SRI KRISHNA INSTITUTE OF TECHNOLOGY

Bengaluru – 90

DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

6th SEMESTER
(CBCS SCHEME)

MOBILE APPLICATION DEVELOPMENT

(18CSMP68)

LAB MANUAL

MOBILE APPLICATION DEVELOPMENT
(Effective from the academic year 2018 -2019)
SEMESTER – VI

Course Code	**18CSMP68**	**IA Marks**	40
Number of Contact Hours/Week	0:0:2	**Exam Marks**	60
Total Number of Contact Hours	3 Hours/Week	**Exam Hours**	03

CREDITS – 02

Laboratory Objectives: Thislaboratory (18CSMP68) will enable students to

- Learn and acquire the art of Android Programming.
- ConfigureAndroid studio to run the applications.
- Understand and implement Android's User interface functions.
- Create, modify and query on SQlite database.
- Inspect different methods of sharing data using services.

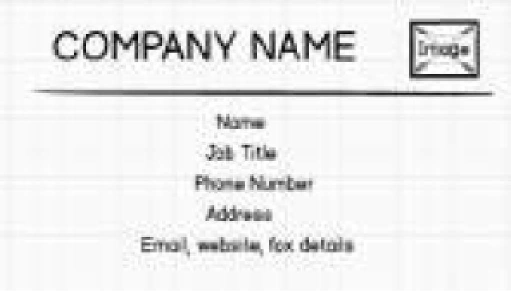
Descriptions (if any):

Installation procedure of the Android Studio/Java software must be demonstrated, carried out in groups.
Students should use the latest version of Android Studio/Java to execute these programs.
All of these diagrams are for representational purpose only.Students are expected to improvise on it.

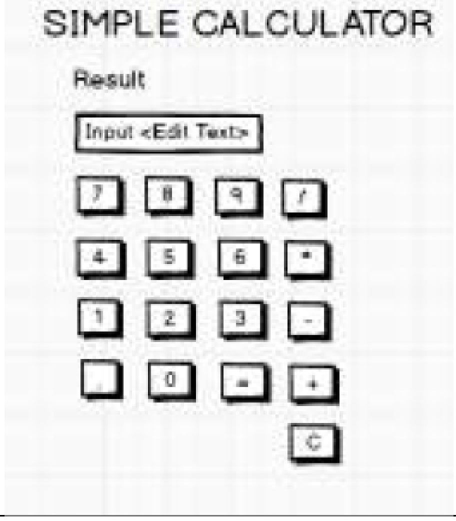
Programs List:

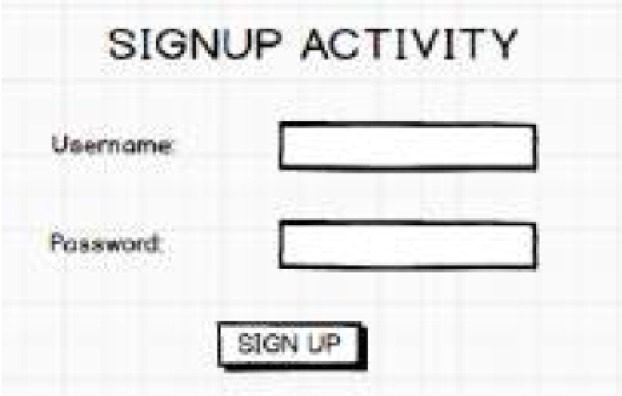
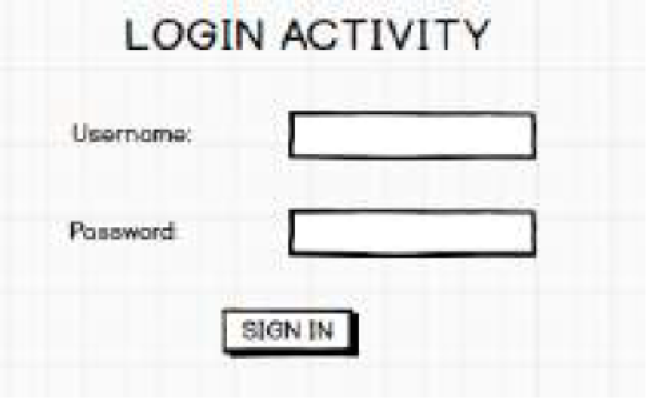
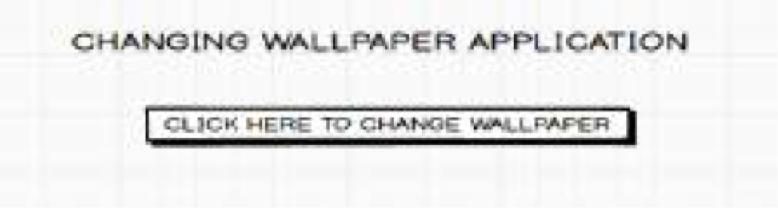
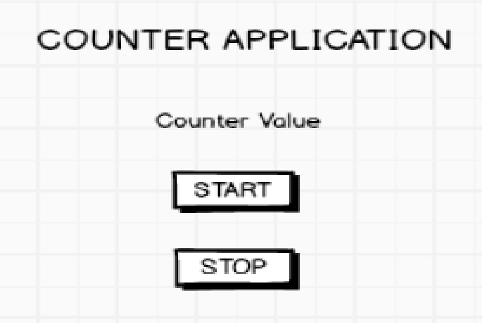
PART – A

1 Create an application to design aVisiting Card. The Visiting card should havea companylogoatthe top right corner. The company name should be displayed in Capital letters, aligned to the center. Information like the name of the employee, job title, phone number, address, email, fax and the website address isto be displayed. Insert a horizontal line between the job title and the phone number.

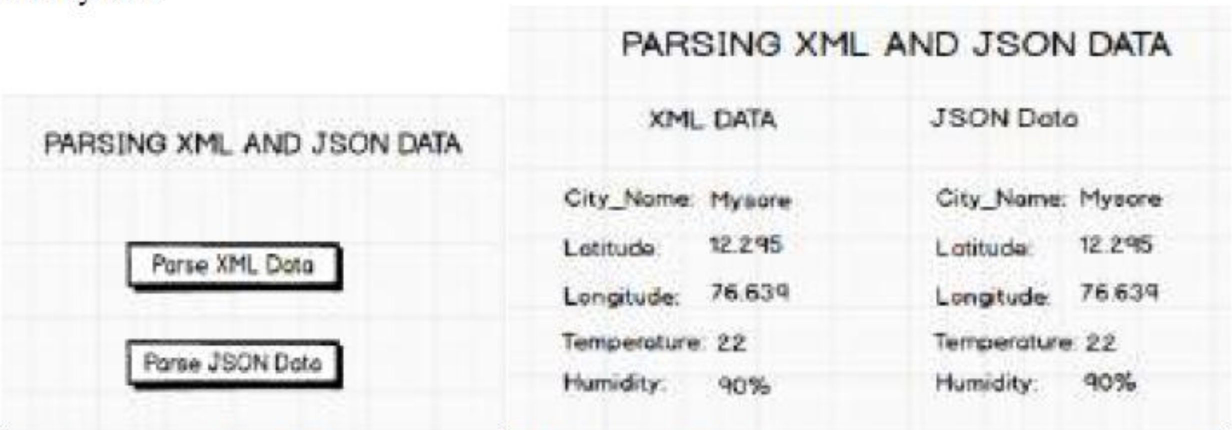


2 Develop an Android application usingcontrols like Button, TextView, EditText for designing a calculatorhaving basic functionality like Addition, Subtraction, Multiplication,andDivision.



3	Create a SIGN Up activity with Username and Password. Validation of password should happen based on the following rules: • Password should contain uppercase and lowercase letters. • Password should contain letters and numbers. • Password should contain special characters. • Minimum length of the password (the default value is 8). On successful **SIGN UP** proceed to the next Login activity. Here the user should **SIGN IN** using the Username and Password created during signup activity. If the Username and Password are matched then navigate to the next activity whichdisplays a message saying "Successful Login" or else display a toast message saying "Login Failed".The user is given only two attempts and after thatdisplay a toast message saying "Failed Login Attempts" and disable the SIGN IN button. Use Bundle to transfer information from one activity to another.  
4	Develop an application to set an image as wallpaper. On click of a button, the wallpaper image should start to change randomly every 30 seconds.  CHANGING WALLPAPER APPLICATION CLICK HERE TO CHANGE WALLPAPER
5	Write a program to create an activity with two buttons START and STOP. On pressingoftheSTART button, the activity must start the counter by displaying the numbers from One and the counter must keep on counting until the STOP button is pressed. Display the counter value in a TextViewcontrol. 

6 Create two files of XML and JSON type with values for City_Name, Latitude, Longitude, Temperature,andHumidity. Develop an application to create an activity with two buttons to parse the XML and JSON files which when clicked should display the data in their respective layouts side by side.

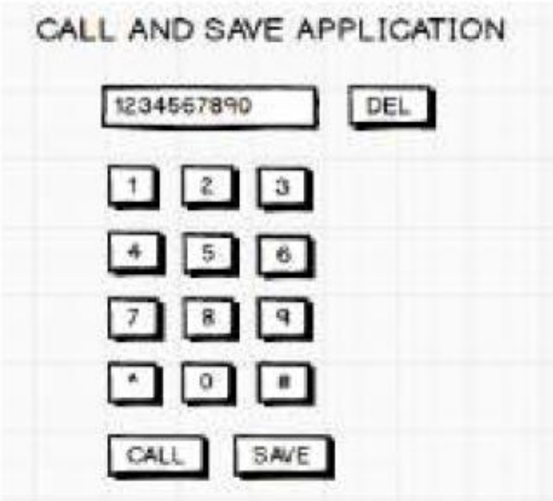


City_Name: Mysore
Latitude: 12.295
Longitude: 76.639
Temperature: 22
Humidity: 90%

City_Name: Mysore
Latitude: 12.295
Longitude: 76.639
Temperature: 22
Humidity: 90%

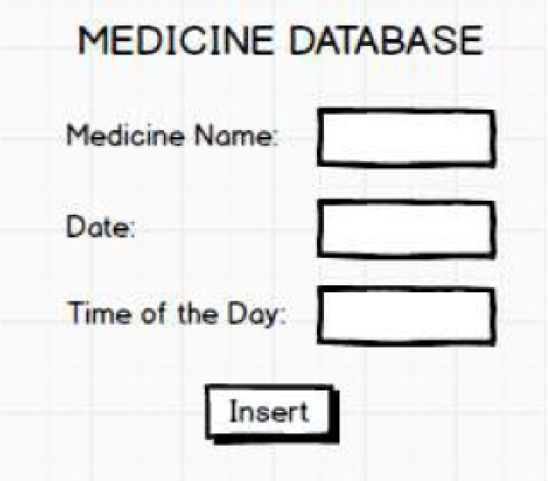
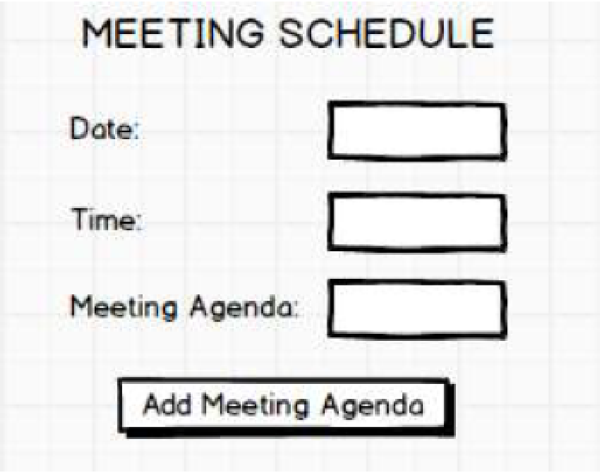
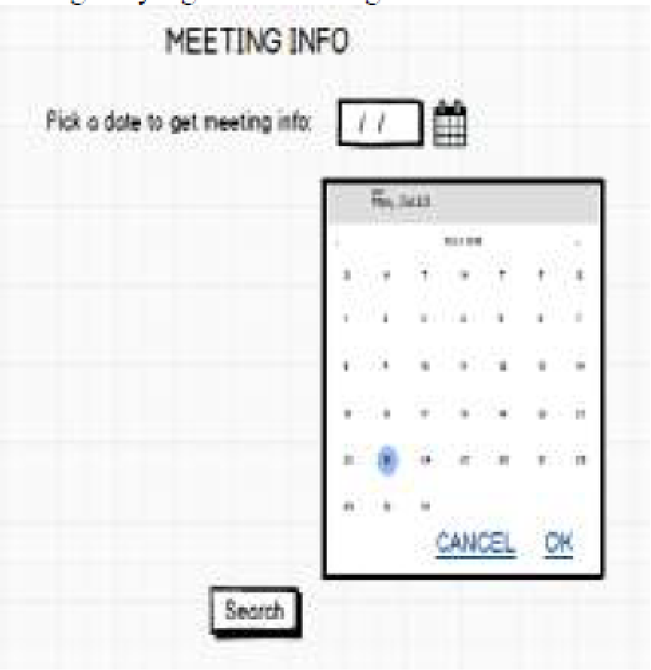
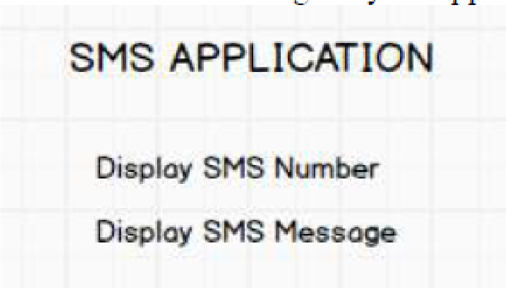
7 Develop a simple application withoneEditTextso that the user can write some text in it. Create a button called “Convert Text to Speech” that converts the user input text into voice.

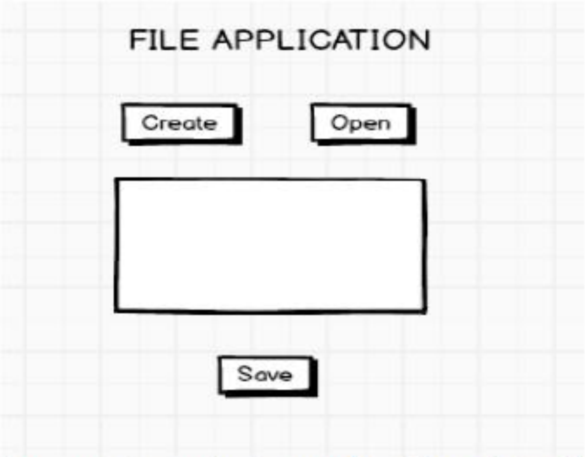
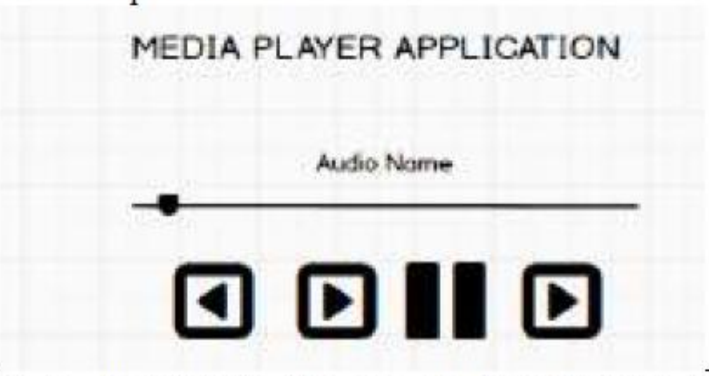
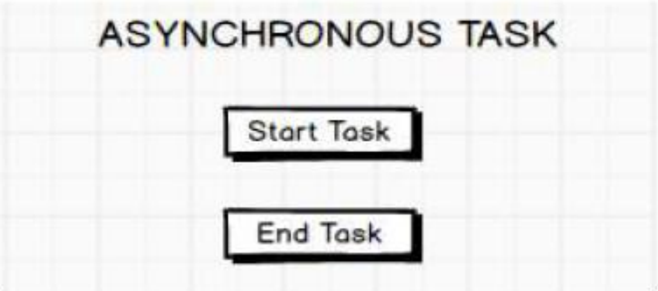
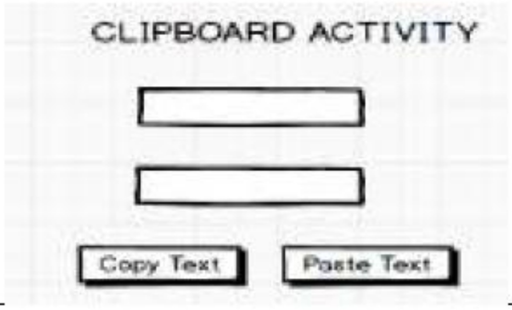
8 Create an activity like a phone dialer withCALLand SAVE buttons. On pressing the CALL button, it must call the phone number and on pressing the SAVE button it must save the number to the phone contacts.



PART - B

1 Write a program to enter Medicine Name, Date and Time of the Day as input from the user and store it in the SQLite database. Input for Time of the Day should be either Morning or Afternoon or Eveningor Night. Trigger an alarm based on the Date and Time of the Day and display the Medicine Name.

	 MEDICINE DATABASE Medicine Name: Date: Time of the Day: Insert
2	Develop a content provider application with an activity called “Meeting Schedule” which takes Date, Time and Meeting Agenda as input from the user and store this information into the SQLite database. Create another application with an activity called “Meeting Info” having DatePicker control, which on the selection of a date should display the Meeting Agenda information for that particular date, else it should display a toast message saying “No Meeting on this Date”.  MEETING SCHEDULE Date: Time: Meeting Agenda: Add Meeting Agenda  MEETING INFO Pick a date to get meeting info: / / CANCEL OK Search
3	Create an application to receive an incoming SMS which is notified to the user. On clicking this SMS notification, the message content and the number should be displayed on the screen. Use appropriate emulator control to send the SMS message to your application.  SMS APPLICATION Display SMS Number Display SMS Message
4	Write a program to create an activity having a Text box, and also Save, Open and Create buttons. The user has to write some text in the Text box. On pressing the Create button the text should be saved as a text file in MkSDcard. On subsequent changes to the text, the Save button should be pressed to store the latest content to the same file. On pressing the Open button, it should display

	the contents from the previously stored files in the Text box. If the user tries to save the contents in the Textbox to a file without creating it, then a toast message has to be displayed saying "First Create a File". 
5	Create an application to demonstrate a basic media playerthat allows the user to Forward, Backward, Play and Pause an audio. Also, make use of the indicator in the seek bar to move the audio forward or backward as required. 
6	Develop an application to demonstrate the use of Asynchronous tasks in android. The asynchronous task should implement the functionality of a simple moving banner. On pressing the **Start Task** button, the banner message should scrollfrom right to left. On pressing the **Stop Task** button, the banner message should stop.Let the banner message be "Demonstration of Asynchronous Task". 
7	Develop an application that makes use of the clipboard framework for copying and pasting of the text. The activity consists of two EditText controls and two Buttons to trigger the copy and paste functionality. 

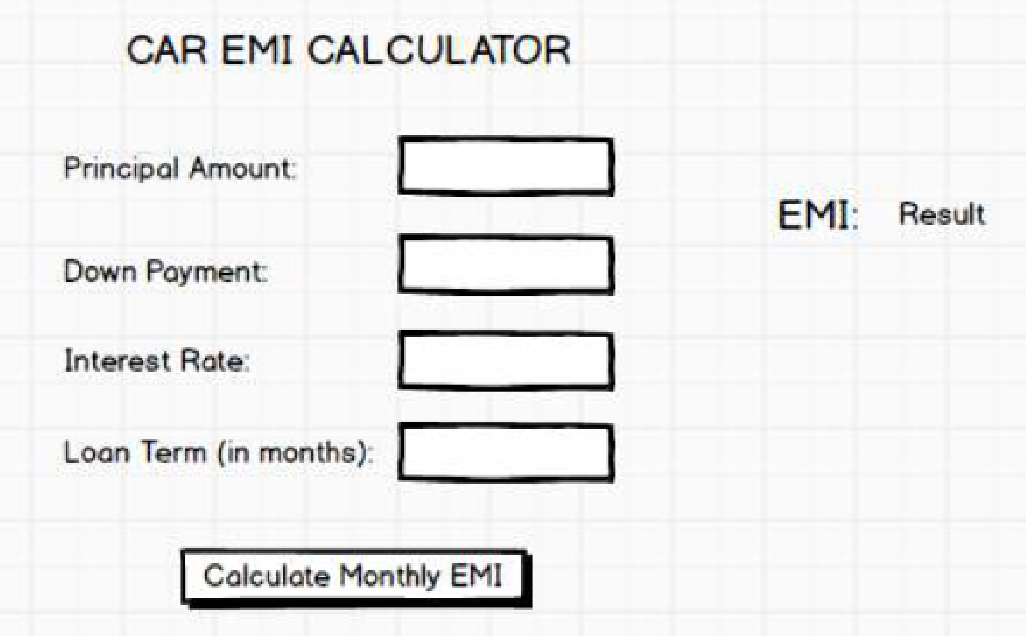
8 Create an AIDL service that calculates Car Loan EMI. The formula to calculate EMI is

$$\mathbf{E = P * (r(1+r)^n)/((1+r)^n-1)}$$

where

E = The EMI payable on the car loan amount
P = The Car loan Principal Amount
r = The interest rate value computed on a monthly basis
n = The loan tenure in the form of months

The down payment amount has to be deducted from the principal amount paid towards buying the Car. Develop an application that makes use of this AIDL service to calculate the EMI. This application should have four EditText to read the PrincipalAmount, Down Payment, Interest Rate, Loan Term (in months) and a button named as "Calculate Monthly EMI". On click of this button, the result should be shown in a TextView. Also, calculate the EMI by varying the Loan Term and Interest Rate values.



Laboratory Outcomes: After studying theselaboratory programs, students will be able to

- Create, test and debug Android application by setting up Android development environment.
- Implement adaptive, responsive user interfaces that work across a wide range of devices.
- Infer long running tasks and background work in Android applications.
- Demonstrate methods in storing, sharing and retrieving data in Android applications.
- Infer the role of permissions and security for Android applications.

Procedure to Conduct Practical Examination

- Experiment distribution
  - For laboratories having only one part: Students are allowed to pick oneexperiment from the lot with equal opportunity.
  - For laboratories having PART A and PART B: Students are allowed to pick oneexperiment from PART A and one experiment from PART B, with equalopportunity.

- Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only.
- Marks Distribution (Courseed to change in accordance with university regulations)
  - For laboratories having only one part – Procedure + Execution + Viva-Voce: 15+70+15= 100 Marks
  - For laboratories having PART A and PART B

i. Part A – Procedure + Execution + Viva = 6 + 28 + 6 = 40 Marks
ii. Part B – Procedure + Execution + Viva = 9 + 42 + 9 = 60 Marks

Text Books:

1. Google Developer Training, **"Android Developer Fundamentals Course – Concept Reference",** Google Developer Training Team, 2017. https://www.gitbook.com/book/google-developer-training/android-developer-fundamentals-course-concepts/details
(Download pdf file from the above link)

Reference Books:

1. Erik Hellman, **"Android Programming – Pushing the Limits",** 1st Edition, Wiley India Pvt Ltd, 2014. ISBN-13: 978-8126547197
2. Dawn Griffiths and David Griffiths, **"Head First Android Development",** 1st Edition, O'Reilly SPD Publishers, 2015. ISBN-13: 978-9352131341
3. Bill Phillips, Chris Stewart and Kristin Marsicano, **"Android Programming: The Big Nerd Ranch Guide",** 3rd Edition, Big Nerd Ranch Guides, 2017. ISBN-13: 978-0134706054

Table of content

Program-1: Create an application to design a Visiting Card. The Visiting card should have a company logo at the top right corner. The company name should be displayed in Capital letters, aligned to the center. Information like the name of the employee, job title, phone number, address, email, fax and the website address is to be displayed. Insert a horizontal line between the job title and the phone number.

XML-CODE

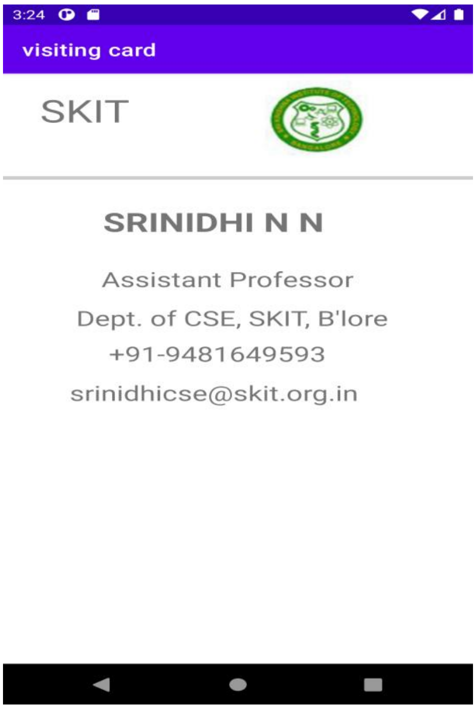
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"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

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginStart="17dp"
        android:layout_marginLeft="17dp"
        android:layout_marginTop="17dp"
        android:layout_marginEnd="244dp"
        android:layout_marginRight="244dp"
        android:layout_marginBottom="486dp"
        android:text="SKIT"
        android:textSize="38dp" />

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="231dp"
        android:layout_height="174dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="-14dp"
        android:layout_marginRight="-14dp"
        android:layout_marginBottom="481dp"
        app:srcCompat="@drawable/logo" />

    <View
        android:id="@+id/view"
        android:layout_width="wrap_content"
        android:layout_height="4dp"
        android:layout_alignParentBottom="true"
        android:background="#4444"
        android:layout_marginBottom="466dp" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="117dp"
        android:layout_marginRight="117dp"
        android:layout_marginBottom="394dp"
        android:text="Srinidhi N N"
        android:textSize="30dp"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="64dp"
        android:layout_marginRight="64dp"
        android:layout_marginBottom="343dp"
        android:text="Assistant Professor"
        android:textSize="25dp" />

        <TextView
                android:id="@+id/textView4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="71dp"
        android:layout_marginBottom="382dp"
        android:text="Dept. of CSE, SKIT,
        B'lore"
        android:textAlignment="center"
        android:textSize="24sp" />
    <TextView
```

```
        android:id="@+id/textView5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="127dp"
        android:layout_marginRight="127dp"
        android:layout_marginBottom="294dp"
        android:text="Ph No: 9481649593"
        android:textSize="20dp" />



    <TextView
        android:id="@+id/textView6"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="44dp"
        android:layout_marginRight="44dp"
        android:layout_marginBottom="189dp"
        android:text="Email: srinidhicse@skit.org.in"
        android:textSize="20dp" />
</RelativeLayout>
```

JAVA-CODE(No Change Required)

```
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

OUTPUT:



SKIT

SRINIDHI N N

Assistant Professor

Dept. of CSE, SKIT, B'lore

+91-9481649593

srinidhicse@skit.org.in

Program-2: Develop an Android application using controls like Button, TextView, EditText for designing a calculator having basic functionality like Addition, Subtraction, Multiplication, and Division.

XML-CODE:

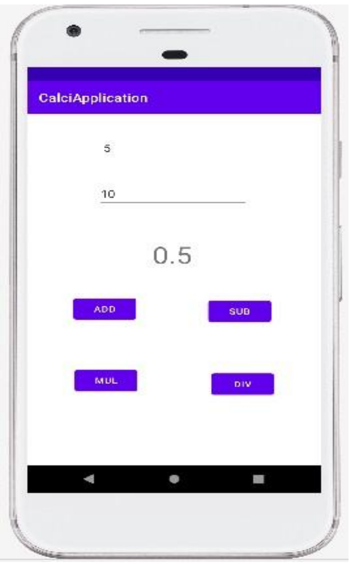
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".MainActivity">
   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentEnd="true"
      android:layout_alignParentBottom="true"
      android:layout_marginEnd="98dp"
      android:layout_marginBottom="653dp"
      android:text="SIMPLE CALCI"
      android:textSize="38dp"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintHorizontal_bias="0.498"
      app:layout_constraintLeft_toLeftOf="parent"
      app:layout_constraintRight_toRightOf="parent"
      app:layout_constraintTop_toTopOf="parent"
      app:layout_constraintVertical_bias="0.042" />
   <EditText
      android:id="@+id/editText1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentEnd="true"
      android:layout_alignParentBottom="true"
      android:layout_marginEnd="115dp"
      android:layout_marginBottom="547dp"
      android:ems="10"
      android:hint="Enter the First Number"
      android:inputType="textPersonName"
      android:text="" />
   <EditText
```

```
        android:id="@+id/editText2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="111dp"
        android:layout_marginBottom="455dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:hint="Enter the Second Number"
        android:text="" />
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="203dp"
        android:layout_marginBottom="350dp"
        android:text="0"
        android:textSize="40dp" />
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="274dp"
        android:layout_marginBottom="237dp"
        android:onClick="doAdd"
        android:text="ADD" />
    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="68dp"
        android:layout_marginBottom="233dp"
        android:onClick="doSub"
        android:text="SUB" />
    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="277dp"
        android:layout_marginBottom="115dp"
        android:onClick="doMul"
        android:text="MUL" />
    <Button
        android:id="@+id/button4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="63dp"
        android:layout_marginBottom="104dp"
        android:onClick="doDiv"
        android:text="DIV" />

</RelativeLayout>
```

JAVA-CODE:

```
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
public class MainActivity extends AppCompatActivity {
    EditText e1,e2;
    TextView tv1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        e1 =  (EditText)findViewById(R.id.editText1);
        e2 = (EditText)findViewById(R.id.editText2);
        tv1 = (TextView)findViewById(R.id.textView1);
    }
    public void doAdd(View V){
        int a1 = Integer.parseInt(e1.getText().toString());
        int a2 = Integer.parseInt(e2.getText().toString());
        int result= a1+a2;
        tv1.setText(""+result);
    }
    public void doSub(View V){
        int a1 = Integer.parseInt(e1.getText().toString());
        int a2 = Integer.parseInt(e2.getText().toString());
        int result= a1-a2;
        tv1.setText(""+result);
    }
    public void doMul(View V){
        int a1 = Integer.parseInt(e1.getText().toString());
        int a2 = Integer.parseInt(e2.getText().toString());
        int result= a1*a2;
        tv1.setText(""+result);
    }
    public void doDiv(View V){
        int a1 = Integer.parseInt(e1.getText().toString());
        int a2 = Integer.parseInt(e2.getText().toString());
        float result= a1/a2;
        tv1.setText(""+result);
    }
}
```

OUTPUT:



Program - 3

Create a SIGN Up activity with Username and Password. Validation of password should happen based on the following rules:

- Password should contain uppercase and lowercase letters.

- Password should contain letters and numbers.

- Password should contain special characters.

- Minimum length of the password (the default value is 8).

On successful SIGN UP proceed to the next Login activity. Here the user should SIGN IN using the Username and Password created during signup activity. If the Username and Password are matched then navigate to the next activity which displays a message saying “Successful Login” or else display a toast message saying “Login Failed”. The user is given only two attempts and after that display a toast message saying “Failed Login Attempts” and disable the SIGN IN button. Use Bundle to transfer information from one activity to another.

activity_signup.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout android="http://schemas.android.com/apk/res/android"
app="http://schemas.android.com/apk/res-auto"
tools="http://schemas.android.com/tools"
layout_width="match_parent"
layout_height="match_parent"
orientation="vertical"
context=".SignUpActivity">
<TextView
layout_width="match_parent"
layout_height="wrap_content"
textSize="48sp"
textAlignment="center"
text="Sign Up" />
```

```
<EditText
id="@+id/emailEditText"
layout_width="match_parent"
layout_height="wrap_content"
xmlns:layout_margin="4dp"
textSize="24sp"
hint="Email ID"
/>
<EditText
id="@+id/passwordEditText"
layout_width="match_parent"
layout_height="wrap_content"
layout_margin="4dp"
layout_marginTop="32dp"
textSize="24sp"
inputType="textPassword"
hint="Password"
/>
<Button
id="@+id/signUpBtn"
layout_width="match_parent"
layout_height="wrap_content"
layout_margin="4dp"
text="Sign Up"
/>
</LinearLayout>
```

SignUpActivity.java

```
import .AppCompatActivity;
import .Intent;
import .Bundle;
import .View;
import .Button;
import .EditText;
import .Toast;
import .Pattern;
public class SignUpActivity extends AppCompatActivity {
EditText emailEditText, passwordEditText;
Button signUpBtn;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_signup);
emailEditText = findViewById(R.id.emailEditText);
passwordEditText = findViewById(R.id.passwordEditText);
signUpBtn = findViewById(R.id.signUpBtn);
signUpBtn.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
String email = emailEditText.getText().toString();
String password = passwordEditText.getText().toString();
if (!isValidPassword(password)) {
Toast.makeText(SignUpActivity.this, "Password doesn't match rules"
, Toast.LENGTH_SHORT).show();
return;
}
Intent intent = new Intent(SignUpActivity.this, LoginActivity.class);
intent.putExtra("email", email);
intent.putExtra("password", password);
startActivity(intent);
}
});
}
Pattern lowerCase = Pattern.compile("^.*[a-z].*$");
Pattern upperCase = Pattern.compile("^.*[A-Z].*$");
Pattern number = Pattern.compile("^.*[0-9].*$");
Pattern specialCharacter = Pattern.compile("^.*[^a-zA-Z0-9].*$");
private Boolean isValidPassword(String password) {
// Checks if password length is less than 8
if (password.length() < 8) {
return false;
}
// Returns false if password doesn't contain a lower case character
if (!lowerCase.matcher(password).matches()) {
return false;
}
// Returns false if password doesn't contain an upper case character
if (!upperCase.matcher(password).matches()) {
return false;
}
// Returns false if password doesn't contain a number
if (!number.matcher(password).matches()) {
```

```
return false;
}
// Returns false if password doesn't contain a special character
if (!specialCharacter.matcher(password).matches()) {
return false;
}
return true;
}
}
```

activity_login.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout android="http://schemas.android.com/apk/res/android"
app="http://schemas.android.com/apk/res-auto"
tools="http://schemas.android.com/tools"
layout_width="match_parent"
android:layout_height="match_parent"
orientation="vertical"
context=".SignUpActivity">
<TextView
layout_width="match_parent"
layout_height="wrap_content"
textSize="48sp"
textAlignment="center"
text="Login" />
<EditText
id="@+id/emailEditText"
layout_width="match_parent"
```

```
layout_height="wrap_content"
layout_margin="4dp"
textSize="24sp"
hint="Email ID"
/>
<EditText
id="@+id/passwordEditText"
layout_width="match_parent"
layout_height="wrap_content"
layout_margin="4dp"
layout_marginTop="32dp"
textSize="24sp"
inputType="textPassword"
hint="Password"
/>
<Button
id="@+id/loginBtn"
layout_width="match_parent"
layout_height="wrap_content"
layout_margin="4dp"
text="Login"
/>
</LinearLayout
```

LoginActivity.java

```
import .AppCompatActivity;
import .Intent;
import .Bundle;
import .View;
import .Button;
import .EditText;
import .Toast;
public class LoginActivity extends AppCompatActivity {
EditText emailEditText, passwordEditText;
Button loginBtn;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_login);
emailEditText = findViewById(R.id.emailEditText);
passwordEditText = findViewById(R.id.passwordEditText);
loginBtn = findViewById(R.id.loginBtn);
String registeredEmail = getIntent().getStringExtra("email");
String registeredPassword = getIntent().getStringExtra("password");
loginBtn.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
String email = emailEditText.getText().toString();
String password = passwordEditText.getText().toString();
if (registeredEmail.equals(email) && registeredPassword.equals(passwor
d)) {
Intent intent = new Intent(LoginActivity.this, LoginSuccessActivit
y.class);
startActivity(intent);
} else {
```

```
Toast.makeText(LoginActivity.this, "Invalid Credentials", Toast.LE
NGTH_SHORT).show();
}
}
});
}
}
```

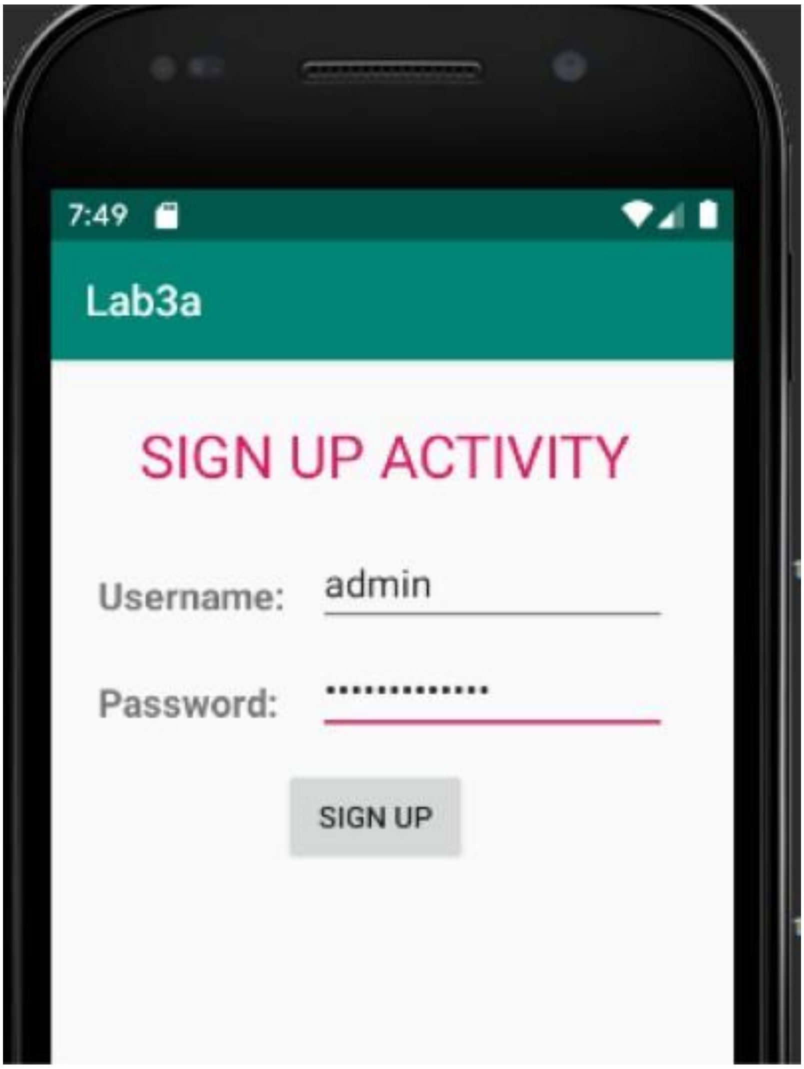
activity_login_success.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout android="http://schemas.android.com/apk/res/android"
app="http://schemas.android.com/apk/res-auto"
tools="http://schemas.android.com/tools"
layout_width="match_parent"
layout_height="match_parent"
orientation="vertical"
context=".LoginSuccessActivity">
<TextView
layout_width="match_parent"
layout_height="wrap_content"
text="Login Successful"
textAlignment="center"
textSize="36sp"/>
</LinearLayout>
```

LoginSuccessActivity.java

```
import .AppCompatActivity;
import .Bundle;
public class LoginSuccessActivity extends AppCompatActivity {
```

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



SIGN UP ACTIVITY

Username: admin

Password: ••••••••••••••

4. Develop an application to set an image as wallpaper. On click of a button, the wallpaper image should start to change randomly every 30 seconds.

First, create the android application as discussed in “Create your First Android Application”. Copy the images and save the images in the drawable folder. Following is the content of the modified res/layout/activity_main.xml.

activity_main.xml

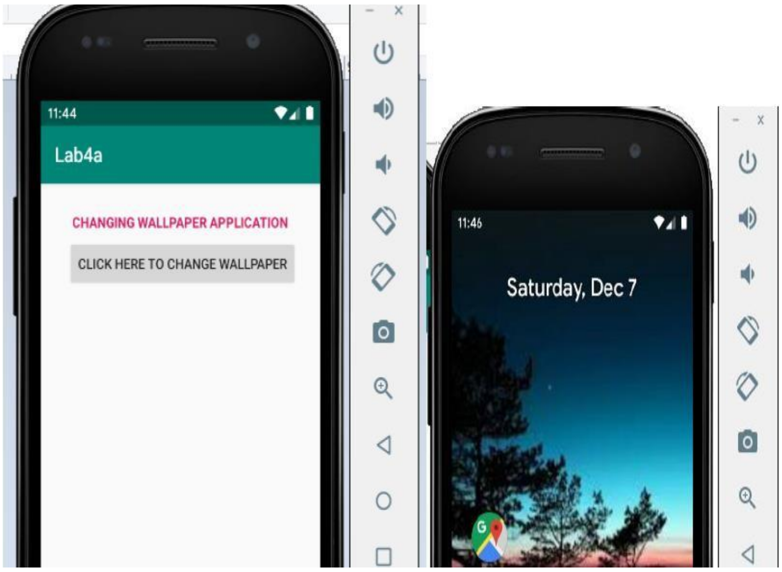
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent" android:layout_height="match_parent"
tools:context=".MainActivity">

<TextView android:layout_width="wrap_content"
android:layout_height="wrap_content"

android:text="CHANGING WALLPAPER APPLICATION"
android:textColor="@color/colorAccent"

android:textStyle="bold" app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintHorizontal_bias="0.496"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.063" />

<Button android:id="@+id/button"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_marginStart="72dp"

android:layout_marginTop="53dp"

android:layout_marginEnd="35dp"

android:layout_marginBottom="590dp"
```

```
android:text="CLICK HERE TO CHANGE WALLPAPER"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.820"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.0" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Save five images (jpg format) in the drawable folder. In this example one.jpg, two.jpg,three.jpg, four.jpg and five.jpg images are saved in drawable folder.

```
MainActivity.java package com.example.lab4a;

import androidx.appcompat.app.AppCompatActivity;

import android.app.WallpaperManager;

import android.graphics.Bitmap;

import android.graphics.BitmapFactory;

import android.graphics.drawable.AnimationDrawable;

import android.graphics.drawable.BitmapDrawable;

import android.graphics.drawable.Drawable;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.Toast;

import java.io.IOException;

import java.util.Timer;

import java.util.TimerTask;

public class MainActivity extends AppCompatActivity {

Button changewallpaper;

Timer mytimer;

Drawable drawable;

WallpaperManager wpm;
```

```
int prev=1;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
mytimer = new Timer();
wpm = WallpaperManager.getInstance(this);
changewallpaper = findViewById(R.id.button);
changewallpaper.setOnClickListener(new View.OnClickListener() {
@Override public void onClick(View view) {
setWallpaper();
}
});
}
private void setWallpaper() {
mytimer.schedule(new TimerTask() {
@Override
public void run() {
if(prev==1) {
drawable = getResources().getDrawable(R.drawable.one);
prev = 2;
}
 else if(prev==2) {
drawable = getResources().getDrawable(R.drawable.two);
prev=3;
}
else if(prev==3) {
drawable = getResources().getDrawable(R.drawable.three);
prev=4;
}
else if(prev==4) {
drawable = getResources().getDrawable(R.drawable.four);
prev=5;
}
else if(prev==5) {
drawable = getResources().getDrawable(R.drawable.five);
prev=1;
}
Bitmap wallpaper = ((BitmapDrawable)drawable).getBitmap();
try {
wpm.setBitmap(wallpaper);
} catch (IOException e) {
e.printStackTrace();
}
}
},0,30000); } }
```


11:44
Lab4a
CHANGING WALLPAPER APPLICATION
CLICK HERE TO CHANGE WALLPAPER
11:46
Saturday, Dec 7

5. Write a program to create an activity with two buttons START and STOP. On pressing of the START button, the activity must start the counter by displaying the numbers from One and the counter must keep on counting until the STOP button is pressed. Display the counter value in a TextView control.

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="COUNTER APPLICATION"
android:textColor="@color/design_default_color_primary_dark"
android:textSize="18sp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.498"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.071" />
<TextView
```

```
android:id="@+id/textView1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="60dp"
android:layout_marginTop="90dp"
android:layout_marginEnd="79dp"
android:layout_marginBottom="596dp"
android:text="Counter Value"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.498"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.071" />
<Button
android:id="@+id/btn_start"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="152dp"
android:layout_marginTop="129dp"
android:layout_marginEnd="171dp"
android:layout_marginBottom="542dp"
android:text="START"
app:layout_constraintHorizontal_bias="0.498"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
```

```
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.071"/>
<Button
android:id="@+id/btn_stop"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="152dp"
android:layout_marginTop="191dp"
android:layout_marginEnd="171dp"
android:layout_marginBottom="542dp"
android:text="STOP"
app:layout_constraintHorizontal_bias="0.498"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.071"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.a5a;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
public class MainActivity extends AppCompatActivity {
```

```
Button btnstart, btnstop;
TextView txtcounter;
int i = 1;
Handler customHandler = new Handler();
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
txtcounter = findViewById(R.id.textView1);
btnstart = findViewById(R.id.btn_start);
btnstop = findViewById(R.id.btn_stop);
btnstart.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
customHandler.postDelayed(updateTimerThread,0);
}
});
btnstop.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
customHandler.removeCallbacks(updateTimerThread);
}
});
}
private final Runnable updateTimerThread = new Runnable() {
@Override
public void run() {
```

```
txtcounter.setText(""+i);
customHandler.postDelayed(this,1000);
i++;
}
};
}
```

Program 6: Develop an application to demonstrate the use of Asynchronous tasks in android. The asynchronous task should implement the functionality of a simple moving banner. On pressing the Start Task button, the banner message should scroll from right to left. On pressing the Stop Task button, the banner message should stop. Let the banner message be "Demonstration of Asynchronous Task".

JAVA CODE:

```
import android.os.Bundle;
import android.util.Log;
import android.util.Xml;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;
import org.xmlpull.v1.XmlPullParser;
import org.xmlpull.v1.XmlPullParserException;
import java.io.IOException;
import java.io.InputStream;
import java.nio.charset.StandardCharsets;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;

public class MainActivity extends AppCompatActivity {

    Button parseXmlBtn, parseJsonBtn;
    TextView displayTextView;

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

        parseJsonBtn = findViewById(R.id.parseJsonBtn);
        parseXmlBtn = findViewById(R.id.parseXmlBtn);

        displayTextView = findViewById(R.id.displayTextView);
    parseXmlBtn.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        try {
          InputStream is = getAssets().open("city.xml");

          DocumentBuilderFactory documentBuilderFactory =
DocumentBuilderFactory.newInstance();
          DocumentBuilder documentBuilder =
documentBuilderFactory.newDocumentBuilder();
          Document document = documentBuilder.parse(is);

          StringBuilder stringBuilder = new StringBuilder();
          stringBuilder.append("XML Data");
          stringBuilder.append("\n--------- ");

          NodeList nodeList = document.getElementsByTagName("place");

          for (int i = 0; i < nodeList.getLength(); i++) {
            Node node = nodeList.item(i);
            if (node.getNodeType() == Node.ELEMENT_NODE) {
              Element element = (Element) node;

              stringBuilder.append("\nName: ").append(getValue("name", element));
              stringBuilder.append("\nLatitude: ").append(getValue("lat", element));
              stringBuilder.append("\nLongitude: ").append(getValue("long", element));
              stringBuilder.append("\nTemperature: ").append(getValue("temperature",
element));
              stringBuilder.append("\nHumidity: ").append(getValue("humidity", element));
              stringBuilder.append("\n ---------");
            }
          }
          displayTextView.setText(stringBuilder.toString());

        } catch (Exception e) {
          e.printStackTrace();
          Toast.makeText(MainActivity.this, "Error Parsing XML",
Toast.LENGTH_SHORT).show();
        }
      }
    });

    parseJsonBtn.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        String json;
        StringBuilder stringBuilder = new StringBuilder();
        try {
          InputStream is = getAssets().open("city.json");
```

```
            int size = is.available();

            byte[] buffer = new byte[size];

            is.read(buffer);

            json = new String(buffer, StandardCharsets.UTF_8);

            JSONArray jsonArray = new JSONArray(json);
            stringBuilder.append("JSON Data");
            stringBuilder.append("\n--------- ");
            for (int i = 0; i < jsonArray.length(); i++) {
              JSONObject jsonObject = jsonArray.getJSONObject(i);
              stringBuilder.append("\nName: ").append(jsonObject.getString("name"));
              stringBuilder.append("\nLatitude: ").append(jsonObject.getString("lat"));
              stringBuilder.append("\nLongitude: ").append(jsonObject.getString("long"));
              stringBuilder.append("\nTemperature:
").append(jsonObject.getString("temperature"));
              stringBuilder.append("\nHumidity: ").append(jsonObject.getString("humidity"));
              stringBuilder.append("\n --------- ");
            }
            displayTextView.setText(stringBuilder.toString());

            is.close();
          } catch (IOException | JSONException e) {
            e.printStackTrace();
            Toast.makeText(MainActivity.this, "Error in parsing JSON data from!",
Toast.LENGTH_SHORT).show();
          }
        }
      });
  }

  private String getValue(String tag, Element element) {
    return
element.getElementsByTagName(tag).item(0).getChildNodes().item(0).getNodeValue();
  }
}
```

XML-CODE:

```
<?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"
  android:orientation="vertical"
  tools:context=".MainActivity">
  <TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="16dp"
    android:text="Parsing XML and JSON Data"
    android:textAlignment="center"
    android:textSize="32sp" />

  <Button
    android:id="@+id/parseXmlBtn"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="16dp"
    android:text="Parse XML" />

  <Button
    android:id="@+id/parseJsonBtn"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="16dp"
    android:text="Parse JSON" />

  <TextView
    android:id="@+id/displayTextView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="16dp"
    android:text=""
    android:textAlignment="center" />

</LinearLayout>
```

Program-7: Develop a simple application with one EditText so that the user can write some text in it. Create a button called “Convert Text to Speech” that converts the user input text into voice.

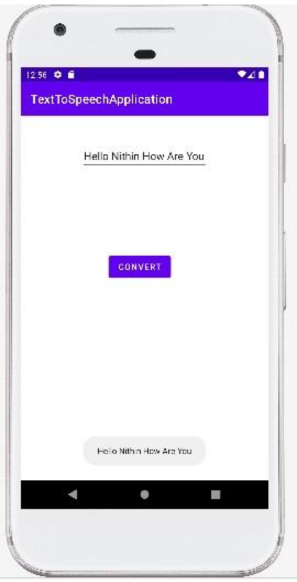
XML-CODE:

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

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="59dp"
        android:layout_marginRight="59dp"
        android:layout_marginBottom="649dp"
        android:text="Text2SpeechApp"
        android:textSize="40dp" />

    <EditText
        android:id="@+id/editText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="101dp"
        android:layout_marginRight="101dp"
        android:layout_marginBottom="514dp"
        android:ems="10"
        android:hint="Enter the text to be converted"
        android:inputType="textPersonName"
        android:text="" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="162dp"
        android:onClick="convert"
        android:layout_marginRight="162dp"
        android:layout_marginBottom="329dp"
        android:text="Convert" />
</RelativeLayout>
```

JAVA-CODE:

```
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.speech.tts.TextToSpeech;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

import java.util.Locale;

public class MainActivity extends AppCompatActivity {
    TextToSpeech t1;
    EditText e1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        e1 = (EditText)findViewById(R.id.editText);
        t1 = new TextToSpeech(getApplicationContext(), new
TextToSpeech.OnInitListener() {
            @Override
            public void onInit(int status) {
                if (status!=TextToSpeech.ERROR){
                    t1.setLanguage(Locale.UK);
                }
            }
        });
    }
    public void convert(View view){
        String tospeak = e1.getText().toString();
Toast.makeText(getBaseContext(),tospeak,Toast.LENGTH_LONG).show();
        t1.speak(tospeak,TextToSpeech.QUEUE_FLUSH,null);
    }
}
```

OUTPUT:



Hello Nithin How Are You

Hello Nithin How Are You

Program 8
Create an activity like a phone dialer with CALL and SAVE buttons. On pressing the CALL button, it must call the phone number and on pressing the SAVE button it must save the number to the phone contacts.

JAVA-CODE:

```
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.provider.ContactsContract;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {
  EditText phoneNumberEditText;
  Button clearBtn, callBtn, saveBtn;

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

    phoneNumberEditText = findViewById(R.id.phoneNumberEditText);

    clearBtn = findViewById(R.id.clearBtn);
    callBtn = findViewById(R.id.callBtn);
    saveBtn = findViewById(R.id.saveBtn);

    clearBtn.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        phoneNumberEditText.setText("");
      }
    });

    callBtn.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        String phoneNumber = phoneNumberEditText.getText().toString();

        Intent intent = new Intent(Intent.ACTION_DIAL);
        intent.setData(Uri.parse("tel:" + phoneNumber));
        startActivity(intent);
      }
    });
```

```
        saveBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String phoneNumber = phoneNumberEditText.getText().toString();

                Intent intent = new Intent(Intent.ACTION_INSERT);
                intent.setType(ContactsContract.Contacts.CONTENT_TYPE);
                intent.putExtra(ContactsContract.Intents.Insert.PHONE, phoneNumber);
                startActivity(intent);
            }
        });

    }

    public void inputNumber(View v) {
        Button btn = (Button)v;
        String digit = btn.getText().toString();
        String phoneNumber = phoneNumberEditText.getText().toString();
        phoneNumberEditText.setText(phoneNumber + digit);
    }

}
```

XML-CODE:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/phoneNumberEditText"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_margin="16dp"
        android:layout_marginTop="24dp"
        android:inputType="phone"
        android:textSize="24sp"
        app:layout_constraintEnd_toStartOf="@+id/clearBtn"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/clearBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="16dp"
        android:text="Clear"
        app:layout_constraintBottom_toBottomOf="@+id/phoneNumberEditText"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toEndOf="@+id/phoneNumberEditText"
        app:layout_constraintTop_toTopOf="@+id/phoneNumberEditText" />

    <TableLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="32dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/phoneNumberEditText">

        <TableRow
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:gravity="center_horizontal">

            <Button
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_margin="8dp"
                android:onClick="inputNumber"
                android:text="7" />

            <Button
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_margin="8dp"
                android:onClick="inputNumber"
                android:text="8" />

            <Button
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_margin="8dp"
                android:onClick="inputNumber"
            android:text="9" />

    </TableRow>

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center_horizontal">

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="8dp"
            android:onClick="inputNumber"
            android:text="4" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="8dp"
            android:onClick="inputNumber"
            android:text="5" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="8dp"
            android:onClick="inputNumber"
            android:text="6" />

    </TableRow>

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center_horizontal">

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="8dp"
            android:onClick="inputNumber"
            android:text="1" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="8dp"
            android:onClick="inputNumber"
            android:text="2" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="8dp"
            android:onClick="inputNumber"
            android:text="3" />

    </TableRow>

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center_horizontal">

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="8dp"
            android:onClick="inputNumber"
            android:text="*" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="8dp"
            android:onClick="inputNumber"
            android:text="0" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="8dp"
            android:onClick="inputNumber"
            android:text="#" />

    </TableRow>

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center_horizontal">
```

```
        <Button
          android:id="@+id/callBtn"
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:layout_margin="8dp"
          android:text="Call" />

        <Button
          android:id="@+id/saveBtn"
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:layout_margin="8dp"
          android:text="Save" />

      </TableRow>
    </TableLayout>


</androidx.constraintlayout.widget.ConstraintLayout>
```

Program 9. (PART-B): 5. Create an application to demonstrate a basic media player that allows the user to Forward Backward, Play and Pause an audio. Also, make use of the indicator in the seek bar to move the audio forward or backward as required.

XML-CODE:

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


    <SeekBar
        android:id="@+id/seekBar"
        android:layout_width="255dp"
        android:layout_height="28dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="66dp"
        android:layout_marginBottom="311dp" />

    <ImageButton
        android:id="@+id/rewind"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="256dp"
        android:layout_marginBottom="219dp"
        app:srcCompat="@android:drawable/ic_media_rew" />

    <ImageButton
        android:id="@+id/playButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="168dp"
        android:layout_marginBottom="223dp"
        app:srcCompat="@android:drawable/ic_lock_power_off" />

    <ImageButton
        android:id="@+id/forward"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="79dp"
        android:layout_marginBottom="220dp"
        app:srcCompat="@android:drawable/ic_media_ff" />


</RelativeLayout>
```

Java CODE:

```
import androidx.appcompat.app.AppCompatActivity;
import android.annotation.SuppressLint;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.widget.ImageButton;
import android.widget.SeekBar;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    private ImageButton playButton,forward,rewind;
    private SeekBar seekbar;
    private MediaPlayer mediaPlayer;
    private Handler handler = new Handler();
    @SuppressLint("ClickableViewAccessibility")

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        playButton = findViewById(R.id.playButton);
        forward = findViewById(R.id.forward);
        rewind = findViewById(R.id.rewind);
    seekbar = findViewById(R.id.seekBar);
    prepareMediaPlayer();
    seekbar.setMax(100);
    playButton.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        if(mediaPlayer.isPlaying()){
          handler.removeCallbacks(updater);
          mediaPlayer.pause();

        }else {
          mediaPlayer.start();

          updateSeekBar();
        }
      }
    });
    forward.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        if(mediaPlayer.getDuration()>mediaPlayer.getCurrentPosition() +
10000){
          mediaPlayer.seekTo(mediaPlayer.getCurrentPosition() + 10000);
          updateSeekBar();
        }
      }
    });
    rewind.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        if(mediaPlayer.getCurrentPosition()>10000){
          mediaPlayer.seekTo(mediaPlayer.getCurrentPosition() - 10000);
          updateSeekBar();
        }
      }
    });
    seekbar.setOnTouchListener((v, event) -> {
      SeekBar s = (SeekBar) v;
      int position = (mediaPlayer.getDuration()/100)*s.getProgress();
      mediaPlayer.seekTo(position);
      return false;
    });
```

```
        mediaPlayer.setOnCompletionListener(new
MediaPlayer.OnCompletionListener() {
            @Override
            public void onCompletion(MediaPlayer mp) {
                seekbar.setProgress(0);
                mediaPlayer.reset();
                prepareMediaPlayer();
            }
        });
    }
    private void prepareMediaPlayer (){
        try {
            mediaPlayer =MediaPlayer.create(this,R.raw.poc);
        }catch (Exception e){
            Toast.makeText(this, e.getMessage(), Toast.LENGTH_LONG).show();
        }
    }
    private Runnable updater = new Runnable() {
        @Override
        public void run() {
            updateSeekBar();
        }
    };
    private void updateSeekBar(){
        if(mediaPlayer.isPlaying()){

seekbar.setProgress((int)((float)mediaPlayer.getCurrentPosition()/mediaPlayer.
getDuration()*100));
            handler.postDelayed(updater,1000);
        }
    }
}
```

Program 10: (PART-B): 6. Develop an application to demonstrate the use of Asynchronous tasks in android. The asynchronous task should implement the functionality of a simple moving banner. On pressing the Start Task button, the banner message should scroll from right to left. On pressing the Stop Task button, the banner message should stop. Let the banner message be "Demonstration of Asynchronous Task".

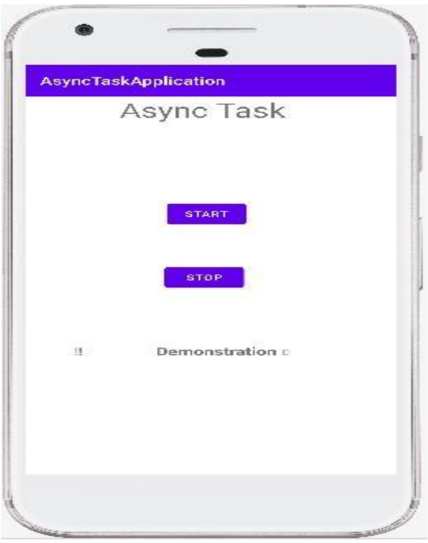
XML-CODE:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="123dp"
        android:layout_marginBottom="630dp"
        android:text="Async Task"
        android:textSize="36sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <Button
        android:id="@+id/buttonstart"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="167dp"
        android:layout_marginBottom="441dp"
        android:text="Start" />
    <Button
        android:id="@+id/buttonstop"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="169dp"
        android:layout_marginBottom="328dp"
        android:text="Stop" />
    <TextView
        android:id="@+id/marqueeText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginStart="50dp"
        android:layout_marginTop="250dp"
        android:layout_marginEnd="117dp"
        android:layout_marginBottom="207dp"
        android:ellipsize="marquee"
        android:marqueeRepeatLimit="marquee_forever"
        android:scrollHorizontally="true"
        android:singleLine="true"
        android:text="Demonstration of Asynchronous Task !!!!"
        android:textSize="20sp"
        android:textStyle="bold"
        android:visibility="invisible" />
</RelativeLayout>
```

JAVA-CODE:

```
import androidx.appcompat.app.AppCompatActivity;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
    TextView marqtxt;
    Button btnstart, btnstop;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        marqtxt = (TextView) findViewById(R.id.marqueeText);
        btnstart = (Button) findViewById(R.id.buttonstart);
        btnstop = (Button) findViewById(R.id.buttonstop);
        btnstart.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                ExampleAsyncTask task = new ExampleAsyncTask();
                task.execute();
            }
        });
        btnstop.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                marqtxt.setSelected(false);
                marqtxt.setVisibility(View.INVISIBLE);
            }
        });
    }
    private class ExampleAsyncTask extends AsyncTask<String, String,String>{
        @Override
        protected void onPreExecute() {
            super.onPreExecute();
     Toast.makeText(getBaseContext(),"Async Task
Started!!!!!!!!",Toast.LENGTH_SHORT).show();
        }
       @Override
        protected String doInBackground(String... strings) {
            try {
                Thread.sleep(250);
            }
            catch (InterruptedException e){
                e.printStackTrace();
            }
            return null;
        }
        @Override
        protected void onPostExecute(String s) {
            super.onPostExecute(s);
            marqtxt.setVisibility(View.VISIBLE);
            marqtxt.setSelected(true);
        }
    }
}
```

OUTPUT:



Async Task

!!   Demonstration o

Program 11: (PART-B): 7. Develop an application that makes use of the clipboard framework for copying and pasting of the text. The activity consists of two EditText controls and two Buttons to trigger the copy and paste functionality.

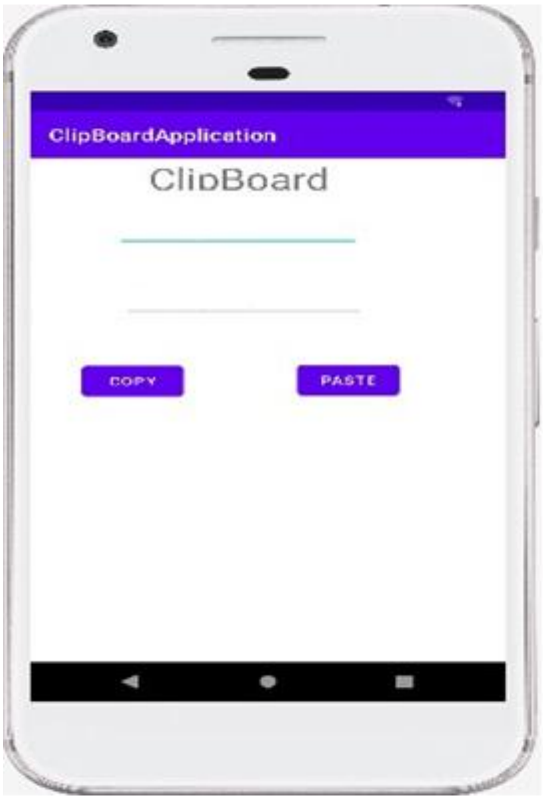
XML-CODE:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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">
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="152dp"
        android:layout_marginBottom="564dp"
        android:text="ClipBoard"
        android:textSize="36sp" />
    <EditText
        android:id="@+id/editText1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="127dp"
        android:layout_marginBottom="496dp"
        android:ems="10"
        android:hint="Enter the text here"
        android:inputType="textPersonName"
        android:text="" />
    <EditText
        android:id="@+id/editText2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="122dp"
        android:layout_marginBottom="411dp"
        android:ems="10"
        android:hint="Copied Text"
        android:inputType="textPersonName"
        android:text="" />
    <Button
        android:id="@+id/copy"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="279dp"
        android:onClick="copy"
        android:layout_marginBottom="312dp"
        android:text="Copy" />
    <Button
        android:id="@+id/paste"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="92dp"
        android:onClick="paste"
        android:layout_marginBottom="313dp"
        android:text="Paste" />
</RelativeLayout>
```

JAVA-CODE:

```
import androidx.appcompat.app.AppCompatActivity;
import android.content.ClipData;
import android.content.ClipboardManager;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
public class MainActivity extends AppCompatActivity {
    ClipboardManager cbm;
    ClipData cd;
    EditText e1,e2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        e1 = (EditText)findViewById(R.id.editText1);
        e2 = (EditText)findViewById(R.id.editText2);
        cbm =(ClipboardManager)getSystemService(CLIPBOARD_SERVICE);
    }
    public void copy(View V){
        String text = e1.getText().toString();
        cd = ClipData.newPlainText("text",text);
        cbm.setPrimaryClip(cd);
    }
    public void paste(View V){
        ClipData cd2 = cbm.getPrimaryClip();
        ClipData.Item item = cd2.getItemAt(0);
        String copied = item.getText().toString();
        e2.setText(copied);
    }
}
```

OUTPUT:



MOBILE APPLICATION DEVELOPMENT

COURSE CODE:18CSMP68

Sample Viva Questions

1. What is an Android?
2. Illustrate the Android lifecycle Activity.
3. What Is an Android SDK?
4. What is Android "compatibility"?
5. What are the key components Android Architecture?
6. Describe the Android Framework.
7. What are the data types supported by AIDL?
8. What is Gradle Framework ?
9. Why do we need AVD?
10. How do you add gradle dependencies ?
11. What is the difference between Mobile Application Testing and Mobile Testing?
12. What are the different data storage options available on the Android platform?
13. The list of data storage options on the Android platform.
14. Describe Activities.
15. What are Intents? What are the types of Intents?
16. What is Application class?
17. What is a View?
18. What is a view Group?
19. What are these UI components that we can use in our application?
20. Define Constraint Layout.
21. Why did we need Constraint Layout?
22. What are the different types of Android widgets?
23. Which are the files that demonstrate implementing and using of the custom widget?
24. Enumerate the three key loops while monitoring an activity?
25. What are the major steps involved in creating a bounded service through AIDL?