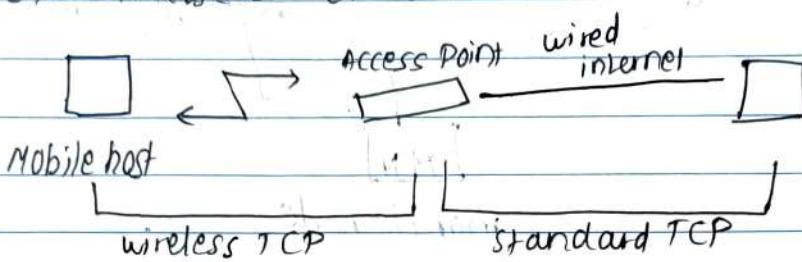
1. The behavior of wireless link: Snooping TCP does not isolate behavior of wireless link or S-TCP. Transmission errors can spread to the correspondent nodes (CH).
2. Encryption at end-to-end: If such encryption schemes are used end-to-end between the correspondent node and mobile node, snooping and buffering dat can be considered worthless.

b) Mobile TCP: Mobile TCP is a protocol designed to address issues related to frequent disconnections in mobile environments. It aims to prevent the sender window from shrinking due to bit errors or disconnections, focussing on improving throughput, reducing delay, maintaining end-to-end TCP semantics, facilitating efficient handovers.

- Merits:
- i) Maintains TCP end-to-end semantics without requiring failed packet retransmission by supervisory host.
 - ii) Does not necessitate changes in the sender's TCP, enhancing capability.
 - iii) Eliminates the need to send old buffer data to new supervisory host during handovers.

- Demerits:
- i) Requires modifications to the mobile host protocol software, adding complexity.
 - ii) Assumes lower bit rates, which may not always hold true, leading to propagated packet loss to the sender.

c) I-TCP (Indirect TCP): It splits end-to-end TCP connection between fixed host and mobile host into two separate connections at an intermediate called MSR (Mobility support Router). MSR can be an access point, foreign agent in case mobile IP, entry point of network like IWF in GSM, GGSN in case of GPRS.



Merits: 1) End to End Semantics Preservation: I-TCP maintains end-to-end TCP semantics without requiring modifications to fixed network computers.

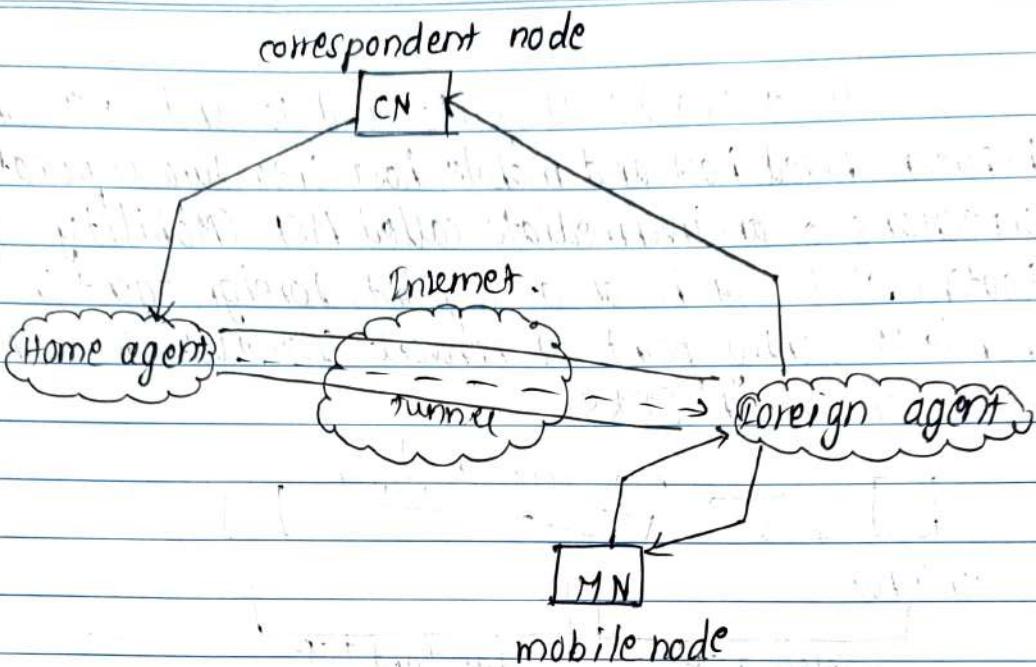
2) Compatibility: Fully compatible with TCP/IP on fixed networks, ensuring seamless integration.

Demerits: 1) Wireless Link Behavior: Does not isolate wireless link behavior, allowing transmission errors to spread to correspondent nodes.

2) Security Risk: The scheme relies on trustworthiness of the foreign agent, if compromised, serious security threats can arise.

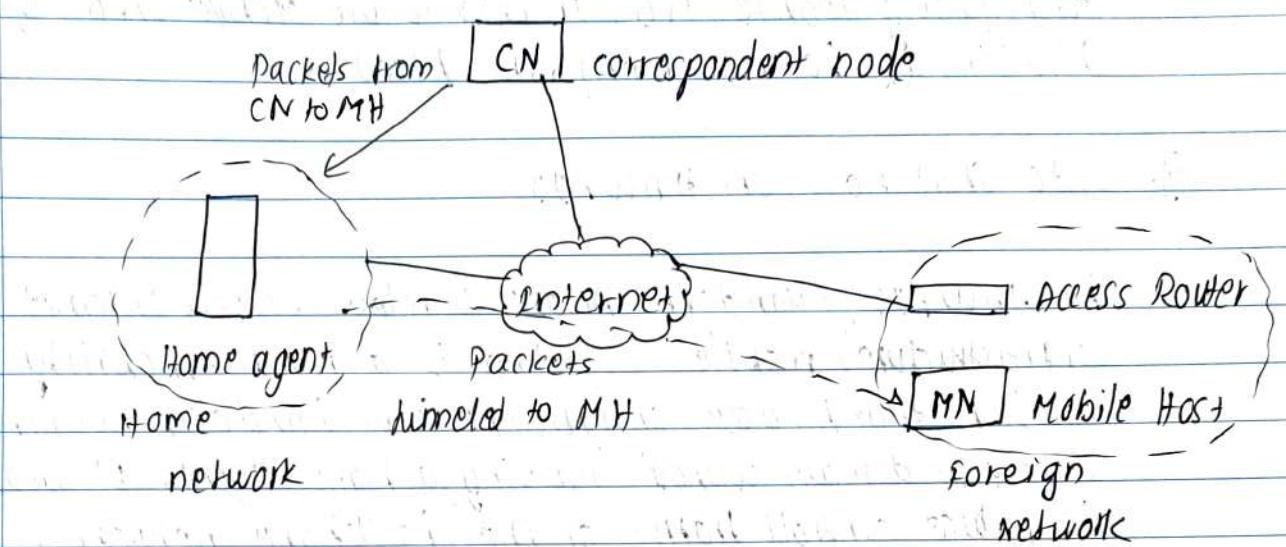
Q2] Explain Mobile (IP) cellular in detail.

→ Mobile IP is a communication protocol that allows users to move from one network to another with same IP address. It ensures that communication will continue without user's session or connections being dropped.



1. Mobile Node(MN): Handheld communication device that user carries.
2. Home Network: Network to which mobile node originally belongs as per its assigned IP address.
3. Home Agent(HA): Router in-home network to which the mobile node was originally connected.
4. Home Address: Permanent IP address assigned to mobile node.
5. Foreign Network: Current network to which the mobile node is visiting.
6. Foreign Agent(FA): Router in a Foreign network to which the mobile node is currently connected.
7. correspondent node (CN): Device on the internet communicating to the mobile node.
8. Care-of Address (COA): temporary address used by a mobile node while it is moving away from its home net.
9. Foreign Agent: COA should be located at the FA. COA is an IP address of FA.
10. co-located COA: COA is co-located if the MN temporarily acquired an additional IP address which acts as COA.

Q3- Explain MIPv6.



MIPv6 Architecture.

Mobile IPv6 is a protocol that enables a mobile node (MN) to maintain its connectivity and session continuity while moving between different networks without changing its home address. It allows MN to seamlessly roam from its home network to its foreign networks, ensuring uninterrupted communication by updating its location information with a Home Agent (HA) and correspondent nodes (CNs). MIPv6 operates at the network layer, providing transparent mobility support above the IP layer. This protocol ensures that mobile devices can stay connected and reachable even when changing their point of attachment in the network, enhancing their efficiency and reliability of mobile communication.

Mobile IPv6 security: The binding update and binding acknowledgement message exchanged between mobile node and home agent must be protected by ESP and authentication algorithm.

Mobile IPv6 configuration: This introduces information about configuring Mobile IPv6. In order to use Mobile IPv6, you must first install `bos.net-mobip6t6e` filter.

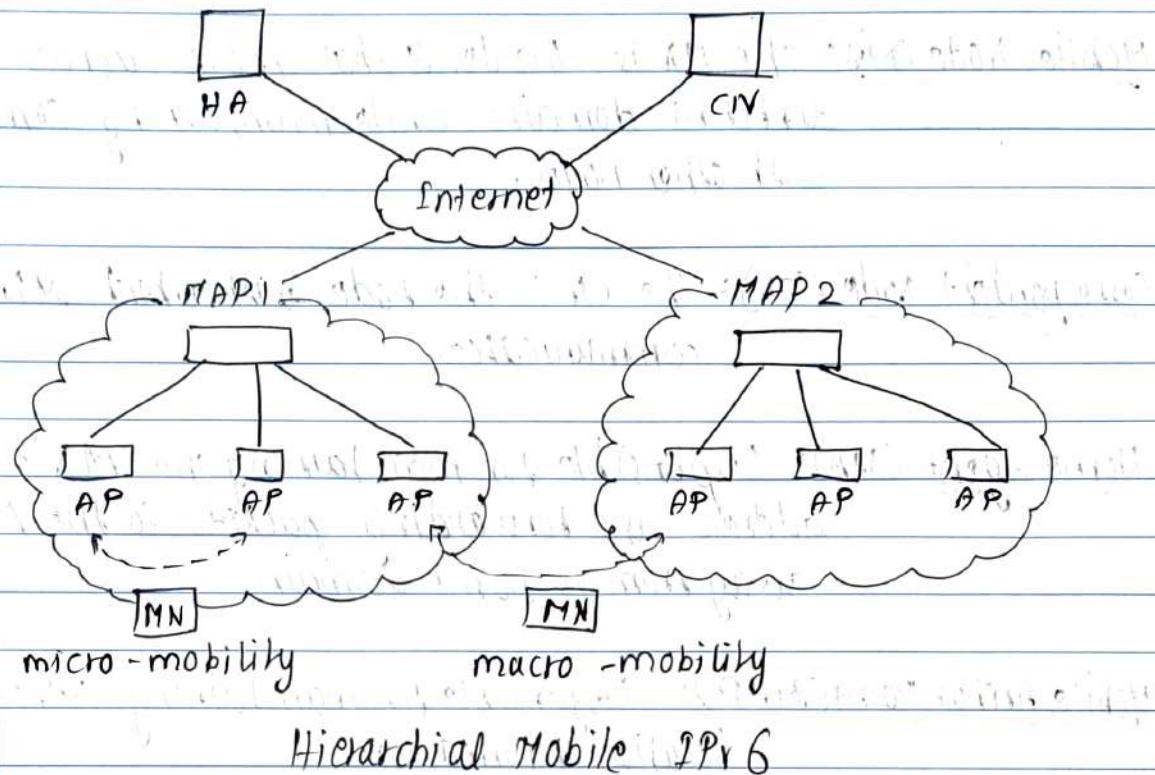
Q4) Write short notes on HAWAII.

The HAWAII (Handoff Aware Wireless Access Internet Infrastructure) mobile IP protocol is a micro-mobility solution that focuses on efficient local handover within a foreign domain without involving a home agent. It aims to reduce control traffic on the backbone network, enhance performance, provide intrinsic support for quality of service (QoS), and improve reliability in wireless communication environment. HAWAII is transparent to mobile hosts compatible with Mobile IP extensions, and it relies on Mobile IP for wide-area inter-domain macro mobility management. The protocol allows mobile nodes to use co-located care address (CCoA) and register with a base station, handling registrations locally to reduce updates to the home agent. HAWAII insures seamless handovers, maintains connectivity during mobility events, and splits the processing and generation of Mobile IP registration messages between the mobile host, base station and home agent.

Characteristics of HAWAII:

1. Reduce mobility related disruptions to user application.
2. Reduce the number of mobility related upload.
3. Simplify QoS support.
4. Improved reliability with soft-state transition.

Q5] Explain HMIPv6 in detail.



HMIPv6 suffers from several drawbacks, such as high signalling cost, latency, and packet loss. The hierarchical mobile IPr6 protocol was proposed to address these issues by introducing a hierarchical structure to M2Pr6 architecture. This structure allows for better scalability, reduced signalling overhead, and improved handover latency, making it an active solution for mobile networks.

HMIPv6 is an extension of M2Pr6 that introduces a hierarchical structure to the network architecture. In HMIPv6, the network is divided into multiple domains, with each domain having a Home Agent (HA) and Mobility Anchor Point (MAP). The HA is responsible for providing the MN's home address, while the MAP is responsible for maintaining the MN's current

location information.

Mobile Node (MN): The MN is the device that moves across different domains while maintaining communication with other nodes.

Correspondent Node (CN): The CN is the node with which MN communicates.

Home Agent (HA): Responsible for maintaining the MN's home address and forwarding packets to the MN when away from its home domain.

Mobile Anchor Point (MAP): Responsible for maintaining MN's current location information.

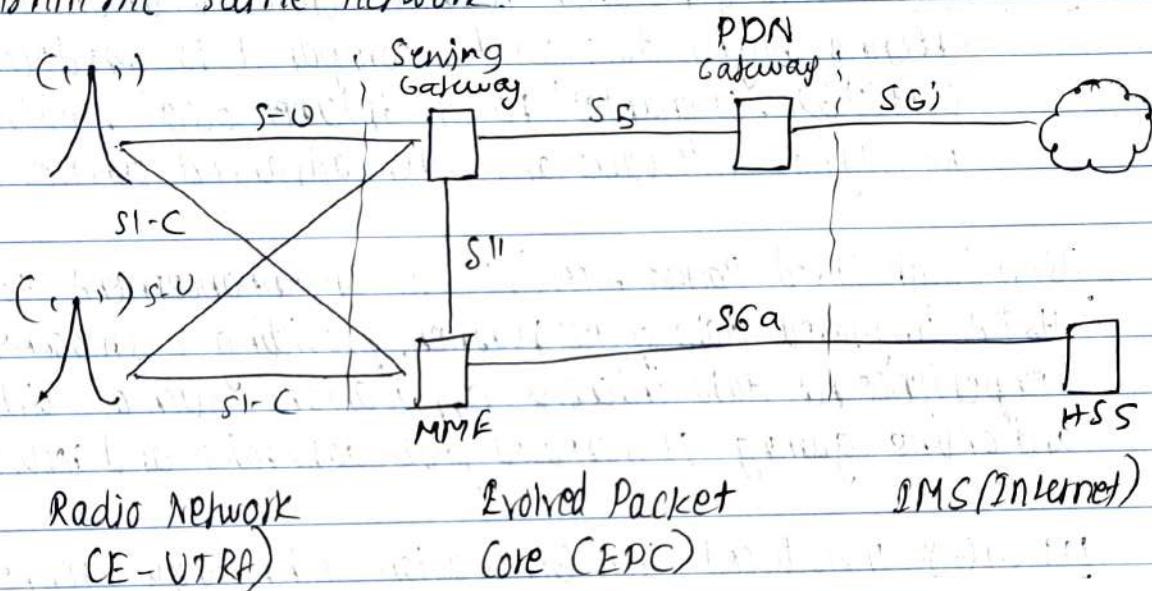
Home Network: The HN is the network in which MN has its home address.

Visited Network: The VN is the network in which the MN is currently located.

Q6] Draw and explain SAE Architecture.

System Architecture Evolution (SAE) is a new architecture designed to simplify LTE networks and establish a flat architecture similar to other IP based communication networks. SAE uses an eNB and Access Gateway (aGW) and removes the RNC and sGSN from the equivalent 3G network.

architecture to create a simple mobile network. This allows the network to be built with All-IP based network architecture. SAP also includes entities to allow full networking with other related websites (WCDMA, WiMAX, WLAN) etc. These entities can specifically manage and permit the non-3GPP technologies to interfere directly with the network and managed within the same network.



MME (Mobile Management Entity): It is responsible for:

- Idle mode UE (User Equipment) tracking
- Paging procedure such as re-transmissions.
- Bearer activation and deactivation process.
- User authentication with HSS.

PGW (PDN Gateway):

- Policy enforcement
- Packet Screening
- Packet Filtering
- Charging Support
- Lawfull Interception

Q7] Describe evolution from UMTS to LTE.

The evolution from UMTS (Universal Mobile Telecommunication Systems) to LTE (Long-Term Evolution) represents a significant advancement in mobile computing and telecommunications.

UMTS (3G): Third generation mobile communication technology offering higher data speeds compared to predecessor, GSM (2G). It enabled faster internet access, multimedia messaging, videocalls and other advanced services.

HSPA (High Speed Packet Access): As an enhancement to UMTS, HSPA improved data speeds further, providing a smoother experience for data-intensive applications such as video streaming and online gaming. It increased peak data rates and reduced latency.

LTE (4G): Introduced 4th Generation technology, represented as a significant leap in mobile communication. It offered even higher data speeds, lower latency and increased capacity compared to UMTS and HSPA.

5G: While not directly part of the UMTS to LTE evolution, it's worth mentioning 5G, the 5th generation mobile network. It introduced new features such as ultra-low latency, massive IoT connectivity and significantly faster data speeds.

Q8] Compare Mobile Generations (2G, 3G, 4G, 5G).

feature	2G (GSM)	3G (UMTS)	4G (LTE)	5G
Data Speed	Up to 384 Kbps	Up to 42 Mbps	Up to 1 Gbps	Up to 20 Gbps.
Latency	Relatively High	Reduced (100-500ms)	further reduced (10ms)	Ultra Low (1ms)
Network Architecture	Circuit Switched	Hybrid	Packet Switched	Fully Packet Switched
Spectral Efficiency	Low	Moderate	High	Very High
Multiple Access Schemes	TDMA/FDMA	CDMA	OFDMA	OFDMA and SCFDMA
Spectrum Usage	Narrow-band	Wide-band	Broad-band	Ultra-Broadband.
Applications	Voice calls, SMS	web Browsing, video calls, Multi media Messaging	HD streaming, Online Gaming	IoT, VR/AR
Technology	GSM	WCDMA/HSPA	LTE	NR (New Radio)

Q9)- what are self organizing networks?

Self-Organizing networks are automated network management systems that aim to optimize the performance and efficiency of wireless communication networks. They utilize intelligent algorithms and automation to enable self-configuration, self-optimization, and self-healing capabilities within a network.

- 1) Self Configuration: SON systems can automatically configure network elements such as base stations, antennas, and frequency without human intervention. Includes initial network setup, neighbor cell detection and parameter tuning.
- 2) Self Optimization: Continuously monitor network performance metrics such as signal strength, interference and traffic load. They dynamically optimize coverage, capacity and quality of service. Load balancing, handover optimization and interference manager.
- 3) Self Healing: They can detect and mitigate network faults and anomalies in real-time. Issues such as equipment failure or sudden changes in network conditions, can be automatically reconfigured network elements to restore and minimize downtime. This may involve routing traffic, adjusting power levels, or activating redundant resources.

Q10)- Explain VOLTE in detail.

Voice over Long-Term Evolution (VoLTE) is a technology that

enables voice calls to be transmitted over LTE (4G) networks as opposed to traditional circuit-switched networks used in 2G and 3G. VoLTE delivers voice calls as data packets over the IP network, providing several benefits compared to traditional circuit-switched calls.

Working: 1. Packet Switching:

Converts voice calls into data packets and transmit them over LTE network using packet switching technology.

2. IP Multimedia Systems: A standardized framework for delivering multimedia services over IP networks. Handles call setup, session management and session authentication.

3. Quality of Service:

Prioritizes voice packets to ensure low latency and high quality voice calls, even during periods of heavy data traffic.

4. Wideband Audio:

Supports wideband audio codecs, such as HD voice, which deliver clearer and more natural sounding voice calls compared to traditional narrowband codecs.

Advantages:

1. High quality voice calls:

Supports HD voice codecs, providing superior audio quality compared to traditional voice calls.

2. Low Latency:

Reduces call setup time and latency, resulting in faster call connection and less delay during conversations.

Demerits:

1. Device and Network Compatibility:

Requires both the device and network to support the technology. Older devices or networks may not be compatible with VoLTE.

2. Battery Consumption: Consumes more battery power than traditional voice calls, especially if the device is not optimized for VoLTE.