Kammavari Sangham (R) 1952,

K.S.Group of Institutions

K.S. INSTITUTE OF TECHNOLOGY

#14, Raghuvanahalli, Kanakapura Main Road, Bengaluru 560-109

Department of Computer Science and Engineering

LABORATORY MANUAL

Mobile Application Development

Sub code: 18CSMP68

Prepared by:

Mr. Harshavardhan.J.R
Associate Professor

Mr. Prashanth.H.S
Assistant Professor

Mrs. Beena.K
Assistant Professor

Document Log

Name of the document	Mobile Application Development Laboratory
Syllabus Scheme	2018-2019
Current version number and date	V1 / 25.03.2021
Subject code	18CSMP68
Editorial Committee	Mobile Application Development Lab Faculty Members
Approved by	HOD, Dept. of CSE
Lab faculty	Prof. Harshavardhan.J.R Prof. Prashanth.H.S Prof. Beena.K
Computer Programmer	Ms. B.Bhuvaneshwari

K S INSTITUTE OF TECHNOLOGY

VISION

"To impart quality technical education with ethical values, employable skills and research to achieve excellence"

MISSION

- **To attract and retain highly qualified, experienced & committed faculty.**
- **To create relevant infrastructure**
- **Network with industry & premier institutions to encourage emergence of new ideas by providing research & development facilities to strive for academic excellence**
- **To inculcate the professional & ethical values among young students with employable skills & knowledge acquired to transform the society**

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

VISION

"To create competent professionals in Computer Science and Engineering with adequate skills to drive the IT industry".

MISSION

- **Impart sound technical knowledge and quest for continuous learning.**
- **To equip students to furnish Computer Applications for the society through experiential learning and research with professional ethics.**
- **Encourage team work through inter-disciplinary project and evolve as leaders with social concerns.**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Program Educational Objectives

PEO1: Excel in professional career by acquiring knowledge in cutting edge Technology and contribute to the society as an excellent employee or as an entrepreneur in the field of Computer Science & Engineering.

PEO2: Continuously enhance their knowledge on par with the development in IT industry and pursue higher studies in computer science &engineering.

PEO3 : Exhibit professionalism, cultural awareness, team work, ethics, and effective communication skills with their knowledge in solving social and environmental problems by applying computertechnology.

Program Specific Outcomes

PSO1 : Ability to understand, analyze problems and implement solutions in Programming languages, as well to apply concepts in core areas of Computer Science in association with professional bodies and clubs.

PSO2 : Ability to use computational skills and apply software knowledge to develop effective solutions and data to address real world challenges.

Program Outcomes (POs)

Engineering Graduates will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Course Details

Course Name: **Mobile Application Development**

Course Code: **18CSMP68**

Course prerequisite: Core Java

Course Objectives

Upon completion of this course, students are expected to:

1. Learn and acquire the art of Android Programming.
2. Configure Android studio to run the applications.
3. Understand and implement Android's User interface functions.
4. Create, modify and query on SQLite database.
5. Inspect different methods of sharing data using services.

SYLLABUS

MOBILE APPLICATION DEVELOPMENT

Subject Code: 18CSMP68 **IA Marks: 40**

No. of Practical Hrs. / Week: 0:0:2 **Exam Marks: 60**

Total No. of Practical Hrs: 3 Hours/Week **Exam Hours: 03**

No. of Credits: 02

Descriptions (if any):

1. The installation procedure of the Android Studio/Java software must be demonstrated and carried out in groups.
2. Students should use the latest version of Android Studio/Java/ Kotlin to execute these programs. Diagrams given are for representational purposes only, students are expected to improvise on them.
3. Part B programs should be developed as an application and are to be demonstrated as a mini project in a group by adding extra features or the students can also develop their application and demonstrate it as a mini-project. (Projects/programs are not limited to the list given in Part B).

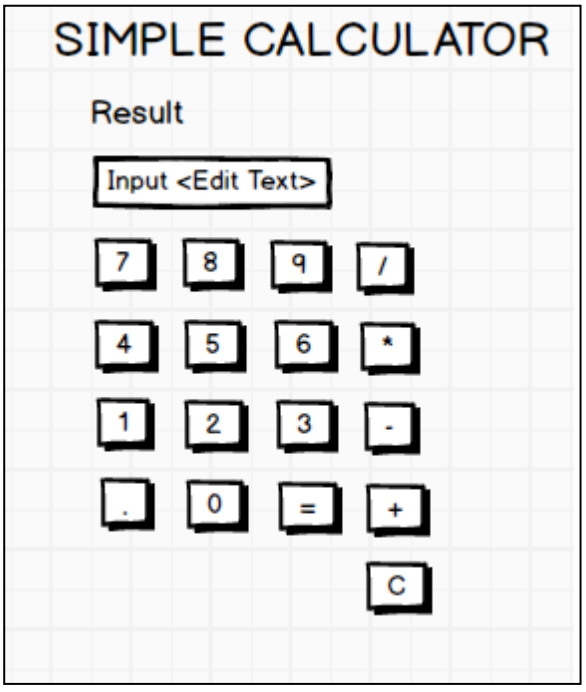
PART A

Program 1

Program 2

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.

Develop an Android application using controls like Button, TextView, EditText for designing a Calculator having basic functionality like Addition, Subtraction, Multiplication, and Division.



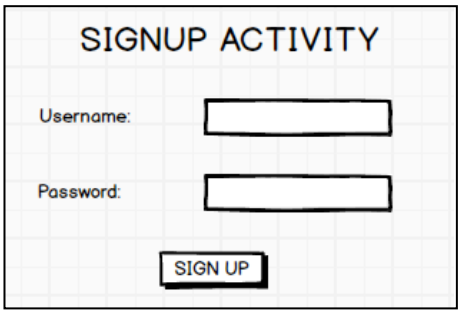
Program 3

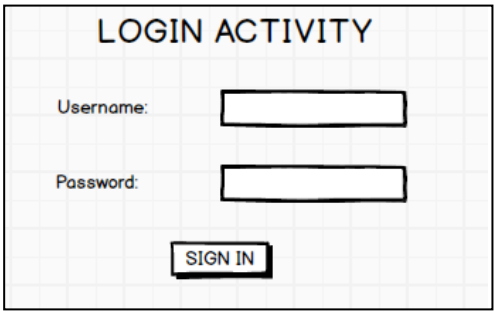
Create a SIGN Up activity with Username and Password. Validation of password should happen based on the following rules:

Program 2

- 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).

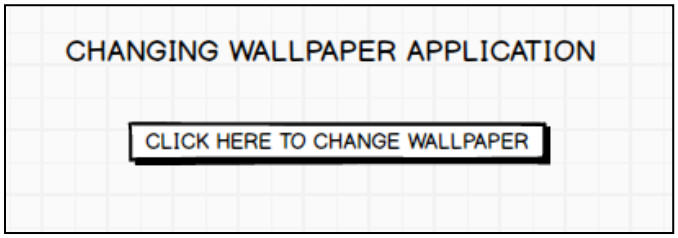
On successful **SIGN UP** proceed to the next Login activity. Here the user should **SIGN IN** using the Username and Password created during signup activity. If the Username and Password are matched then navigate to the next activity which displays a message saying "Successful Login" or else display 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.





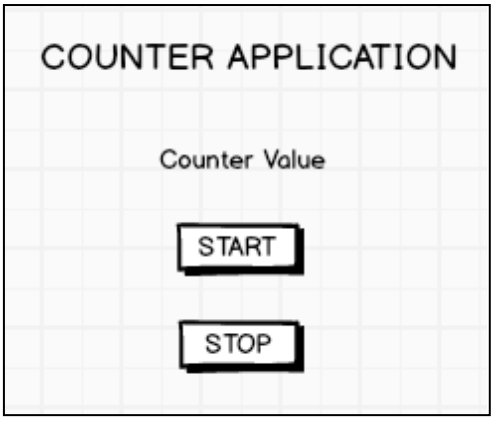
Program 4

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.



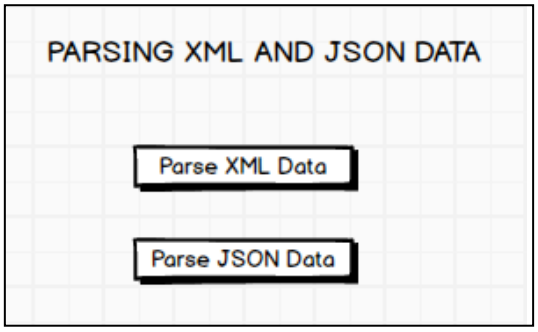
Program 5

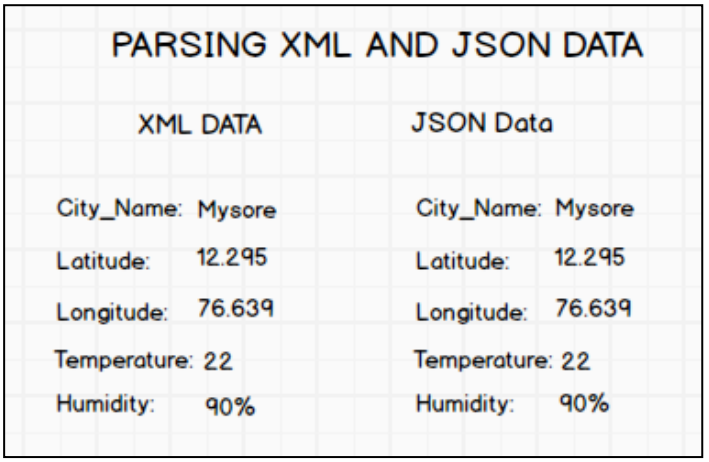
Write a program to create an activity with two buttons START and STOP. On Pressing of the START button, the activity must start the counter by displaying the numbers from One and the counter must keep on counting until the STOP button is pressed. Display the counter value in a TextView control.



Program 6

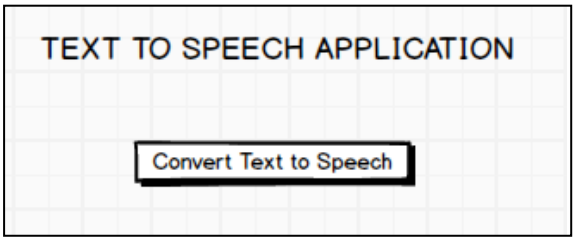
Create two files of XML and JSON type with values for City_Name, Latitude, Longitude, Temperature, and Humidity. Develop an application to create an activity with two buttons to parse the XML and JSON files which when clicked should display the data in their respective layouts side by side.





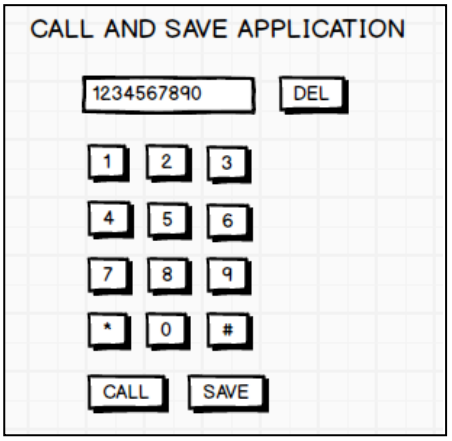
Program 7

Develop a simple application with one Edit Text 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.



Program 8

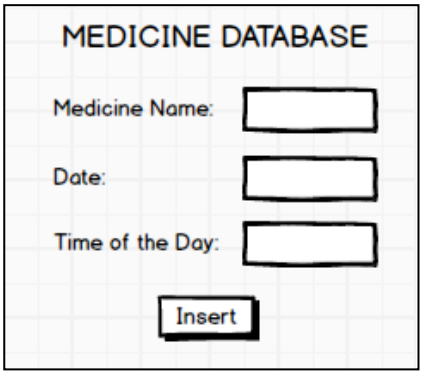
Create an activity like a phone dialer with CALL and SAVE buttons. On pressing the CALL button, it must call the phone number and on pressing the SAVE button it must save the number to the phone contacts.



PART B

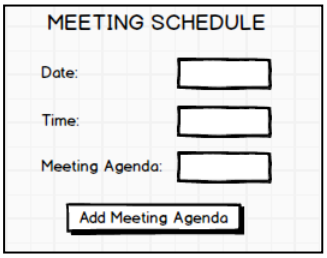
Program 1

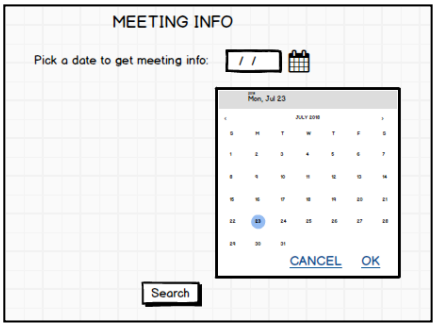
Write a program to enter Medicine Name, Date and Time of the Day as input from the user and store it in the SQLite database. Input for Time of the Day should be either Morning or Afternoon or Evening or Night. Trigger an alarm based on the Date and Time of the Day and display the Medicine Name.



Program 2

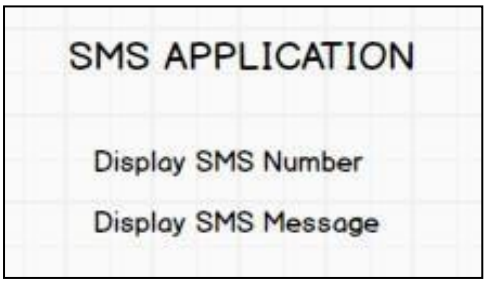
Develop a content provider application with an activity called "Meeting Schedule" which takes Date, Time and Meeting Agenda as input from the user and store this information into the SQLite database. Create another application with an activity called "Meeting Info" having DatePicker control, which on 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".





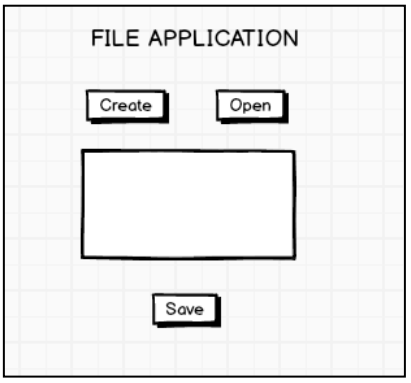
Program 3

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.



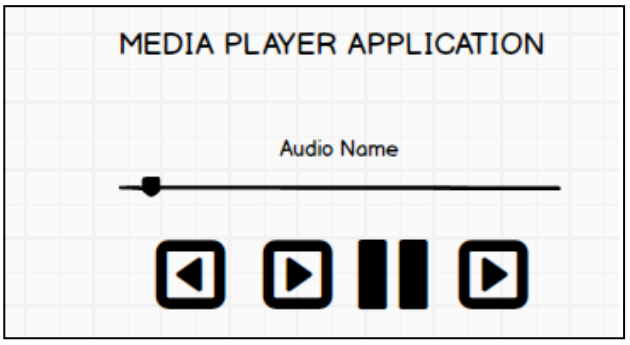
Program 4

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 "FirstCreate a File".



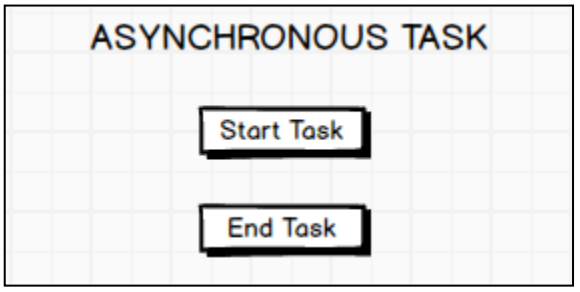
Program 5

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.



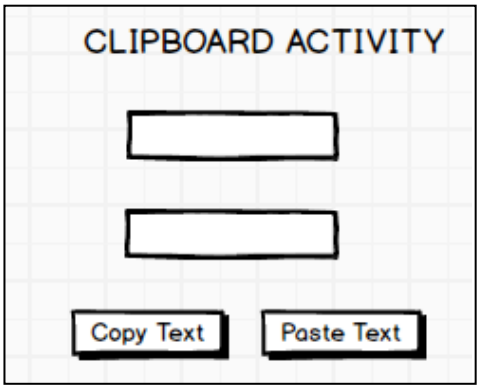
Program 6

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".



Program 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.



Program 8

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

$E = P * (r(1+r)^n)/((1+r)^n-1)$

where

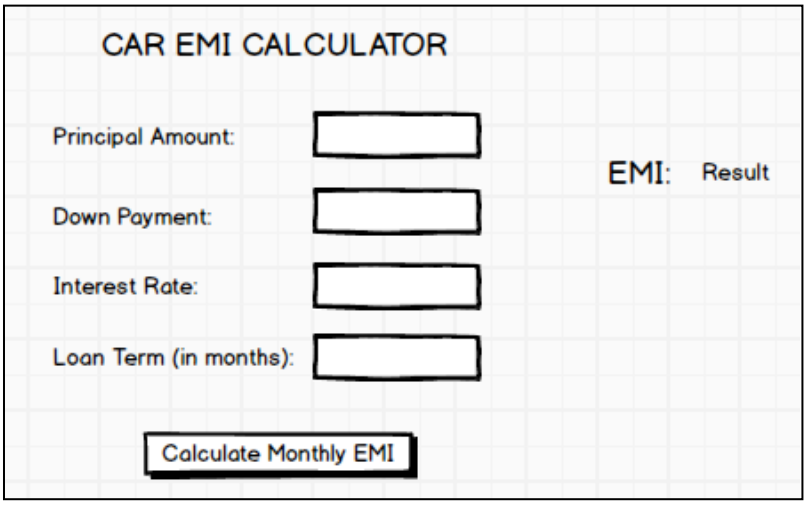
E = The EMI payable on the car loan amount

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 Principal Amount, 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.



Course Outcomes

After successful completion of the Course, the participants will be able to

CO#	COURSE OUTCOMES	K-LEVEL
18CSMP68.1	Learn and acquire the art of Android Programming.	**Understanding (K2)**
18CSMP68.2	Configure Android studio to run the applications.	**Understanding (K2)**
18CSMP68.3	Understand and implement Android's User interface functions.	**Applying (K3)**
18CSMP68.4	Create, modify and query on SQlite database.	**Applying (K3)**
18CSMP68.5	Inspect different methods of sharing data using services.	**Applying (K3)**

CO-PO Mapping

CO No.	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
18CSMP68.1	3	2	2	-	2	1	-	-	1	1	-	1
18CSMP68.2	3	2	2	-	2	1	-	-	1	1	-	1
18CSMP68.3	3	2	2	-	2	1	-	-	1	1	-	1
18CSMP68.4	3	2	2	-	-	-	-	2	1	1	-	1
18CSMP68.5	3	2	2	-	-	-	-	1	1	1	-	1

Procedure to Conduct Practical Examination

Experiment distribution

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

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

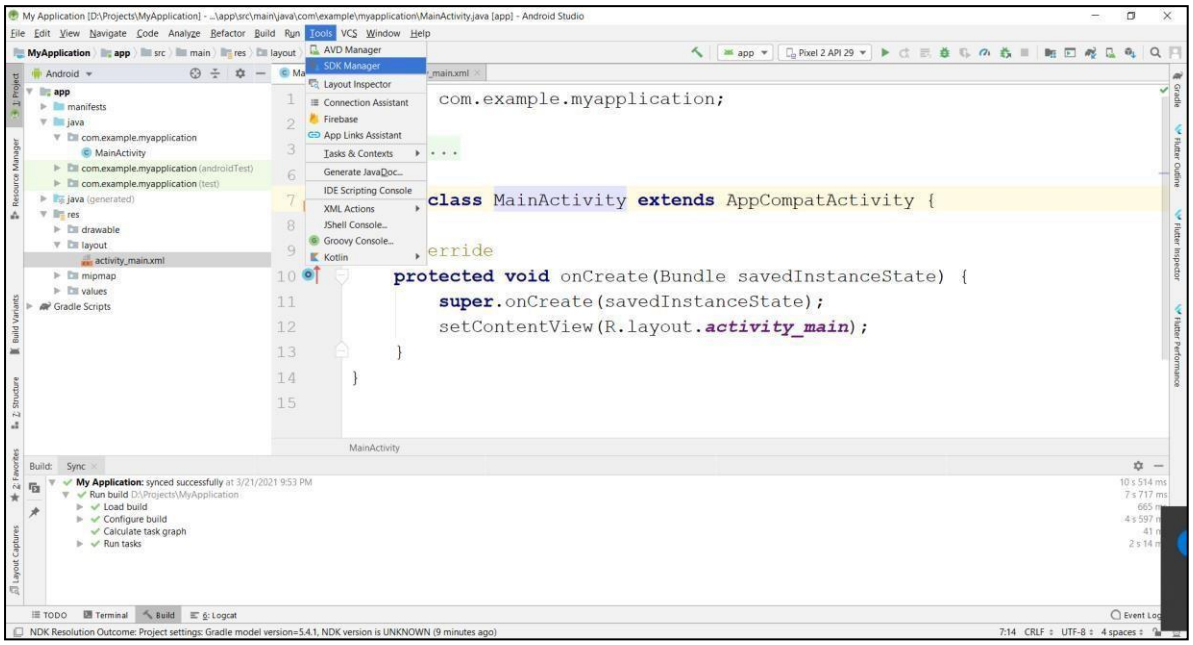
1. Android Studio Tutorials

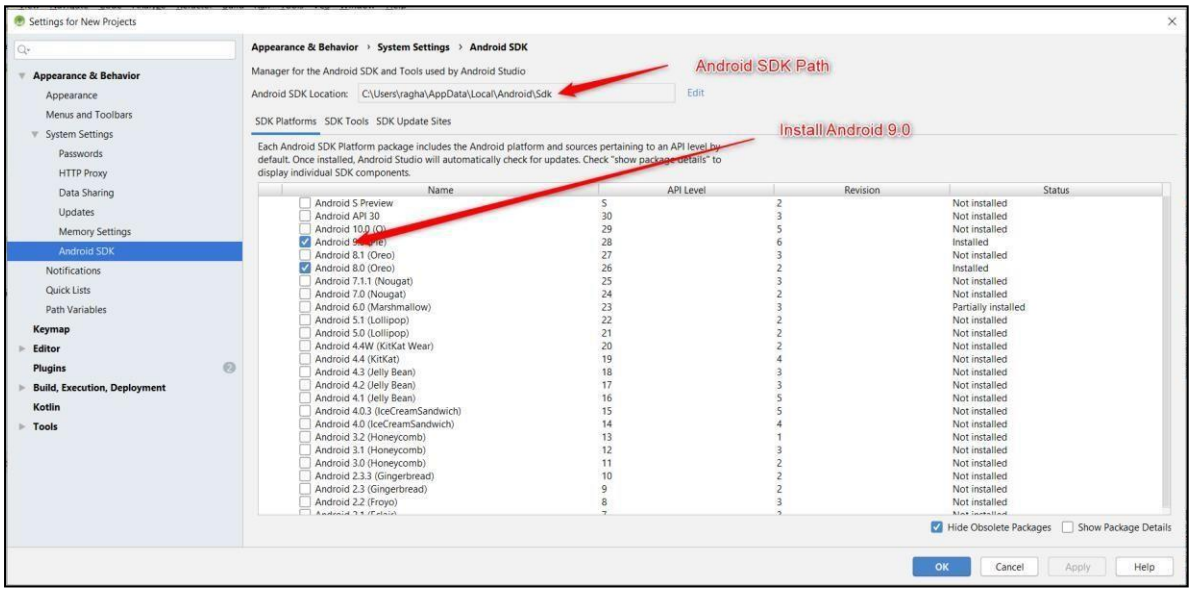
Install Android Studio and Packages:

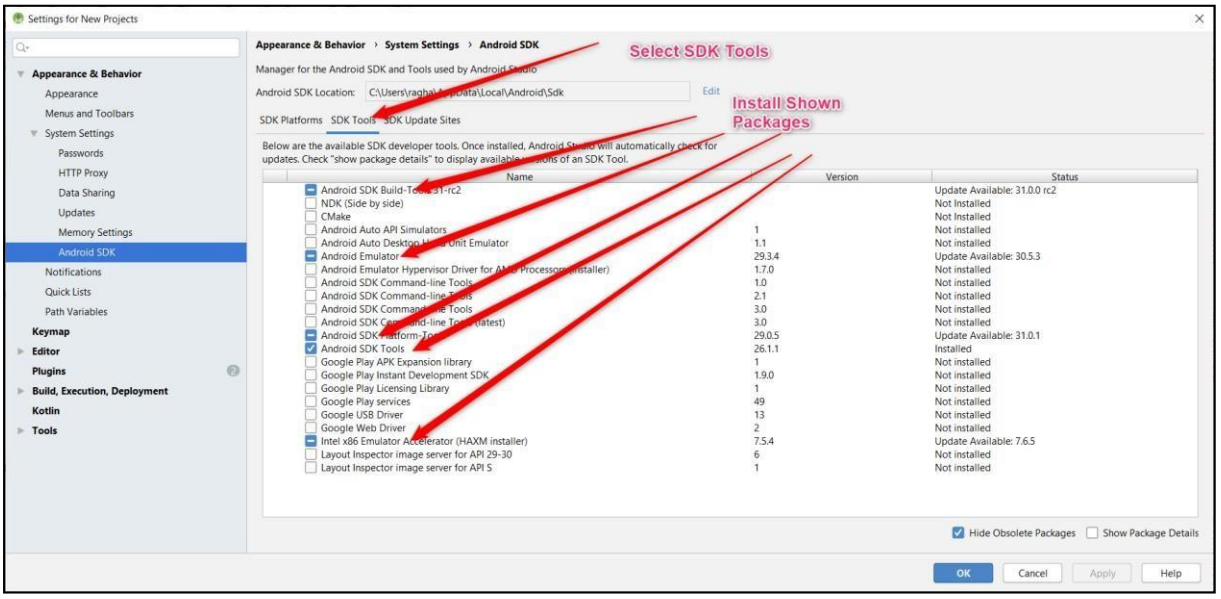
Download Android Version 4.0.2 from the below link
https://redirector.gvt1.com/edgedl/android/studio/install/4.0.2.0/android-studio-ide-193.6821437-windows.exe

Configure Android SDK packages:

Go to Tools ➔ SDK Manager

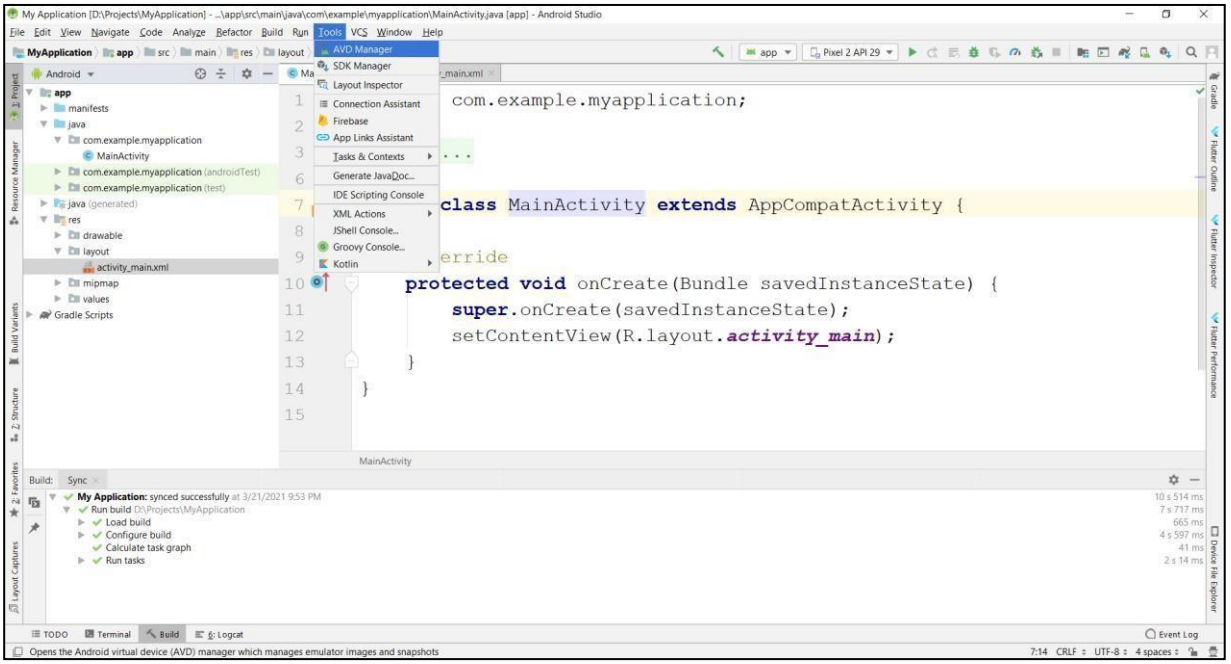


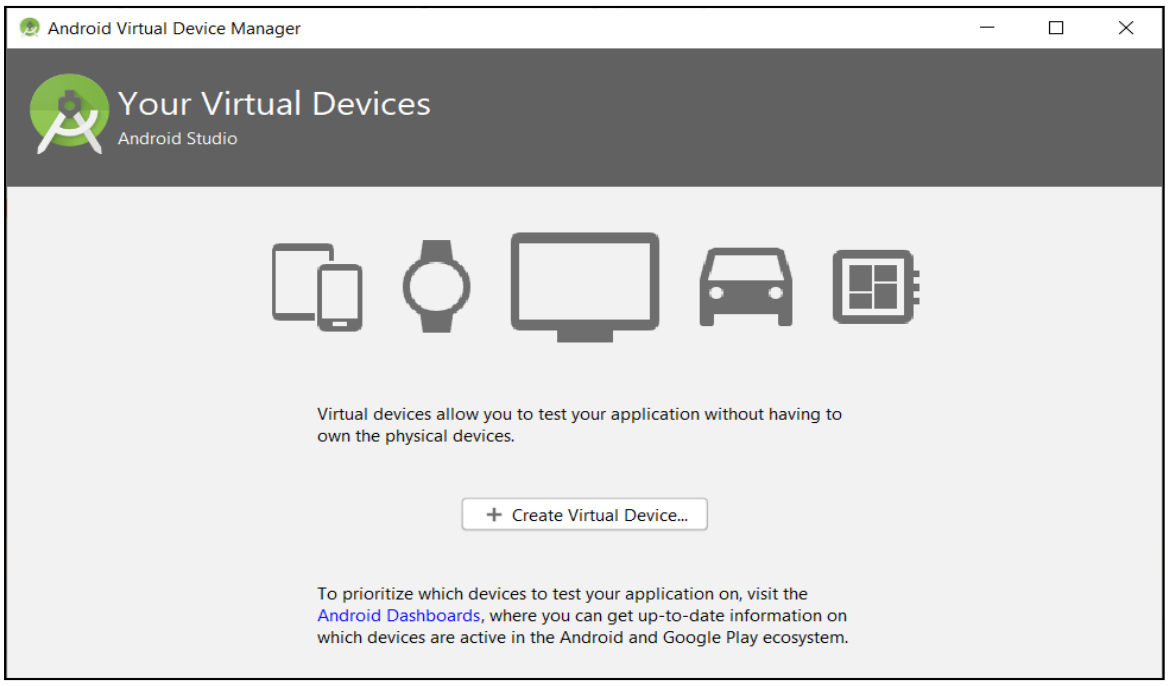




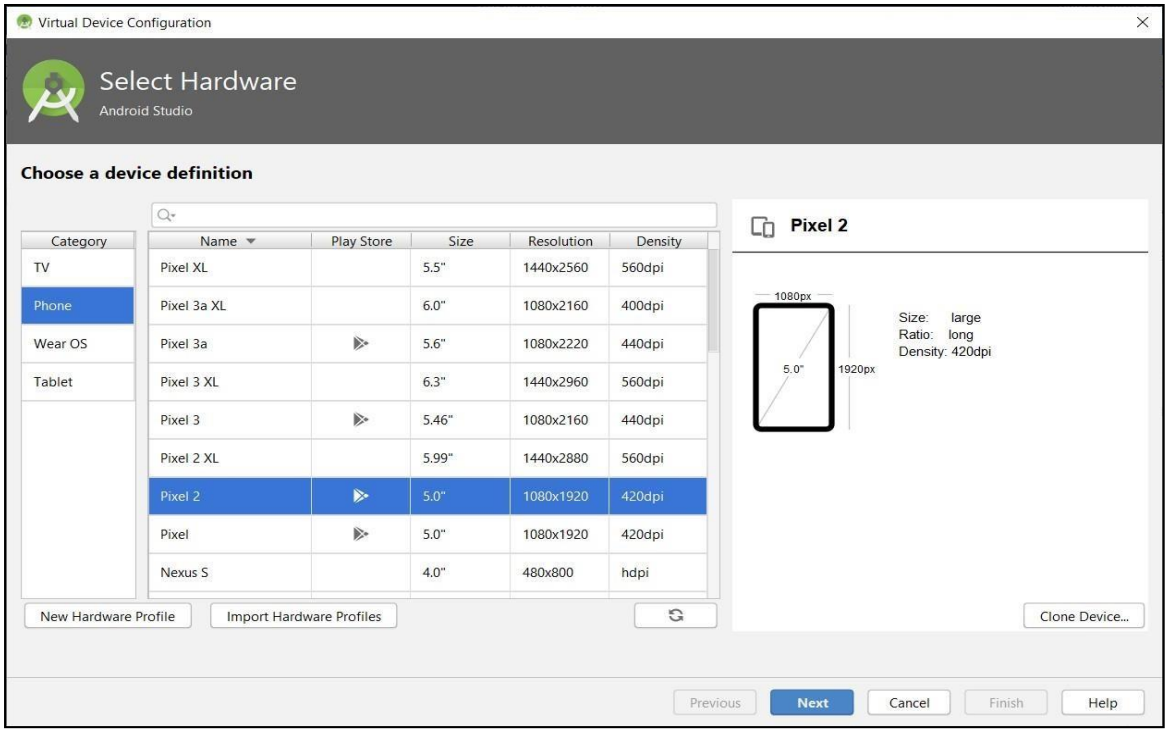
Creating Emulator

Go to Tools ➔ Select AVD Manager

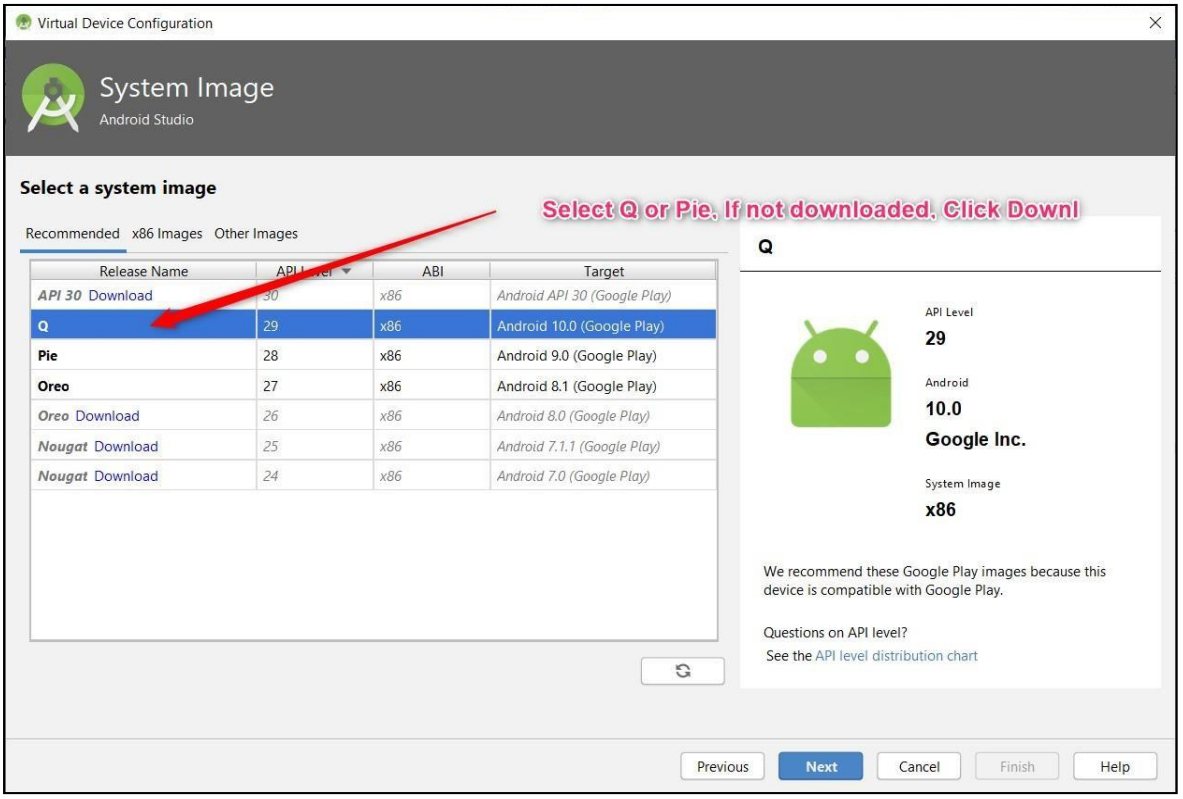




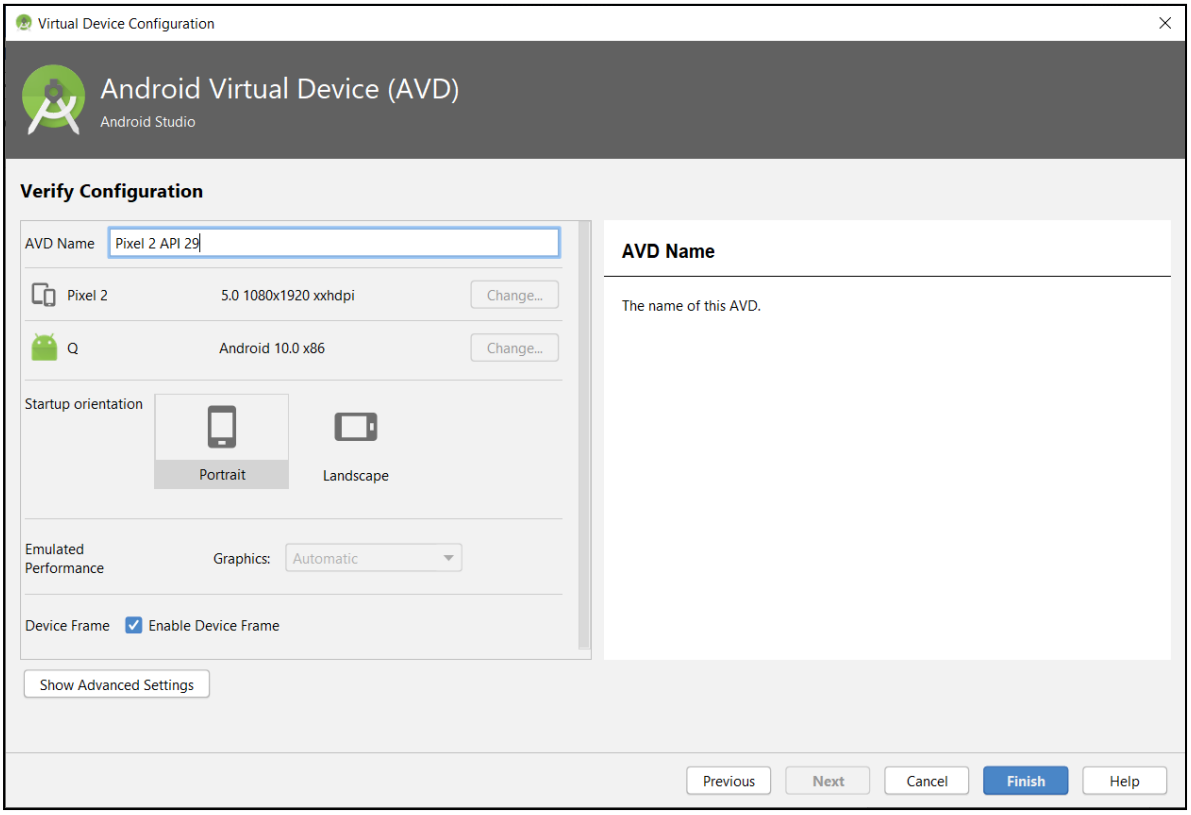
Select Create Virtual Device ➔Select Phone ➔Pixel 2 ➔ Press Next



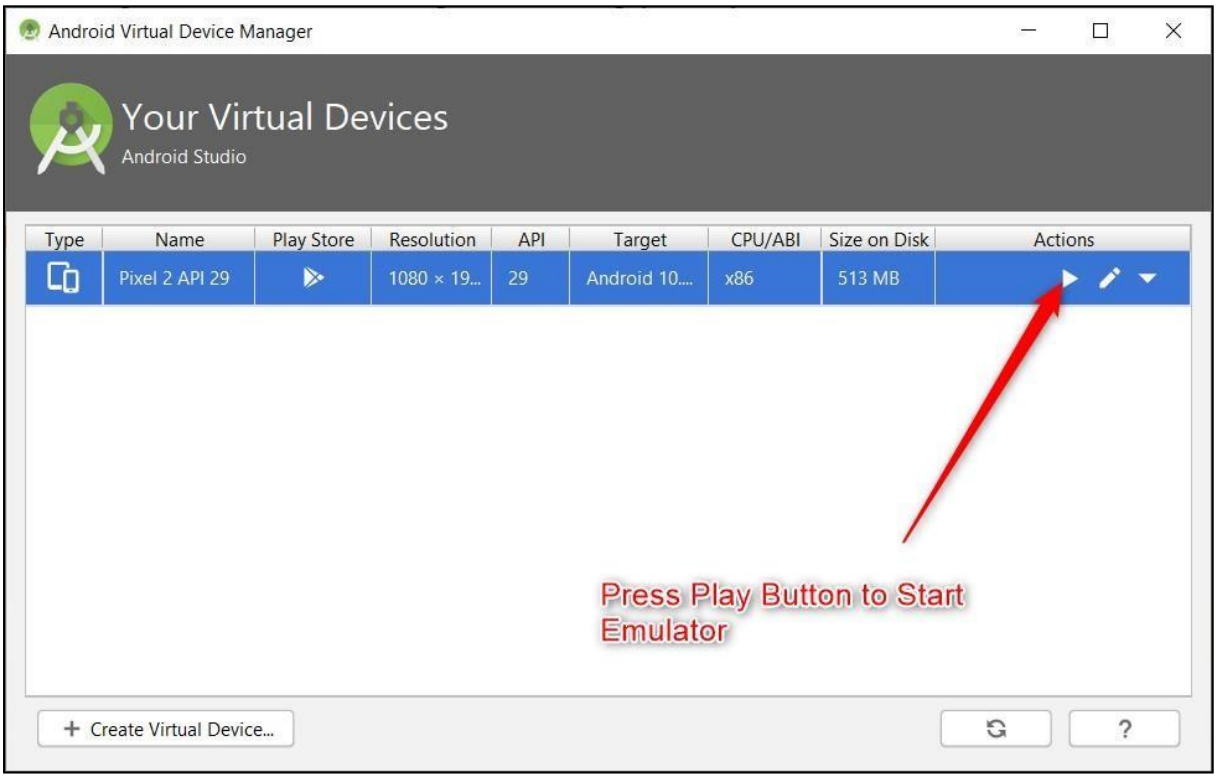
Select Android Q, if not already downloaded press download, After download completes Select Q and Press Next Button.

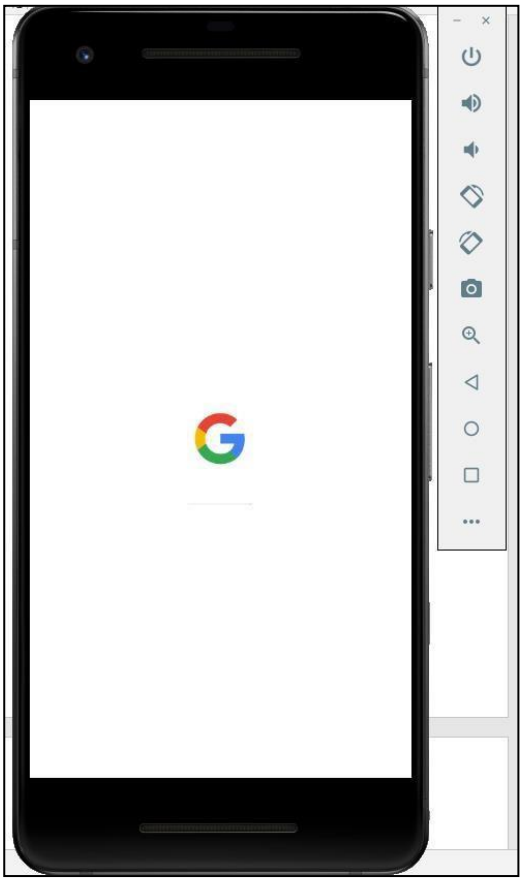


Enter AVD Name and Press Finish.



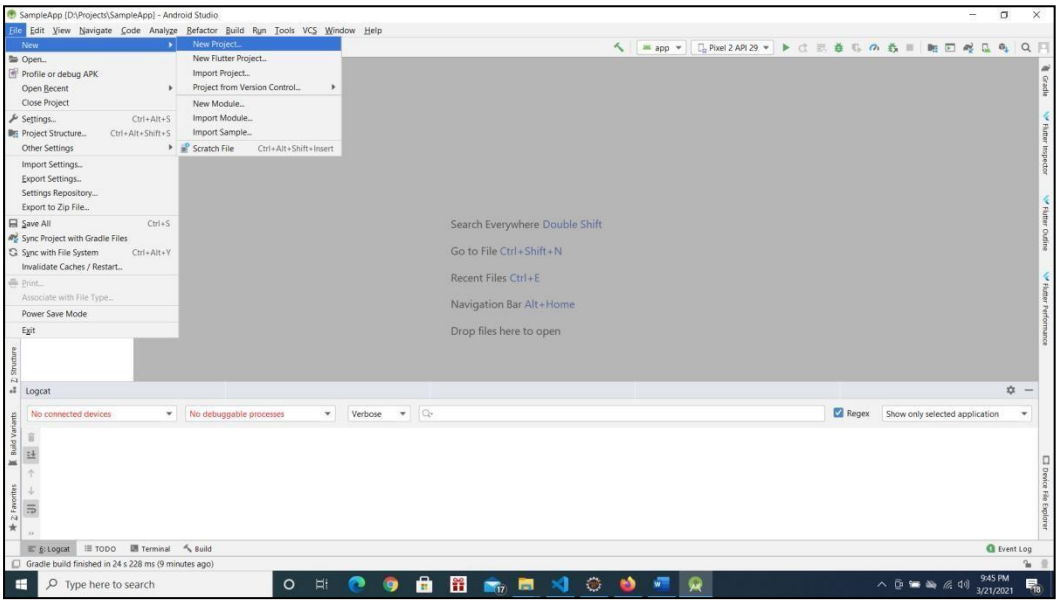
Press Play Button to Start Emulator


Android Virtual Device Manager
Your Virtual Devices
Android Studio
Type
Name
Play Store
Resolution
API
Target
CPU/ABI
Size on Disk
Actions
Pixel 2 API 29
1080 × 19...
29
Android 10....
x86
513 MB
Press Play Button to Start Emulator
Create Virtual Device...

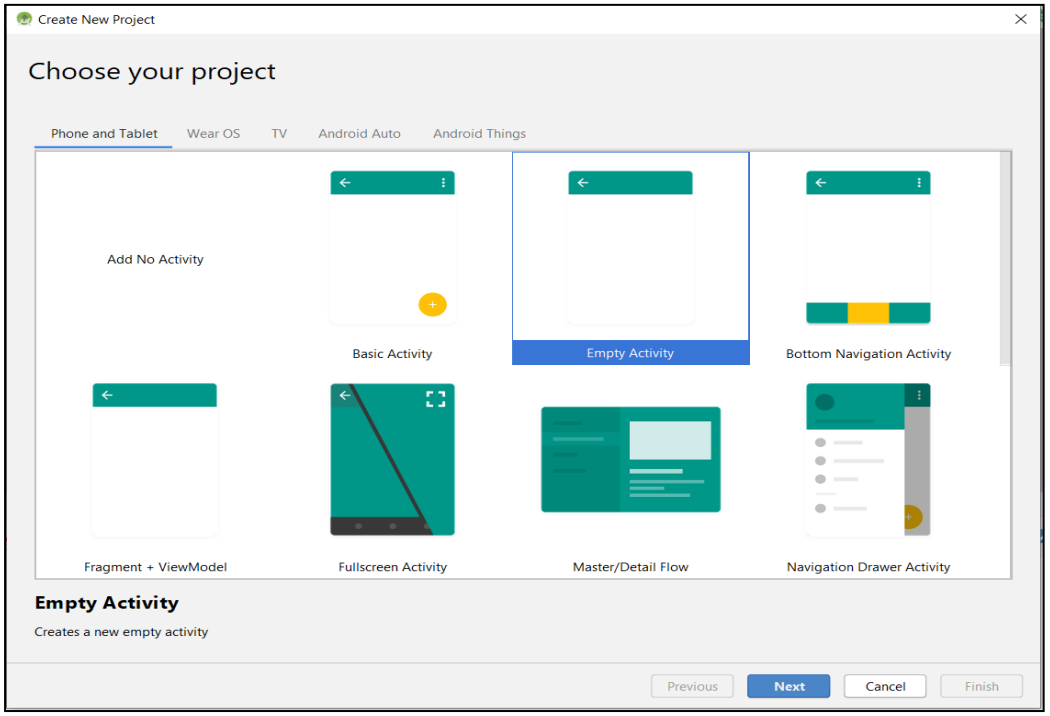


Creating a New Project in Android

While creating a New Project for First Time, make sure Android Studio is connected to internet, It downloads the required packages from internet.
Go to File ➔New ➔New Project



Choose Phone and Tablet ➔ Empty Activity ➔ Press Next



In Configure your Project Screen, Enter below details and Press Finish Button.
Enter Name of the Application ➔ This will be application name this will be visible with Home Screen Icon.

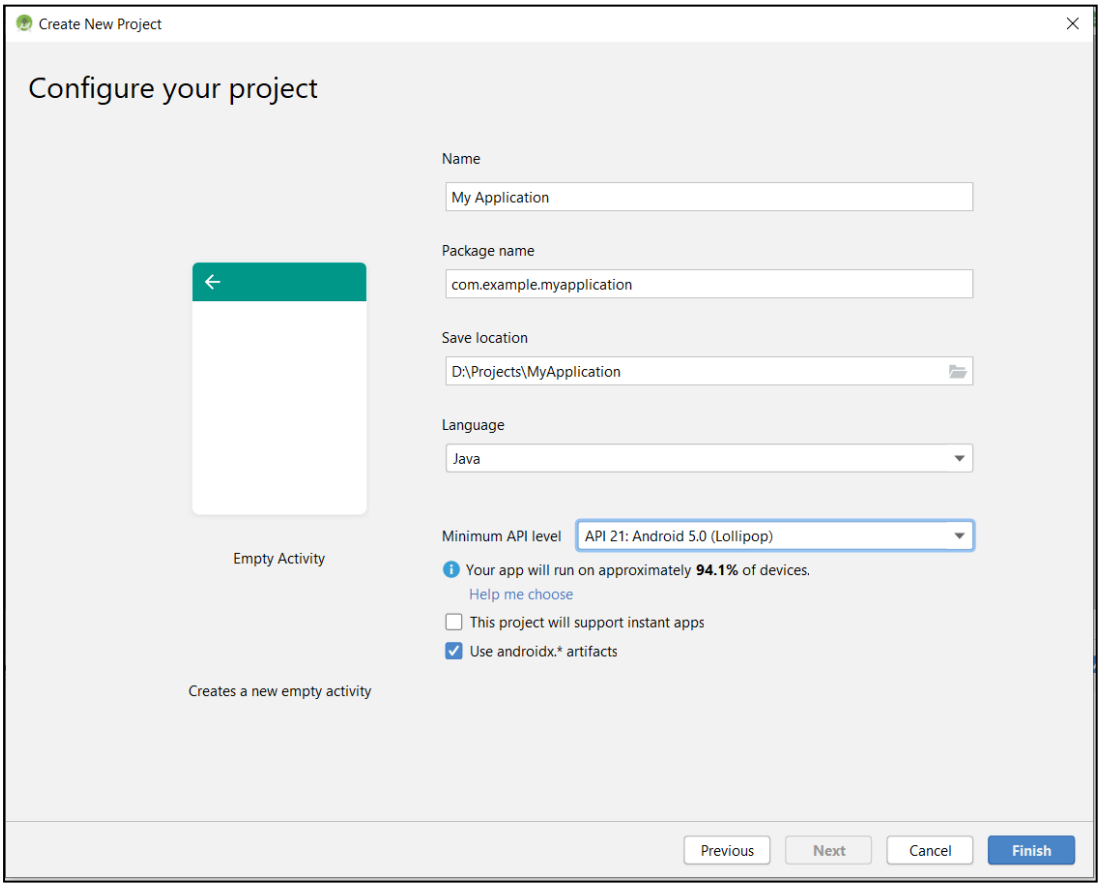
Package Name ➔ Enter package name atleast two identifier (Eg: com.example). Best Practice is 3 or more identifier (Eg: com.example.firstapp).

Save Location ➔Location where to save the Project

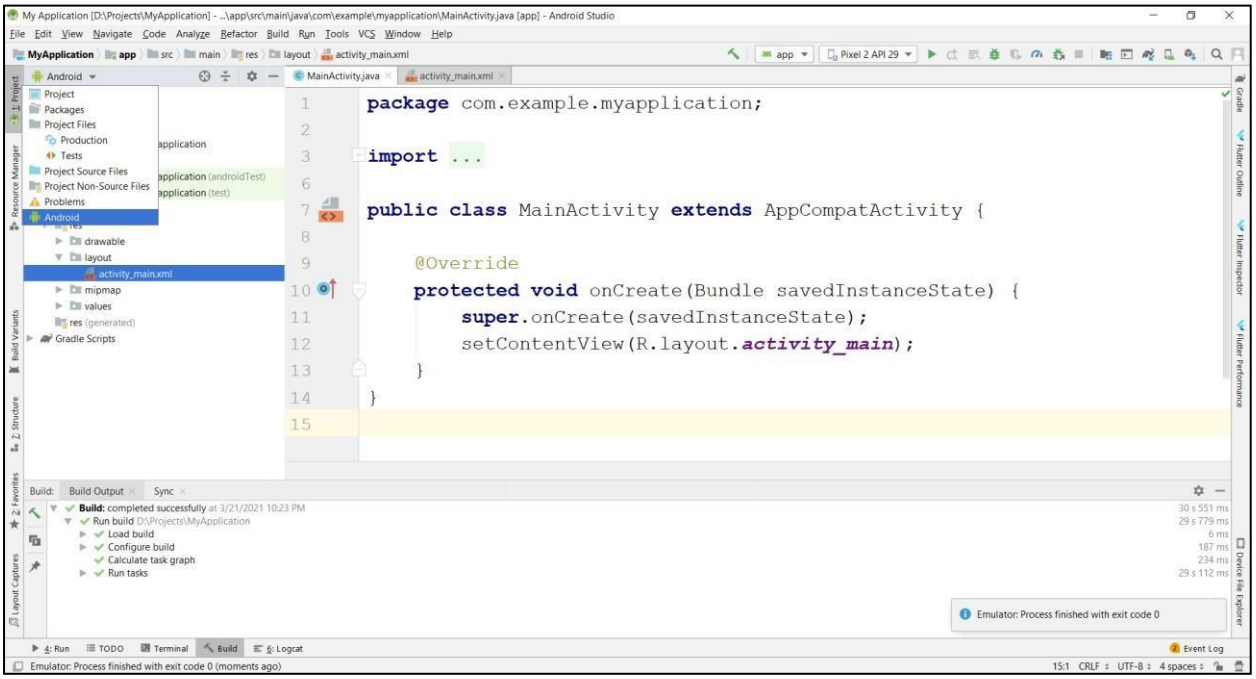
Language ➔ Choose Java

Minimum API Level ➔ Android 5.0

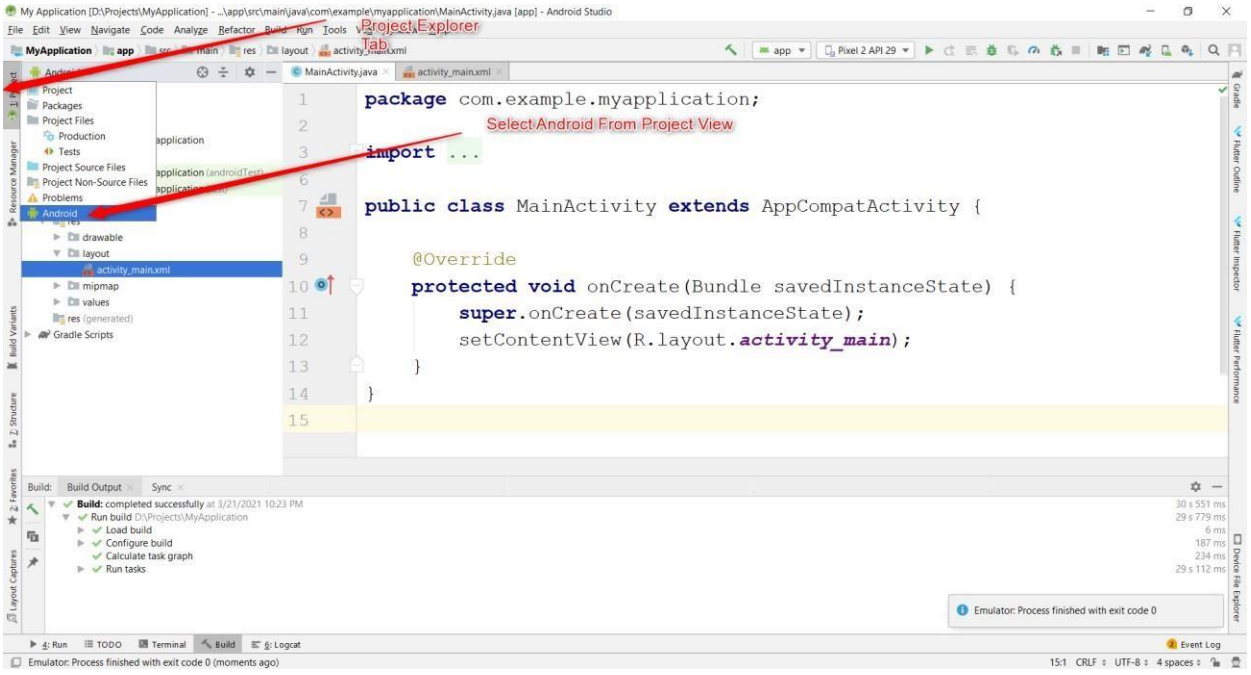
Select Checkbox Use androidx.artifacts folder as below screenshot.



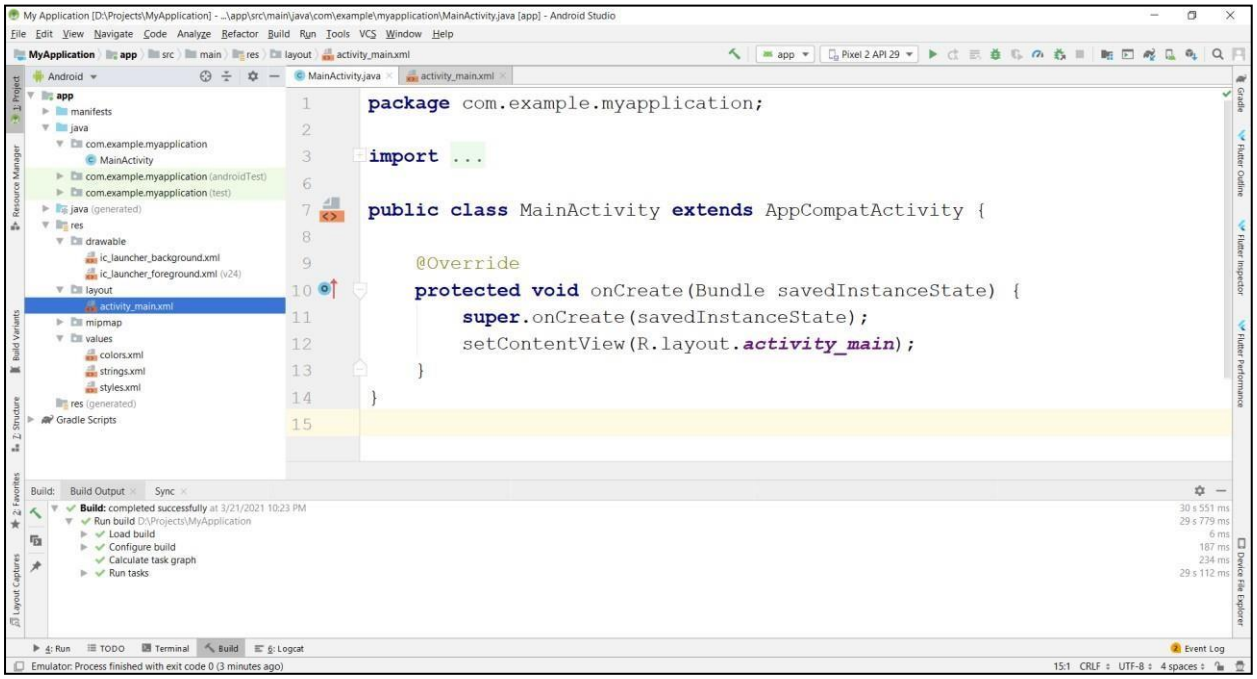
Android Project Structure:



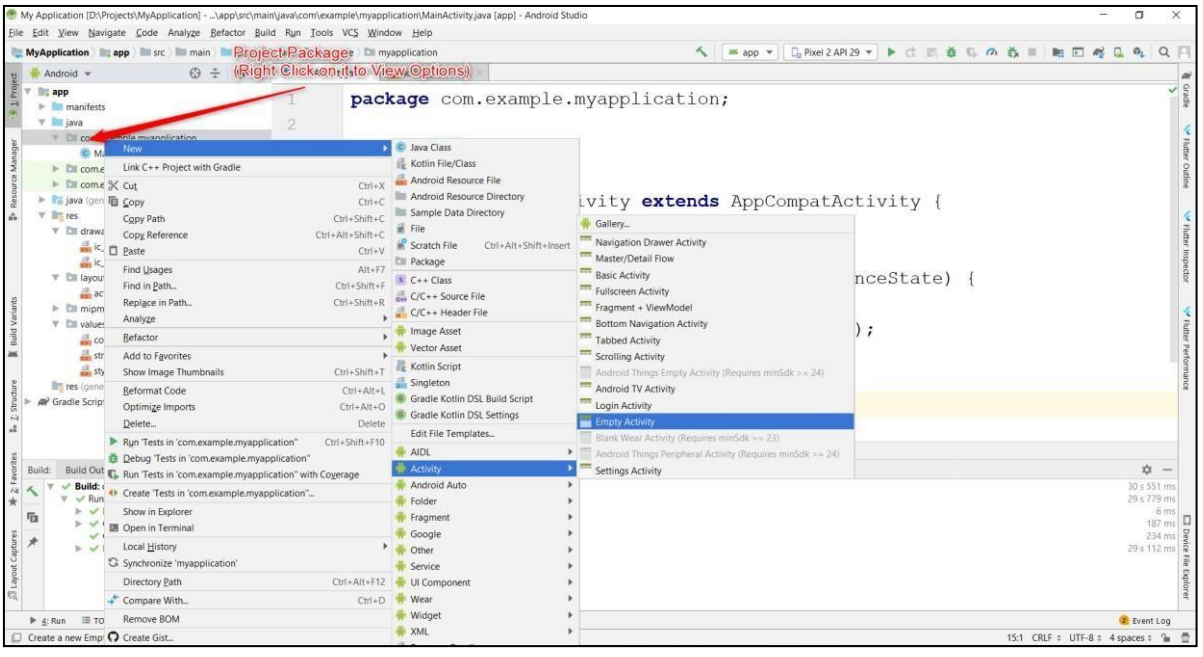
Select Project Explorer and Select Android from Project View



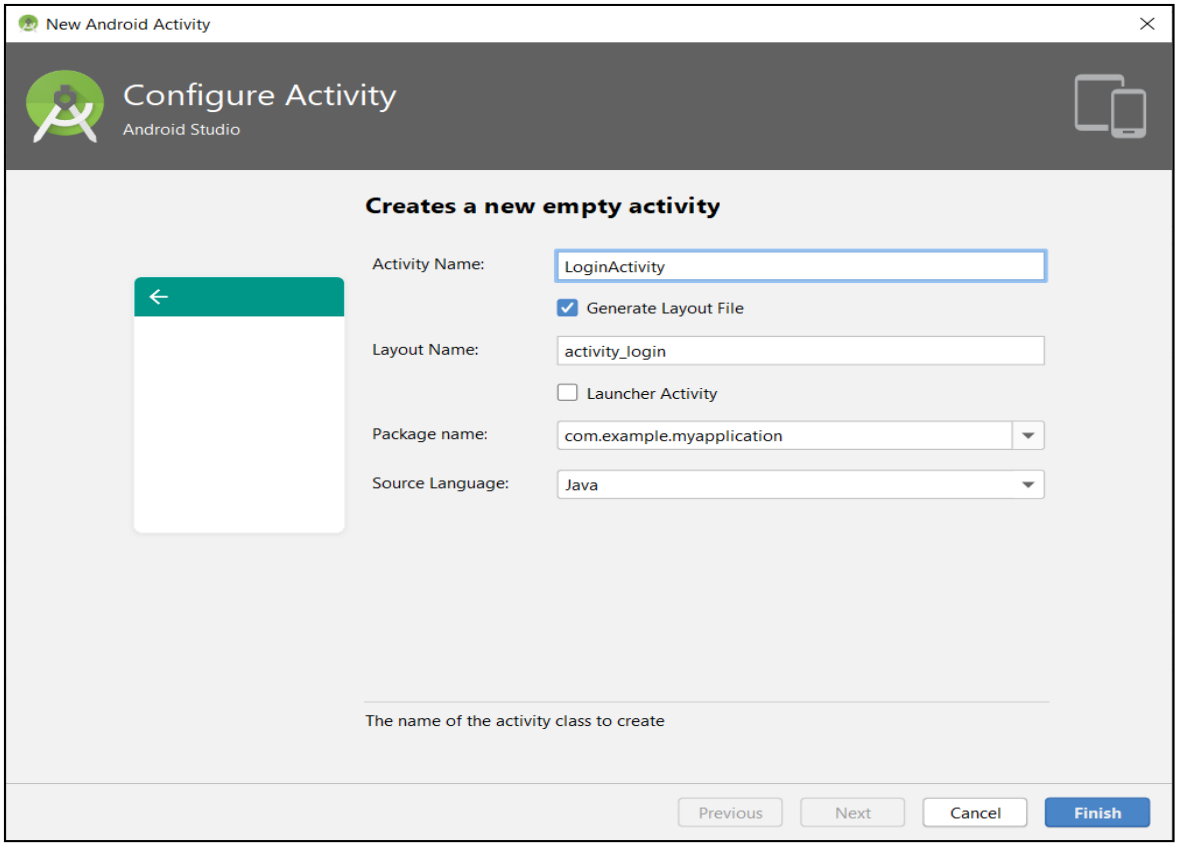
Basic View:

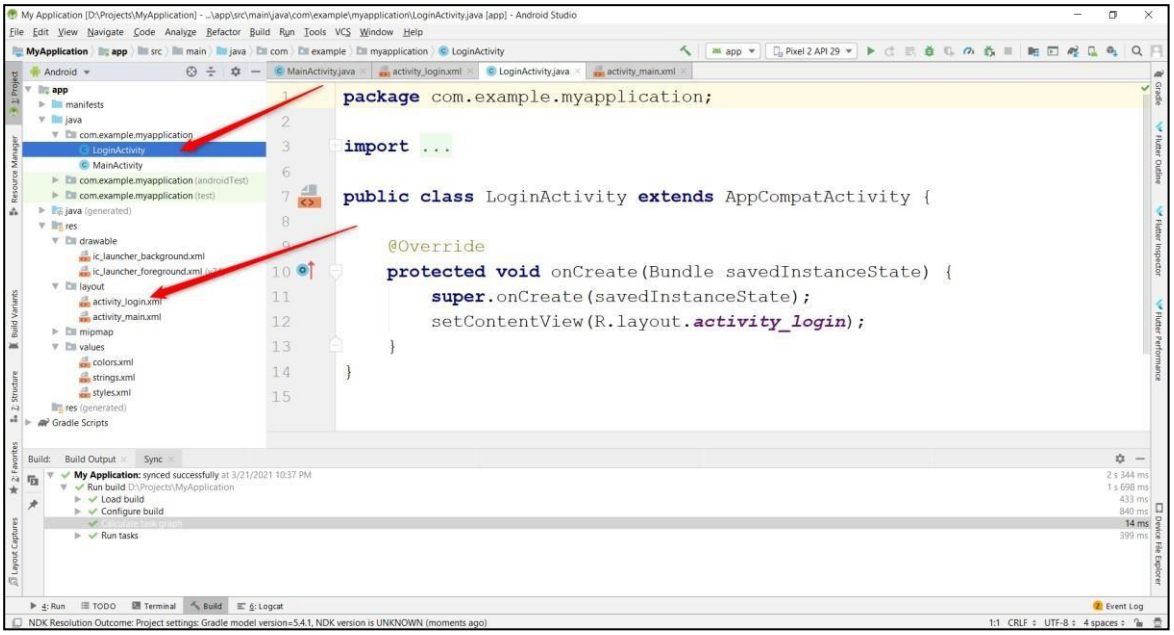


2. Importing an Existing Project in Android Studio
3. Creating an Activity in Android
   Right Click on Package ➔ New ➔ Activity➔ Empty Activity



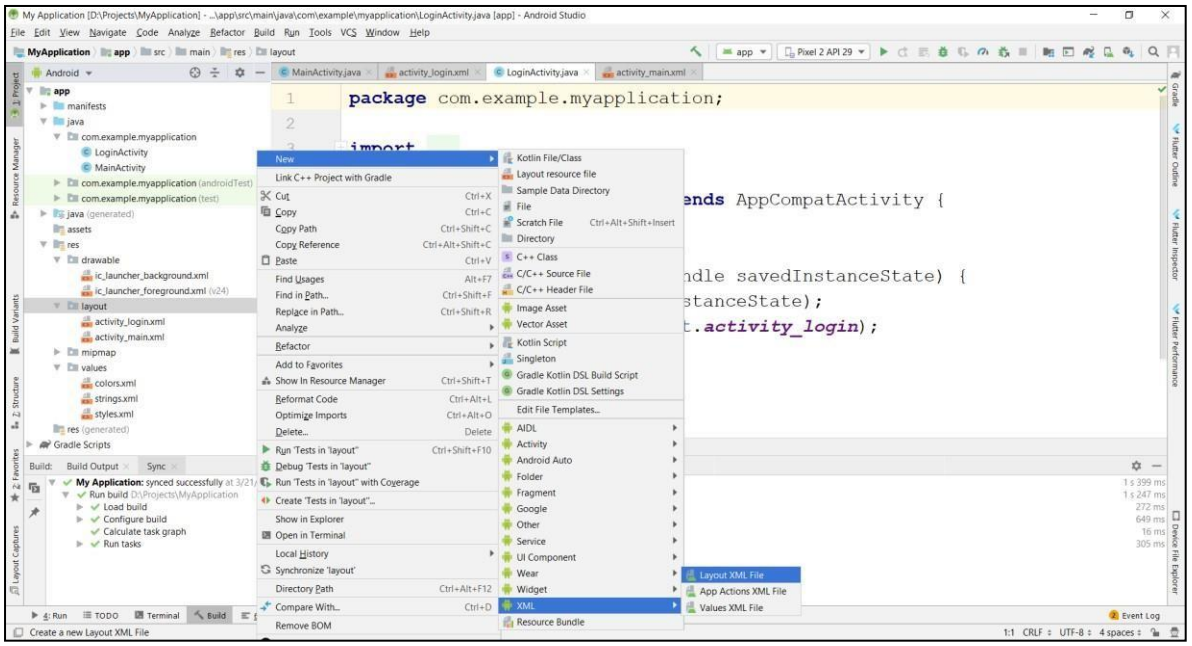
Enter Activity Name and Press Finish


New Android Activity
Configure Activity
Android Studio
Creates a new empty activity
Activity Name:
LoginActivity
Generate Layout File
Layout Name:
activity_login
Launcher Activity
Package name:
com.example.myapplication
Source Language:
Java
The name of the activity class to create
Previous
Next
Cancel
Finish

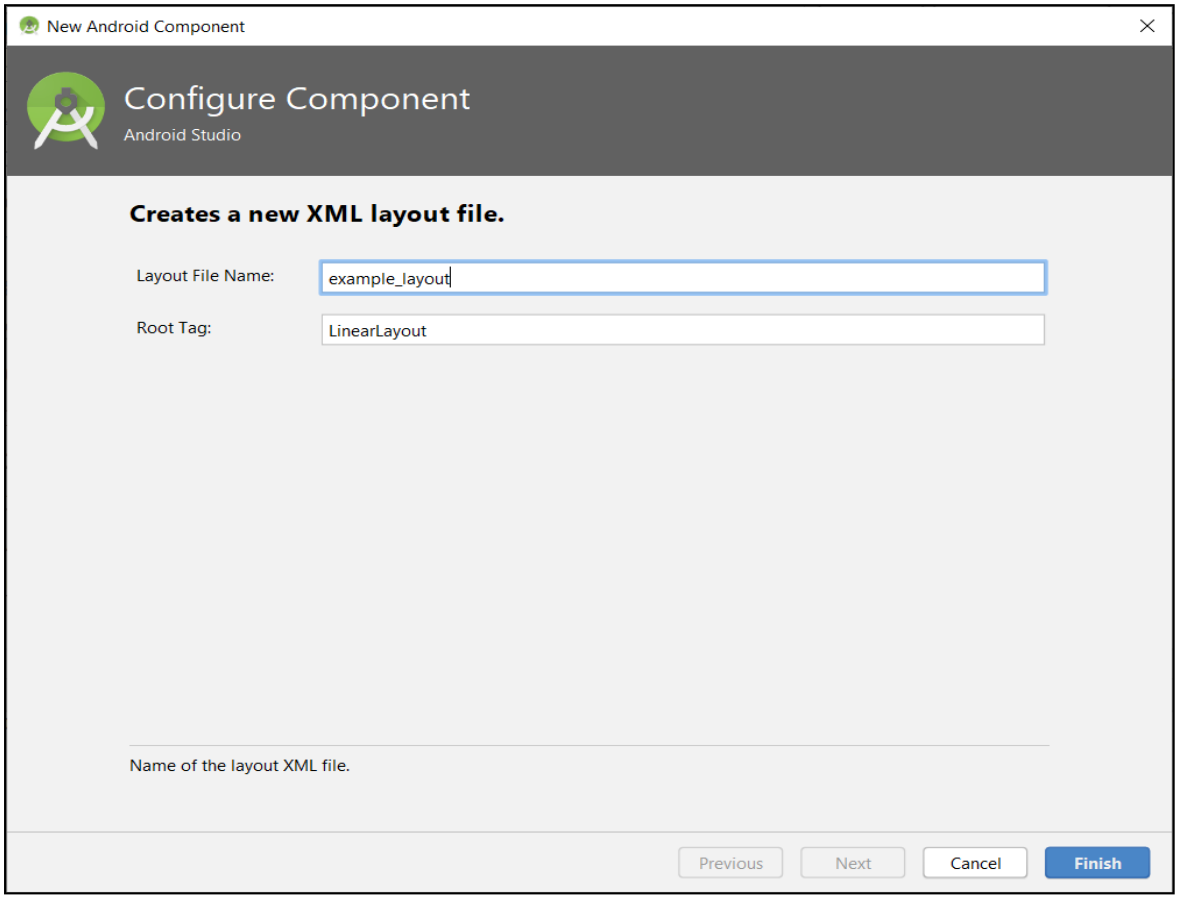

package com.example.myapplication;
import ...
public class LoginActivity extends AppCompatActivity {
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_login);
}
}

Creating a Layout in Android

Right Click on Layout Folder ➔ New ➔ XML➔Layout XML File

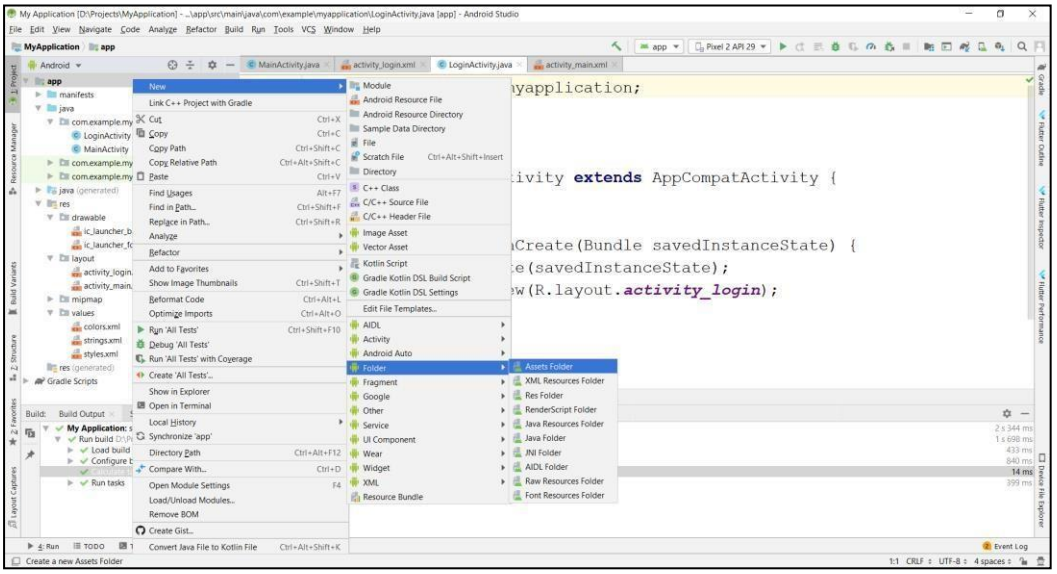


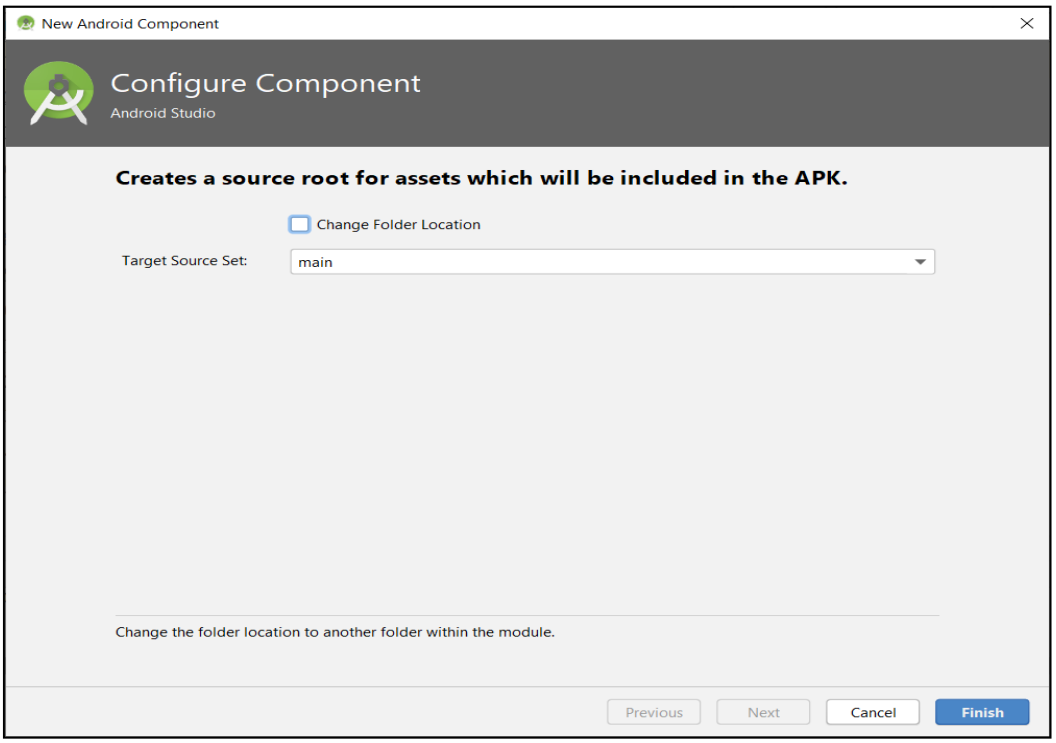
Enter xml file name and press Finish

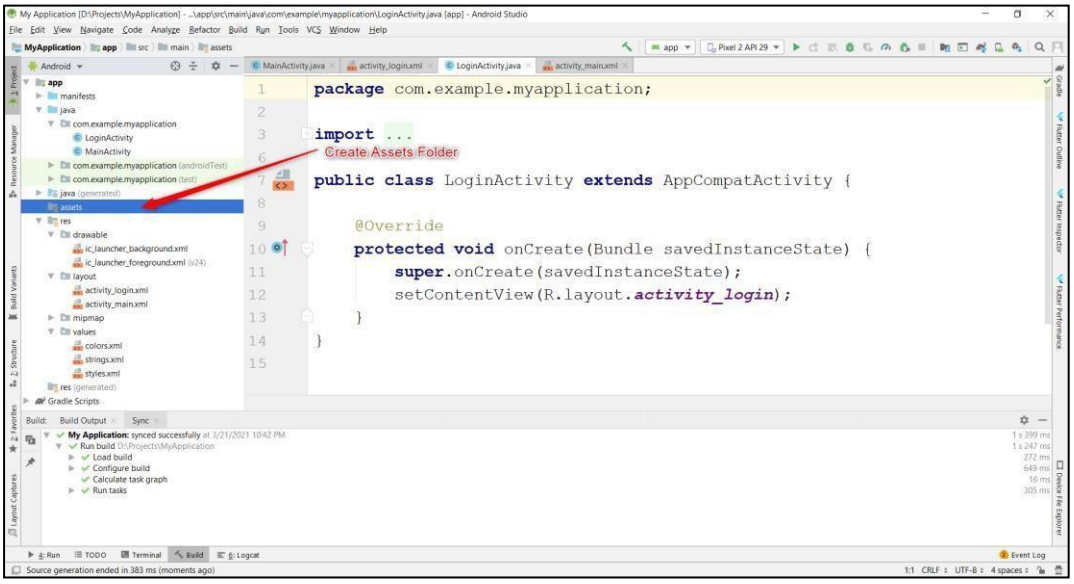


Creating Assets Folder in Android

Right Click on app folder ➔ New ➔ Folder ➔ Assets Folder ➔ Press Finish Button

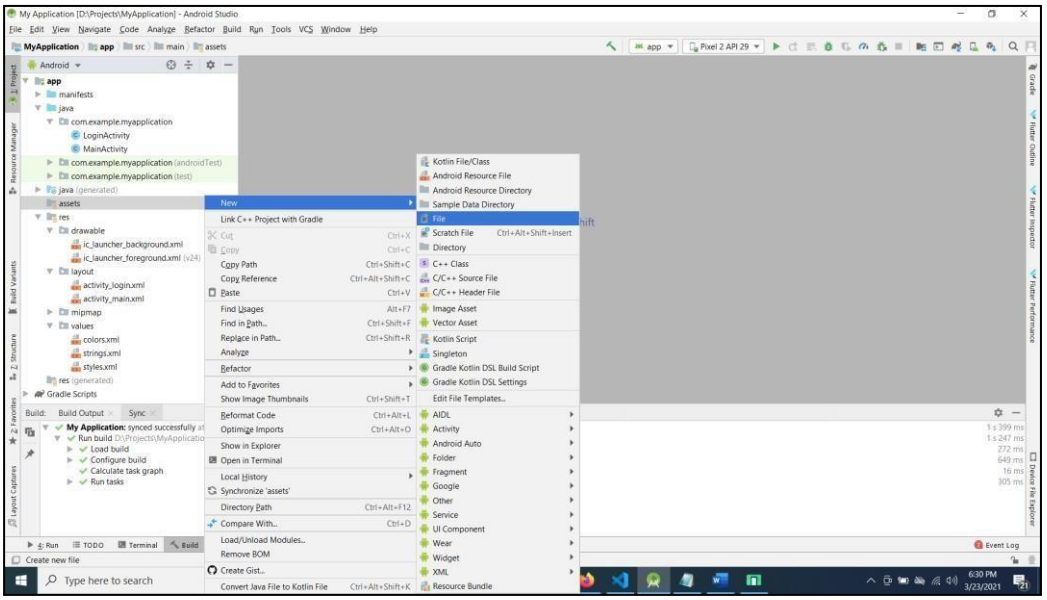




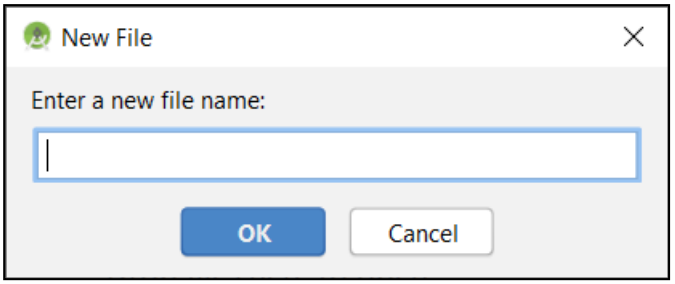


Creating File in assets Folder:

Right Click on assets folder ➔ New ➔ File



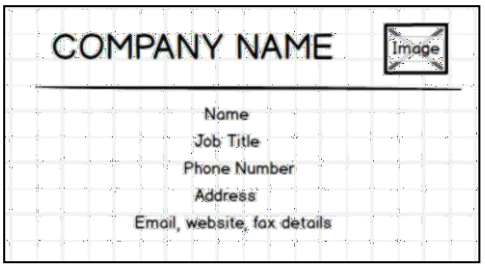
Enter filename with extension (Eg: abc.xml)



Programs
PART A

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.



1. Create a New Android Project with Empty Activity.
2. Open activity_main.xml file from res➔ layout folder, check/add Linear Layout as the root view.
3. Create layout using nested Relative Layout and TextView.
4. Use View background property to draw the line
5. Add Image to drawable folder and reference the image in the layout using @drawable/<image_name>
6. Use android:layout_gravity/android:gravity properties to center the components.

Design

activity_main.xml

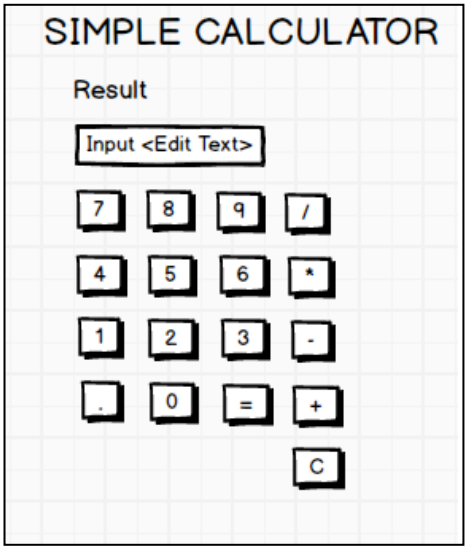
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"

android:orientation="vertical"
android:paddingLeft="20dp"
android:paddingTop="25dp"
android:paddingRight="20dp"
tools:context=".MainActivity">


<RelativeLayout
android:layout_width="match_parent"
android:layout_height="59dp">


<TextView
android:id="@+id/textView"
android:layout_width="wrap_content"
android:layout_height="44dp"
android:layout_alignParentStart="true"
android:layout_alignParentBottom="true"
android:layout_marginStart="31dp"
android:layout_marginLeft="20dp"
android:layout_marginBottom="10dp"
android:gravity="center"
android:text="GLOBAL TECHNOLOGY LTD"
android:textColor="#E61717"
android:textSize="20sp" />

<ImageView
android:id="@+id/imageView4"
android:layout_width="48dp"
android:layout_height="match_parent"
android:layout_alignParentBottom="true"
android:layout_marginLeft="11dp"
android:layout_marginBottom="0dp"
android:layout_toRightOf="@id/textView"
app:srcCompat="@drawable/gat_logo" />

</RelativeLayout>

<View
android:layout_width="match_parent"
android:layout_height="2dp"
android:background="#000000"
/>


<TextView
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Amith"
android:textSize="16dp"
```

```
android:layout_marginBottom="10dp"
android:layout_marginTop="10dp"
android:textColor="#000000"
android:gravity="center"
/>

<TextView
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="software developer"
android:textSize="16dp"
android:layout_marginBottom="10dp"
android:layout_marginTop="10dp"
android:textColor="#000000"
android:gravity="center"
/>
<View
android:layout_width="match_parent"
android:layout_height="2dp"
android:background="#000000"
/>
<TextView
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="+91-91082-75635"
android:textSize="16dp"
android:layout_marginBottom="10dp"
android:layout_marginTop="10dp"
android:textColor="#000000"
android:gravity="center"
/>

<TextView
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Bangalore"
android:textSize="16dp"
android:layout_marginBottom="10dp"
android:layout_marginTop="10dp"
android:textColor="#000000"
android:gravity="center"
/>

<TextView
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Email:info@gat.ac.in, Website:https://gat.ac.in/, Fax:+91-80-
28603158"
android:textSize="16dp"
android:layout_marginBottom="10dp"
android:layout_marginTop="10dp"
android:textColor="#000000"
android:gravity="center"
/>

</LinearLayout>
```

Sample Output

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.



1. Create a New Android Project with Empty Activity.
2. Open activity_main.xml file from res➔ layout folder, check/add Constraint Layout as the root view.
3. Create Layout using Drag and Drop framework.
4. Open MainActivty.java file, Override onCreate() method and bring activity_main.xml file on screen using setContentView() and bring the view references using findViewById() method.
5. Add Listeners to Button Click Event:
6. Create a class which implements OnClickListener interface.
7. Override onClick() method of OnClickListener Interface.
8. Register the button for click event by calling setOnClickListener() method of View class and pass the object of the class that implemented OnClickListener Interface.
9. Create a logic to Add/Subtract/Multiply/Divide to perform arithmetic operation on 2 operands (Eg: 10+20), If more than 2 operands or wrong input, display invalid input messages.

Design

activity_main.xml

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

<Button
android:id="@+id/button_clear"
android:layout_width="87dp"
android:layout_height="53dp"
android:layout_marginTop="30dp"
android:text="C"
app:layout_constraintStart_toStartOf="@+id/button_add"
app:layout_constraintTop_toBottomOf="@+id/button_add" />

<Button
android:id="@+id/button_sub"
android:layout_width="87dp"
android:layout_height="53dp"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="-"
app:layout_constraintStart_toEndOf="@+id/button_three"
app:layout_constraintTop_toBottomOf="@+id/button_mul" />

<Button
android:id="@+id/button_add"
android:layout_width="87dp"
android:layout_height="53dp"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="+"
app:layout_constraintStart_toEndOf="@+id/button_equal"
app:layout_constraintTop_toBottomOf="@+id/button_sub" />

<Button
android:id="@+id/button_mul"
android:layout_width="87dp"
android:layout_height="53dp"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="*"
app:layout_constraintStart_toEndOf="@+id/button_six"
app:layout_constraintTop_toBottomOf="@+id/button_div" />
```

```
<Button
android:id="@+id/button_equal"
android:layout_width="62dp"
android:layout_height="53dp"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="="
app:layout_constraintStart_toEndOf="@+id/button_zero"
app:layout_constraintTop_toBottomOf="@+id/button_three" />

<Button
android:id="@+id/button_zero"
android:layout_width="62dp"
android:layout_height="53dp"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="0"
app:layout_constraintStart_toEndOf="@+id/button_dot"
app:layout_constraintTop_toBottomOf="@+id/button_two" />

<Button
android:id="@+id/button_dot"
android:layout_width="62dp"
android:layout_height="53dp"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="."
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/button_one" />

<Button
android:id="@+id/button_three"
android:layout_width="62dp"
android:layout_height="53dp"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="3"
app:layout_constraintStart_toEndOf="@+id/button_two"
app:layout_constraintTop_toBottomOf="@+id/button_six" />

<Button
android:id="@+id/button_two"
android:layout_width="62dp"
android:layout_height="53dp"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="2"
app:layout_constraintStart_toEndOf="@+id/button_one"
app:layout_constraintTop_toBottomOf="@+id/button_five" />
```

```
<Button
android:id="@+id/button_one"
android:layout_width="62dp"
android:layout_height="53dp"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="1"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/button_four" />

<Button
android:id="@+id/button_six"
android:layout_width="62dp"
android:layout_height="53dp"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="6"
app:layout_constraintStart_toEndOf="@+id/button_five"
app:layout_constraintTop_toBottomOf="@+id/button_nine" />

<Button
android:id="@+id/button_seven"
android:layout_width="62dp"
android:layout_height="53dp"
android:layout_marginStart="20dp"
android:layout_marginTop="20dp"
android:text="7"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/txt_result" />

<Button
android:id="@+id/button_eight"
android:layout_width="62dp"
android:layout_height="53dp"
android:layout_marginStart="20dp"
android:layout_marginTop="20dp"
android:text="8"
app:layout_constraintStart_toEndOf="@+id/button_seven"
app:layout_constraintTop_toBottomOf="@+id/txt_result" />

<Button
android:id="@+id/button_nine"
android:layout_width="62dp"
android:layout_height="53dp"
android:layout_marginStart="20dp"
android:layout_marginTop="20dp"
android:text="9"
app:layout_constraintStart_toEndOf="@+id/button_eight"
app:layout_constraintTop_toBottomOf="@+id/txt_result" />
```

```
<Button
android:id="@+id/button_four"
android:layout_width="62dp"
android:layout_height="53dp"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="4"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/button_seven" />

<TextView
android:id="@+id/textView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="30dp"
android:text="SIMPLE CALCULATOR"
android:textSize="26dp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />

<TextView
android:id="@+id/textView2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="20dp"
android:layout_marginTop="20dp"
android:text="Result"
android:textSize="18dp"
android:textStyle="bold"
app:layout_constraintEnd_toStartOf="@+id/textView"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView" />

<EditText
android:id="@+id/txt_result"
android:layout_width="310dp"
android:layout_height="46dp"
android:layout_marginTop="20dp"
android:ems="10"
android:inputType="textPersonName"
app:layout_constraintStart_toStartOf="@+id/textView2"
app:layout_constraintTop_toBottomOf="@+id/textView2" />

<Button
android:id="@+id/button_div"
android:layout_width="87dp"
android:layout_height="53dp"
android:layout_marginStart="20dp"
android:layout_marginTop="20dp"
```

```
android:text="/"
app:layout_constraintStart_toEndOf="@+id/button_nine"
app:layout_constraintTop_toBottomOf="@+id/txt_result" />

<Button
android:id="@+id/button_five"
android:layout_width="62dp"
android:layout_height="53dp"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="5"
app:layout_constraintStart_toEndOf="@+id/button_four"
app:layout_constraintTop_toBottomOf="@+id/button_eight" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.partaprogram2;

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;

import java.util.regex.Pattern;

public class MainActivityextendsAppCompatActivityimplementsView.OnClickListener {
    Button btnOne, btnTwo, btnThree, btnFour, btnFive, btnSix;
    Button btnSeven, btnEight, btnNine, btnZero;
    Button btnAdd,btnSub,btnMul,btnDiv;
    Button btnClear,btnEqual,btnDot;

EditTexttxtResult;

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

btnOne=(Button)findViewById(R.id.button_one);
btnOne.setOnClickListener(this);

btnTwo=(Button)findViewById(R.id.button_two);
btnTwo.setOnClickListener(this);

btnThree=(Button)findViewById(R.id.button_three);
btnThree.setOnClickListener(this);
```

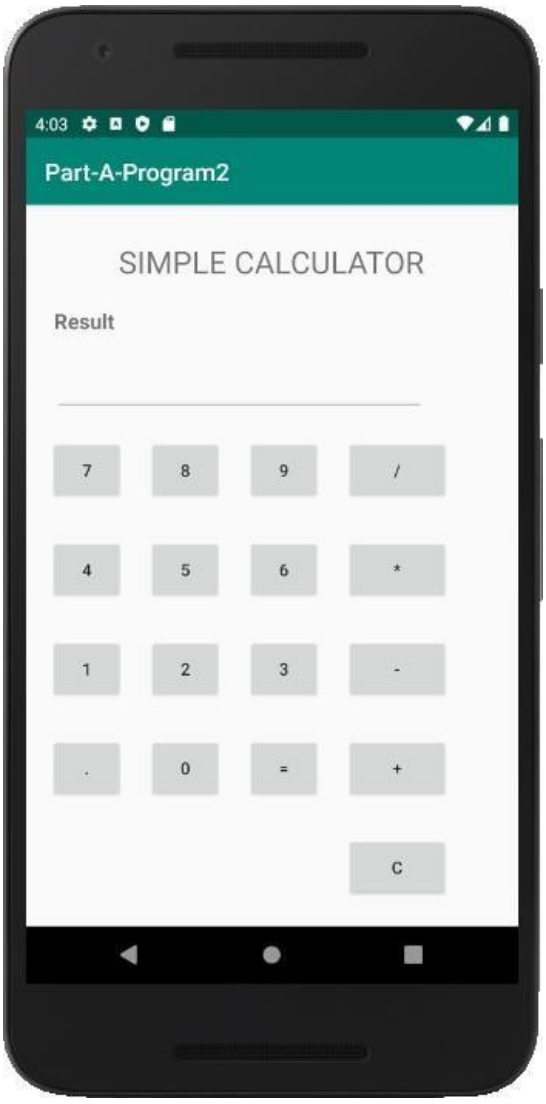
```
btnFour=(Button)findViewById(R.id.button_four);
btnFour.setOnClickListener(this);

btnFive=(Button)findViewById(R.id.button_five);
btnFive.setOnClickListener(this);

btnSix=(Button)findViewById(R.id.button_six);
btnSix.setOnClickListener(this);

btnSeven=(Button)findViewById(R.id.button_seven);
btnSeven.setOnClickListener(this);

btnEight=(Button)findViewById(R.id.button_eight);
btnEight.setOnClickListener(this);

btnNine=(Button)findViewById(R.id.button_nine);
btnNine.setOnClickListener(this);

btnZero=(Button)findViewById(R.id.button_zero);
btnZero.setOnClickListener(this);


btnAdd=(Button)findViewById(R.id.button_add);
btnAdd.setOnClickListener(this);

btnSub=(Button)findViewById(R.id.button_sub);
btnSub.setOnClickListener(this);

btnMul=(Button)findViewById(R.id.button_mul);
btnMul.setOnClickListener(this);

btnDiv=(Button)findViewById(R.id.button_div);
btnDiv.setOnClickListener(this);


btnClear=(Button)findViewById(R.id.button_clear);
btnClear.setOnClickListener(this);

btnEqual=(Button)findViewById(R.id.button_equal);
btnEqual.setOnClickListener(this);

btnDot=(Button)findViewById(R.id.button_dot);
btnDot.setOnClickListener(this);

txtResult=(EditText)findViewById(R.id.txt_result);
txtResult.setText("");
}
```

```
public void onClick(View v)
{
if(v.equals(btnOne))
txtResult.append("1");
if(v.equals(btnTwo))
txtResult.append("2");
if(v.equals(btnThree))
txtResult.append("3");
if(v.equals(btnFour))
txtResult.append("4");
if(v.equals(btnFive))
txtResult.append("5");
if(v.equals(btnSix))
txtResult.append("6");
if(v.equals(btnSeven))
txtResult.append("7");
if(v.equals(btnEight))
txtResult.append("8");
if(v.equals(btnNine))
txtResult.append("9");
if(v.equals(btnZero))
txtResult.append("0");
if(v.equals(btnDot))
txtResult.append(".");
if(v.equals(btnClear))
txtResult.setText("");

if(v.equals(btnEqual))
{
try {

String data = txtResult.getText().toString();
if (data.contains("/")) {
String[] operands = data.split("/");
if(operands.length==2) {
double operand1 = Double.parseDouble(operands[0]);
double operand2 = Double.parseDouble(operands[1]);
double result = operand1 / operand2;
txtResult.setText(String.valueOf(result));
                    }
else
{
Toast.makeText(getBaseContext(),"Invalid Input",
Toast.LENGTH_LONG).show();
}

}
else if (data.contains("*")) {
String[] operands = data.split(Pattern.quote("*"));
if(operands.length==2) {
double operand1 = Double.parseDouble(operands[0]);
double operand2 = Double.parseDouble(operands[1]);
```

```
double result = operand1 * operand2;
txtResult.setText(String.valueOf(result));
}
else
{
Toast.makeText(getBaseContext(),"Invalid Input",
Toast.LENGTH_LONG).show();
}

}
else if (data.contains("+")) {
String[] operands = data.split(Pattern.quote("+"));
if(operands.length==2) {
double operand1 = Double.parseDouble(operands[0]);
double operand2 = Double.parseDouble(operands[1]);
double result = operand1 + operand2;
txtResult.setText(String.valueOf(result));
}
else
{
Toast.makeText(getBaseContext(),"Invalid Input",
Toast.LENGTH_LONG).show();
}
}
else if (data.contains("-")) {
String[] operands = data.split("-");
if(operands.length==2) {
double operand1 = Double.parseDouble(operands[0]);
double operand2 = Double.parseDouble(operands[1]);
double result = operand1 - operand2;
txtResult.setText(String.valueOf(result));
}
else
{
Toast.makeText(getBaseContext(),"Invalid Input",
Toast.LENGTH_LONG).show();
}
}
}

catch(Exception e) {
Toast.makeText(getBaseContext(),"Invalid Input",
Toast.LENGTH_LONG).show();
}
}
```

```
if(v.equals(btnAdd))
txtResult.append("+");
if(v.equals(btnSub))
txtResult.append("-");
if(v.equals(btnMul))
txtResult.append("*");
if(v.equals(btnDiv))
txtResult.append("/");
}
}
```

Sample Output

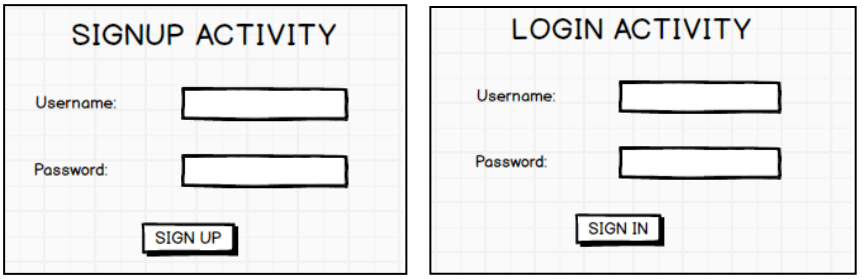


Program 3

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

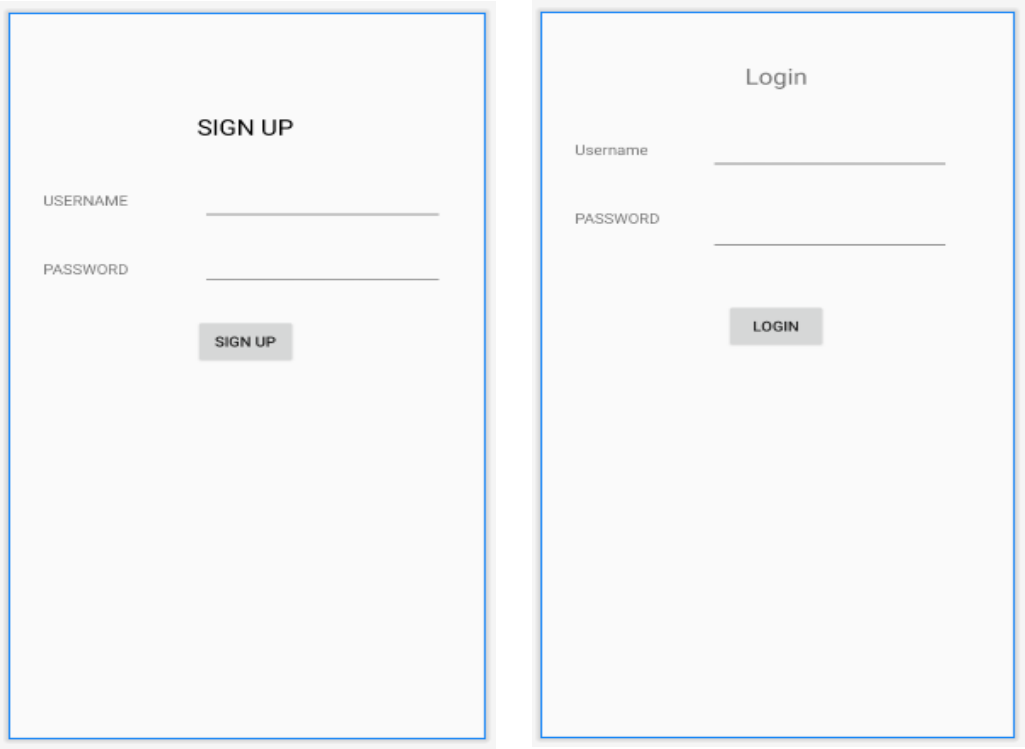
- Password should contain uppercase and lowercase letters.
- Password should contain letters and numbers.
- Password should contain special characters.
- Minimum length of the password (the default value is 8).

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.



1. Create a New Android Project with Empty Activity.
2. Open activity_main.xml file from res➔ layout folder, check/add Constraint Layout as the root view.
3. Create Signup Layout using Drag and Drop framework design the layout.
4. Create One more Empty Activity LoginActivity using Android Studio Create Activity Flow (Refer Android Studio Tutorial)
5. Open activity_login.xml file from res➔layout folder, check/add Constraint Layout as the root view.
6. Create Login Layout using Drag and Drop framework.
7. Add Listeners to Button Click Event:
   - Create a class which implements OnClickListener interface.
   - Override onClick() method of OnClickListener Interface.
   - Register the button for click event by calling setOnClickListener() method of View class and pass the object of the class that implemented OnClickListener Interface.
8. Use Regular Expression**"^(?=.*[A-Z])(?=.*[a-z])(?=.*\\d)(?=.*[@$!])[A-Za-z\\d@$!]{8,}$"** to validate the password.

Design



Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayoutxmlns:android="http://schemas.andr
oid.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/textView2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="100dp"
android:text="SIGN UP"
android:textColor="@android:color/background_dark"
android:textSize="22dp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />

<TextView
android:id="@+id/textView3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
```

```
android:layout_marginStart="30dp"
android:layout_marginTop="50dp"
android:text="USERNAME"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView2" />

<TextView
android:id="@+id/textView4"
android:layout_width="82dp"
android:layout_height="34dp"
android:layout_marginTop="50dp"
android:text="PASSWORD"
app:layout_constraintStart_toStartOf="@+id/textView3"
app:layout_constraintTop_toBottomOf="@+id/textView3" />

<EditText
android:id="@+id/txt_username"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="40dp"
android:layout_marginEnd="10dp"
android:ems="10"
android:inputType="textPersonName"
app:layout_constraintBottom_toBottomOf="@+id/textView3"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toEndOf="@+id/textView3"
app:layout_constraintTop_toTopOf="@+id/textView3" />

<EditText
android:id="@+id/txt_password"
android:layout_width="0dp"
android:layout_height="40dp"
android:layout_marginTop="26dp"
android:ems="10"
android:inputType="textPassword"
app:layout_constraintEnd_toEndOf="@+id/txt_username"
app:layout_constraintStart_toStartOf="@+id/txt_username"
app:layout_constraintTop_toBottomOf="@+id/txt_username" />

<Button
android:id="@+id/btn_signup"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="30dp"
android:text="Sign Up"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/txt_password" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Activity_login.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayoutxmlns:android="http://schemas.andr
oid.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=".LoginActivity">

<TextView
android:id="@+id/textView7"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="50dp"
android:text="Login"
android:textSize="22dp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />

<TextView
android:id="@+id/textView9"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="30dp"
android:layout_marginTop="50dp"
android:text="Username"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView7" />

<EditText
android:id="@+id/txt_login_username"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="20dp"
android:layout_marginEnd="20dp"
android:ems="10"
android:inputType="textPersonName"
app:layout_constraintBottom_toBottomOf="@+id/textView9"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toEndOf="@+id/textView9"
app:layout_constraintTop_toTopOf="@+id/textView9" />

<TextView
android:id="@+id/textView10"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="30dp"
android:layout_marginTop="50dp"
```

```
android:text="PASSWORD"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView9" />

<EditText
android:id="@+id/txt_login_password"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:ems="10"
android:inputType="textPassword"
app:layout_constraintEnd_toEndOf="@+id/txt_login_username"
app:layout_constraintStart_toStartOf="@+id/txt_login_username"
app:layout_constraintTop_toTopOf="@+id/textView10" />

<Button
android:id="@+id/btn_login_signin"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="50dp"
android:text="Login"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/txt_login_password" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.parta.program3;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class MainActivityextends AppCompatActivityimplements View.OnClickListener {

EditTexttxtUsername;
EditTexttxtPassword;

Button btnSignup;

String regularExpression="^(?=.*[A-Z])(?=.*[a-z])(?=.*\\d)(?=.*[@$!])[A-Za-
z\\d@$!]{8,}$";
```

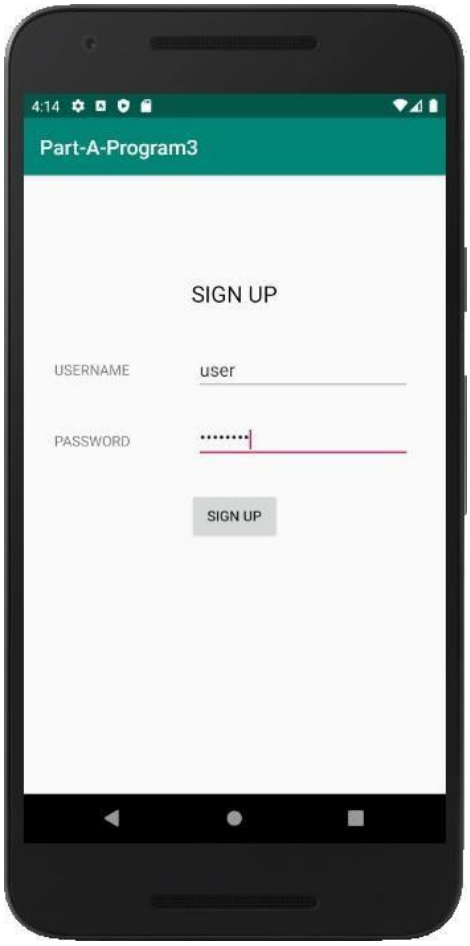
```
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
txtUsername=(EditText)findViewById(R.id.txt_username);
txtPassword=(EditText)findViewById(R.id.txt_password);

btnSignup=(Button)findViewById(R.id.btn_signup);
btnSignup.setOnClickListener(this);

}

public void onClick(View v)
{
     String username=txtUsername.getText().toString();
     String password=txtPassword.getText().toString();

      if(validatePassword(password))  {
      Bundle bundle = new Bundle();
      bundle.putString("user", username);
      bundle.putString("Lab@2018", password);

      Intent it = new Intent(this, LoginActivity.class);
      it.putExtra("data", bundle);

      startActivity(it);
      }
else
{
Toast.makeText(getBaseContext(), "Invalid Password",
Toast.LENGTH_LONG).show();
}
}

public booleanvalidatePassword(String password)
    {
       Pattern pattern= Pattern.compile(regularExpression);
        Matcher matcher=pattern.matcher(password);
return matcher.matches();

    }
}
```

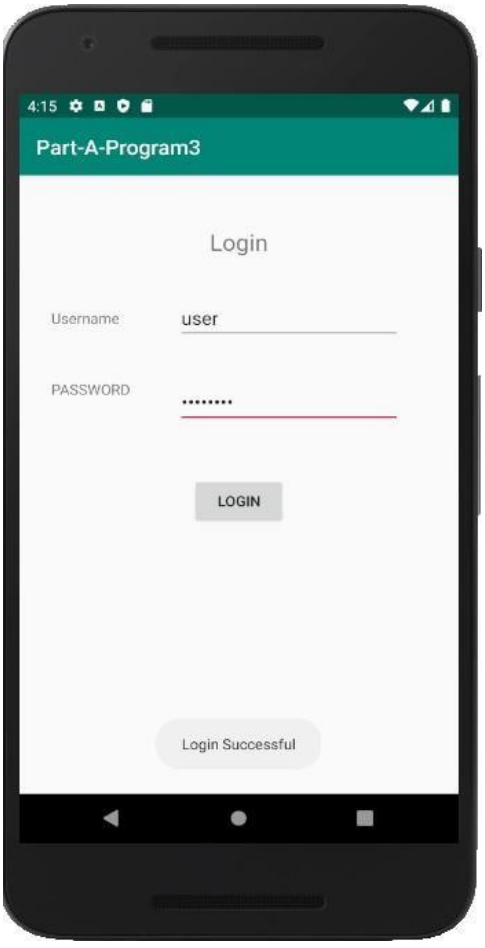
LoginActivity.java

```
package com.example.parta.program3;

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 LoginActivityextends AppCompatActivityimplements View.OnClickListener {

EditTexttxtLoginUsername;
EditTexttxtLoginPassword;
    Button btnLogin;

    String user,pass;

int count=0;

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

txtLoginUsername=(EditText)
findViewById(R.id.txt_login_username);
txtLoginPassword=(EditText)
findViewById(R.id.txt_login_password);

btnLogin=(Button)findViewById(R.id.btn_login_signin);
btnLogin.setOnClickListener(this);

        Bundle bundle=getIntent().getBundleExtra("data");
user=bundle.getString("user");
pass=bundle.getString("Lab@2018");

   }

public void onClick(View v)
    {
        String user1=txtLoginUsername.getText().toString();
        String pass1=txtLoginPassword.getText().toString();

if(user.equals(user1)&&pass.equals(pass1))
        {
Toast.makeText(this,"Login Successful"
,Toast.LENGTH_LONG).show();
        }
else
```

```
{
count++;
if(count==3)
            {
btnLogin.setEnabled(false);
Toast.makeText(this,
"Failed Login Attempts"
,Toast.LENGTH_LONG).show();
            }
else
{
Toast.makeText(this,"Login Failed "+count
,Toast.LENGTH_LONG).show();
            }
        }

    }
}
```

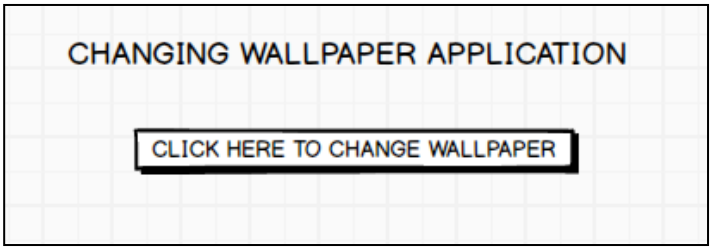
Sample Output





Program 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.



1. Create a New Android Project with Empty Activity.
2. Open activity_main.xml file from res➔ layout folder, check/add LinearLayout as the root view.
3. Create the layout
4. Add 3 or More images to drawable folder (res➔drawable)
5. Declare uses permission android.permission.SET_WALLPAPPER in the AndroidManifest.xml file
6. Schedule Timer task to change the wallpaper on every 30 seconds interval.
7. Initialize and use WallpaperManager.setBitmap() method to change the wallpaper.

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayoutxmlns: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"
android:gravity="center"
tools:context=".MainActivity">

<Button
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Click here to Change Wallpaper"
android:id="@+id/btn_start_change_wallpaper"/>

</LinearLayout>
```

MainActivity.java

```
package com.example.program4;

import androidx.appcompat.app.AppCompatActivity;

import android.app.WallpaperManager;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

import java.util.Timer;
import java.util.TimerTask;

public class MainActivityextends AppCompatActivityimplements View.OnClickListener{

Button btnChangeWallpaper;
booleanrunning;
int[] imagesArray=new int[]{

R.drawable.img1,
R.drawable.img2,
R.drawable.img3,
R.drawable.img4,
R.drawable.img5,
R.drawable.img6,
R.drawable.img7,
R.drawable.img8,
R.drawable.img9,
R.drawable.img10,
R.drawable.img11,
R.drawable.img12
};

int i=0;

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

btnChangeWallpaper=(Button)
findViewById(R.id.btn_start_change_wallpaper);
btnChangeWallpaper.setOnClickListener(this);

}
```

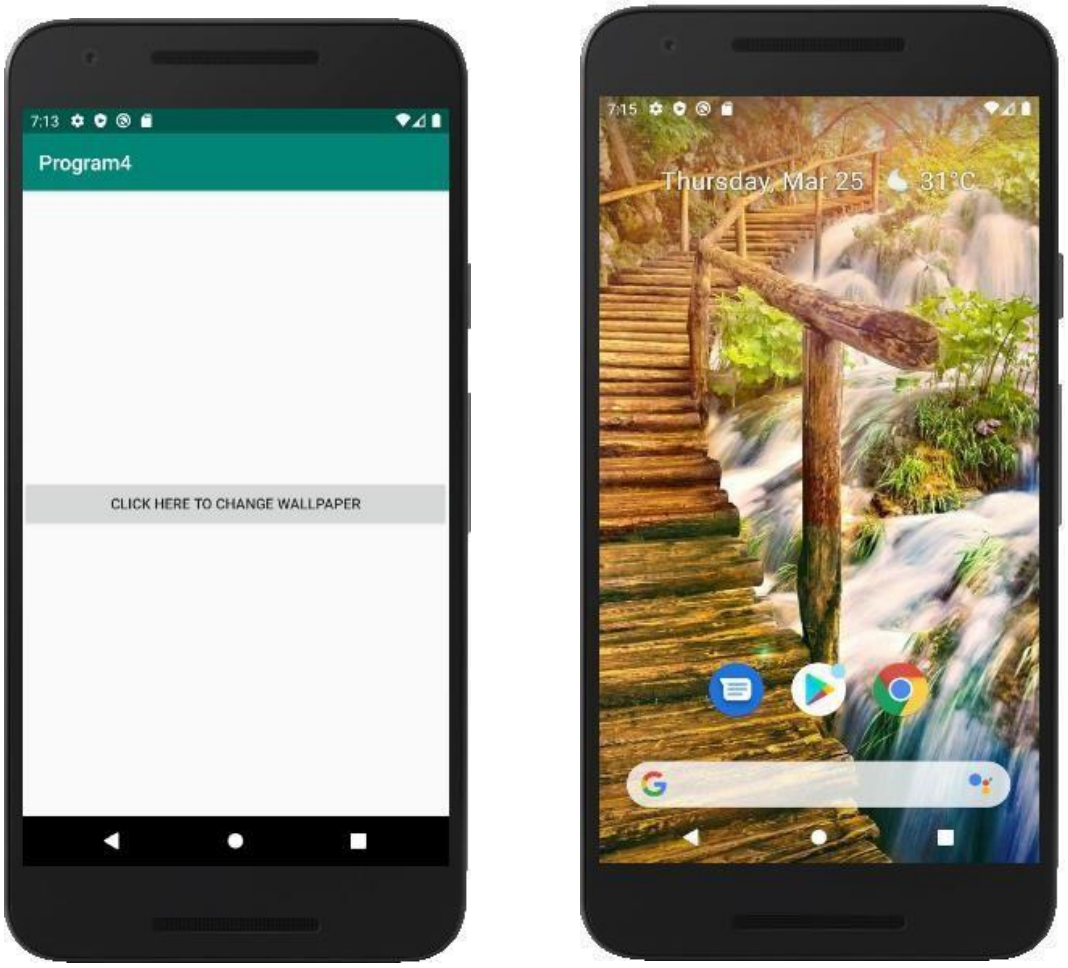
```
public void onClick(View v)
{
   if(!running)
   {
     new Timer().schedule (new MyTimer(),0,3000);
     running=true;
    }
}

class MyTimerextends TimerTask
{
public void run()
{
try
{
WallpaperManagerwallpaperManager=
WallpaperManager.getInstance(getBaseContext());

if(i==12)
i=1;

if(i==11)
i=2;

if(i==10)
i=3;

if(i==9)
i=4;

if(i==8)
i=5;

if(i==7)
i=6;

if(i==6)
i=7;

if(i==5)
i=8;

if(i==4)
i=9;

if(i==3)
i=10;
```

```
wallpaperManager.setBitmap (BitmapFactory.decodeResource(getResources()
                                ,imagesArray[i]));
i++;
            }
catch(Exception e)
            {

            }

        }
    }

}
```

AndriodManifest.xml

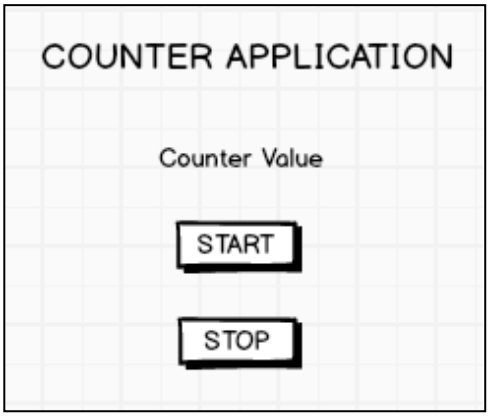
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.program4">

<uses-permission android:name="android.permission.SET_WALLPAPER"/>

<application
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:roundIcon="@mipmap/ic_launcher_round"
android:supportsRtl="true"
android:theme="@style/AppTheme">
<activity android:name=".MainActivity">
<intent-filter>
<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
</application>

</manifest>
```

Sample Output



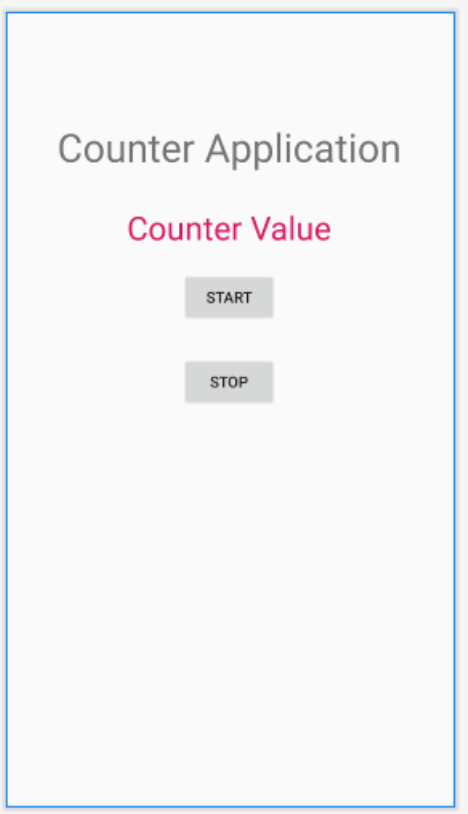
Program 5

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 TextViewcontrol.



1. Create a New Android Project with Empty Activity.
2. Open activity_main.xml file from res➔ layout folder, check/add ConstraintLayout as the root view.
3. Create the layout design using Drag and Drop framework.
4. Add Listeners to Button Click Event:
   - Create a class which implements OnClickListener interface.
   - Override onClick() method of OnClickListener Interface.
   - Register the button for click event by calling setOnClickListener() method of View class and pass the object of the class that implemented OnClickListener Interface.
5. Create a Thread to start the counter logic.
6. Steps to Create a Thread
   - Create a class that extends Thread Class.
   - Override run method of Thread Class.
   - Use start() method of thread class to start the thread.
7. Create Handler class to receive message from child thread, Handler executes in Main Thread.
8. Steps to Create Handler
   - Create Object of type Handler.
   - OverridhandleMessage() of handler class.
9. Pass the counter value to be displayed to the handler.
10. Update the UI to display the counter value received from thread.

Design



activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayoutxmlns:android="http://schemas.andr
oid.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/lbl_counter"
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_marginTop="100dp"
android:text="Counter Application"
android:textSize="36sp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />

<TextView
android:id="@+id/lbl_text"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="30dp"
android:text="Counter Value"
android:textColor="@color/colorAccent"
android:textSize="30sp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView" />

<Button
android:id="@+id/btn_start"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="20dp"
android:text="Start"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/lbl_text" />

<Button
android:id="@+id/btn_stop"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="30dp"
android:text="Stop"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/btn_start" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

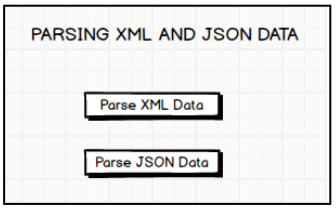
```
package com.example.program5;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import org.w3c.dom.Text;

public class MainActivityextendsAppCompatActivityimplementsView.OnClickListener {

TextViewlblCounter;
Button btnStart,btnStop;

int counter=0;
booleanrunning=false;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
lblCounter=(TextView)findViewById(R.id.lbl_text);
btnStart=(Button)findViewById(R.id.btn_start);
btnStop=(Button)findViewById(R.id.btn_stop);
btnStop.setOnClickListener(this);
btnStart.setOnClickListener(this);
}

public void onClick(View v)
{
if(v.equals(btnStart))
{
   counter=0;
   running=true;
   new MyCounter().start();
  }
  else if(v.equals(btnStop))
  {
    running=false;
  }
}

Handler handler=new Handler()
{
public void handleMessage(Message m)
{
   lblCounter.setText(String.valueOf(m.what));
}

};
```

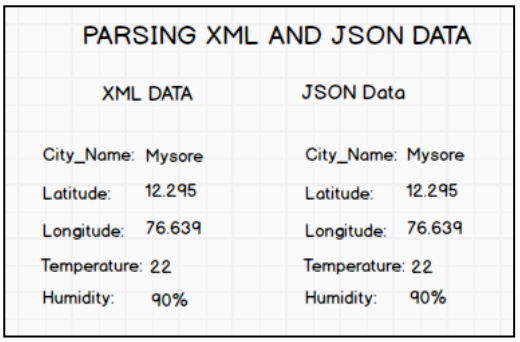
```
class MyCounterextendsThread
{
  public void run()
  {
    while(running)
    {
     counter++;
     handler.sendEmptyMessage(counter);

     try {
    Thread.sleep(1000);
    }
    catch(Exception e) {  }
      }
   }
 }
}
```

Sample Output

Program 6

Create two files of XML and JSON type with values for City_Name, Latitude, Longitude, Temperature, and Humidity. 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.





1. Create a New Android Project with Empty Activity.
2. Open activity_main.xml file from res➔ layout folder, check/add ConstraintLayout as the root view.
3. Create the layout design using Drag and Drop framework.
4. Add Listeners to Button Click Event:
   - Create a class which implements OnClickListener interface.
   - Override onClick() method of OnClickListener Interface.
   - Register the button for click event by calling setOnClickListener() method of View class and pass the object of the class that implemented OnClickListener Interface.
5. Create assets folder (Refer Section Android Studio Tutorial)
6. Create **input.xml** file inside assets folder and paste the below Xml Data

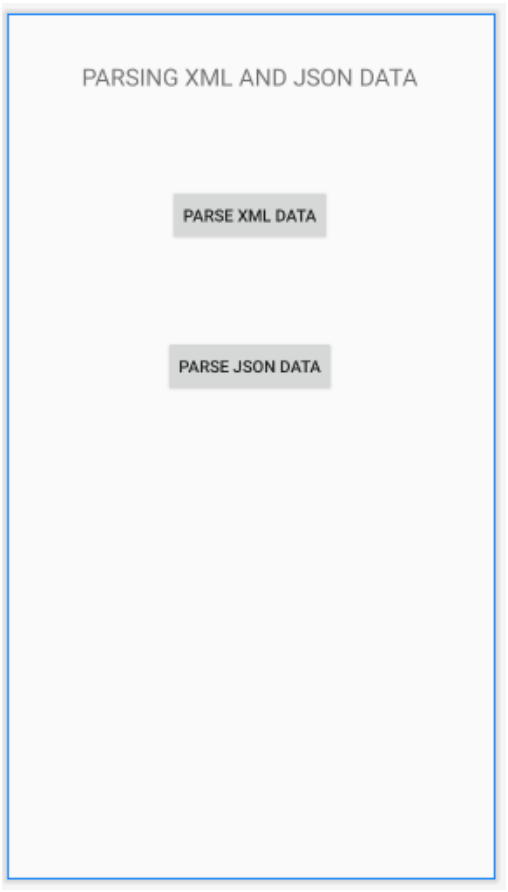
```
<?xml version="1.0"?>
<records>
<employee>

<city_name>Mysore</city_name>
<Latitude>12.295</Latitude>
<Longitude>76.639</Longitude>
<Temperature>22</Temperature>
<Humidity>90%</Humidity>
</employee>
</records>
```

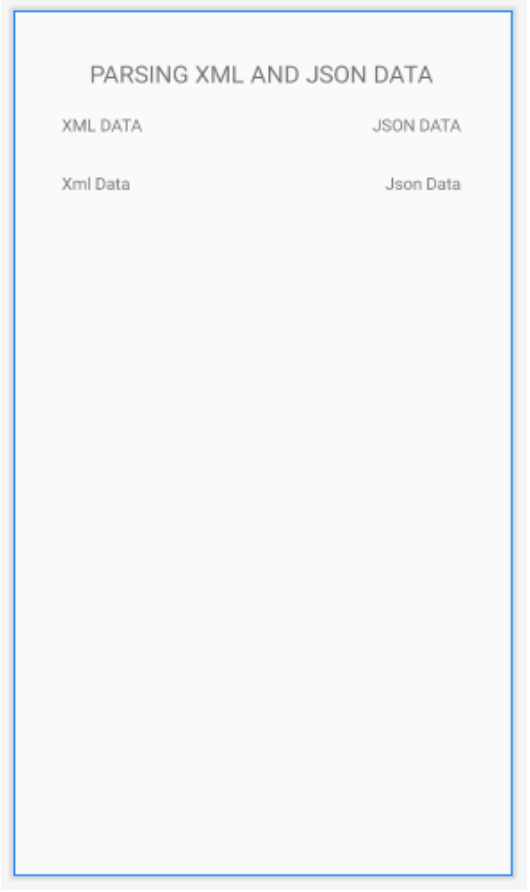
7. Create **input.json** file inside assets folder and paste the below Json Data

```
{
"employee": {
"city_name": "Mysore",
"Latitude": "12.295",
"Longitude": "76.639",
"Temperature": 22,
"Humidity": "90%"
}
}
```

8. Read the XML and Json Data in the files and display on screen

Design





activity_main.xml

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

<Button
android:id="@+id/btn_parsexml"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="80dp"
android:text="Parse XML Data"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView4" />
```

```
<Button
android:id="@+id/btn_parsejson"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="80dp"
android:text="Parse Json Data"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/btn_parsexml" />

<TextView
android:id="@+id/textView4"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="40dp"
android:text="PARSING XML AND JSON DATA"
android:textSize="20dp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

activity_view.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayoutxmlns:android="http://schemas.andr
oid.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=".ViewActivity">

<TextView
android:id="@+id/lbl_xml_data"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="30dp"
android:text="Xml Data"
app:layout_constraintStart_toStartOf="@+id/textView2"
app:layout_constraintTop_toBottomOf="@+id/textView2" />

<TextView
android:id="@+id/textView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="40dp"
android:text="PARSING XML AND JSON DATA"
android:textSize="20dp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
android:id="@+id/textView2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="40dp"
android:layout_marginTop="20dp"
android:text="XML DATA"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView" />

<TextView
android:id="@+id/textView3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="20dp"
android:layout_marginEnd="40dp"
android:text="JSON DATA"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView" />

<TextView
android:id="@+id/lbl_json_data"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="30dp"
android:text="Json Data"
app:layout_constraintEnd_toEndOf="@+id/textView3"
app:layout_constraintTop_toBottomOf="@+id/textView3" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.parta_program6;

import androidx.appcompat.app.AppCompatActivity;

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

public class MainActivityextendsAppCompatActivityimplementsView.OnClickListener {

Button btnParseXml,btnParseJson;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
btnParseXml=(Button)findViewById(R.id.btn_parsexml);
btnParseJson=(Button)findViewById(R.id.btn_parsejson);
btnParseJson.setOnClickListener(this);
btnParseXml.setOnClickListener(this);
}
```

```
@Override
public void onClick(View v) {

if(v.equals(btnParseJson))
{
    Intent it=new Intent(this,ViewActivity.class);
    it.putExtra("mode",1);
    startActivity(it);

}
else if(v.equals(btnParseXml))
{
     Intent it=new Intent(this,ViewActivity.class);
     it.putExtra("mode",2);
    startActivity(it);
}
}
}
```

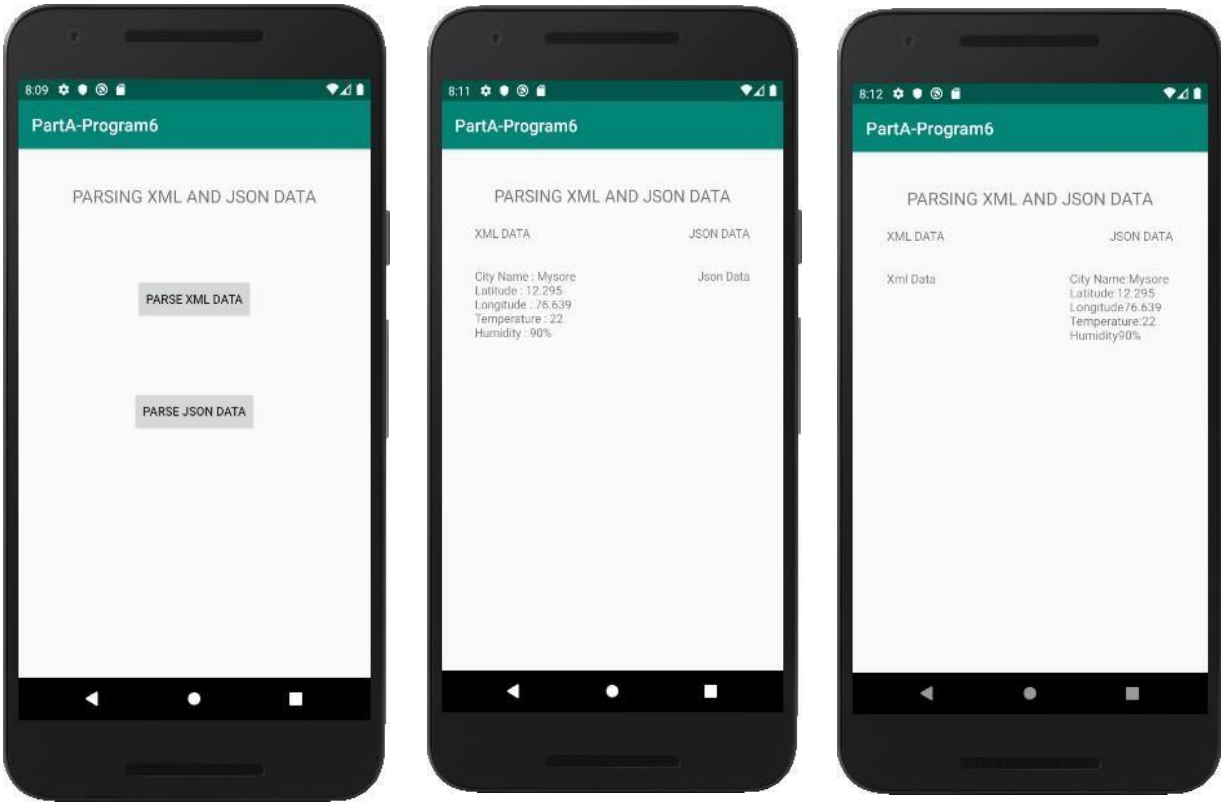
ViewActivity.java

```
package com.example.parta_program6;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.TextView;

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 java.io.InputStream;

import  javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;

public class ViewActivityextendsAppCompatActivity {

TextViewlblXmlData,lblJsonData;

int mode=0;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_view);
lblXmlData=(TextView)findViewById(R.id.lbl_xml_data);
lblJsonData=(TextView)findViewById(R.id.lbl_json_data);
mode=getIntent().getIntExtra("mode",0);
```

```
if(mode==1)
parseJson();

else
parseXmlDocument();

}

public String parseXmlDocument()
{
try {

InputStream is = getAssets().open("input.xml");

DocumentBuilderFactorydbFactory = DocumentBuilderFactory.newInstance();
DocumentBuilderdBuilder = dbFactory.newDocumentBuilder();
Document doc = dBuilder.parse(is);

Element element=doc.getDocumentElement();
element.normalize();

NodeListnList = doc.getElementsByTagName("employee");

for (int i=0; i<nList.getLength(); i++) {

Node node = nList.item(i);
if (node.getNodeType() == Node.ELEMENT_NODE) {
Element element2 = (Element) node;
lblXmlData.setText("City Name : " + getValue("city_name", element2)+"\n");
lblXmlData.append("Latitude : " + getValue("Latitude", element2)+"\n");
lblXmlData.append("Longitude : " + getValue("Longitude", element2)+"\n");
lblXmlData.append("Temperature : " + getValue("Temperature", element2)+"\n");
lblXmlData.append("Humidity : " + getValue("Humidity", element2)+"\n");

   }
 }
}
catch (Exception e) {e.printStackTrace();}
return  null;
}

private static String getValue(String tag, Element element) {
NodeListnodeList = element.getElementsByTagName(tag).item(0).getChildNodes();
Node node = nodeList.item(0);
return node.getNodeValue();
}

public void parseJson()
{
try {
InputStreaminputStream=getAssets().open("input.json");
byte[] data=new byte[inputStream.available()];
inputStream.read(data);
```

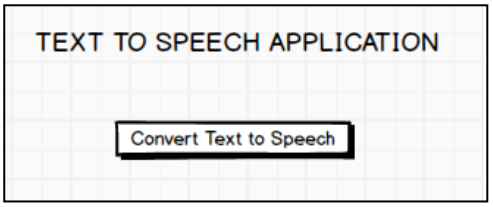
```
String readData=new String(data);
JSONObjectjsonObject=new JSONObject(readData);
JSONObject jsonObject1=jsonObject.getJSONObject("employee");
lblJsonData.setText("City Name:"+jsonObject1.getString("city_name")+"\n");
lblJsonData.append("Latitude:"+jsonObject1.getString("Latitude")+"\n");
lblJsonData.append("Longitude"+jsonObject1.getString("Longitude")+"\n");
lblJsonData.append("Temperature:"+jsonObject1.getInt("Temperature")+"\n");
lblJsonData.append("Humidity"+jsonObject1.getString("Humidity")+"\n");

}
catch (Exception e) {e.printStackTrace();}
 }
}
```

Sample Output



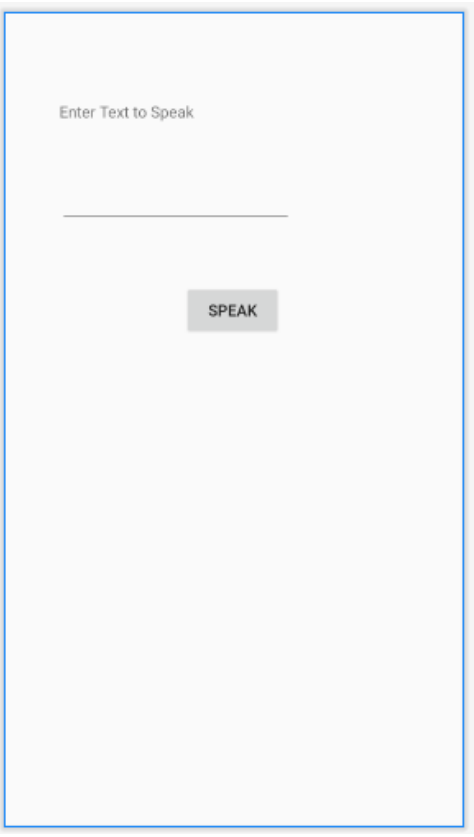
Program 7

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.



1. Create a New Android Project with Empty Activity.
2. Open activity_main.xml file from res➔ layout folder, check/add ConstraintLayout as the root view.
3. Create the layout design using Drag and Drop framework.
4. Add Listeners to Button Click Event:
   - Create a class which implments OnClickListener interface.
   - Override onClick() method of OnClickListener Interface.
   - Register the button for click event by calling setOnClickListener() method of View class and pass the object of the class that implemented OnClickListener Interface.
5. Initialize TextToSpeech Engine and the Language to Speak using setLanguage() method
6. Use Speak() method to speak the text passed to it.

Design



activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayoutxmlns:android="http://schemas.andr
oid.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/txt_texttospeak"
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_marginStart="50dp"
android:layout_marginTop="80dp"
android:text="Enter Text to Speak"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />

<EditText
android:id="@+id/editText"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="48dp"
android:ems="10"
android:inputType="textPersonName"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="@+id/textView"
app:layout_constraintTop_toBottomOf="@+id/textView" />

<Button
android:id="@+id/btn_speak"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="52dp"
android:text="Speak"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/editText" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

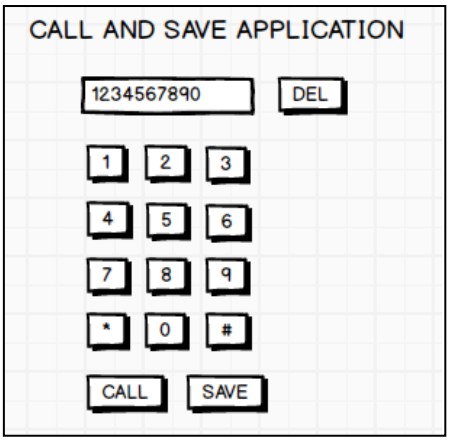
MainActivity.java

```
package  com.example.parta.parta_program7;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.speech.tts.TextToSpeech;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import java.util.Locale;

public class MainActivityextendsAppCompatActivityimplementsView.OnClickListener {

EditTexttxtSpeak;
Button btnSpeak;
TextToSpeechtextToSpeech;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
txtSpeak=(EditText)findViewById(R.id.editText);
btnSpeak=(Button)findViewById(R.id.btn_speak);
btnSpeak.setOnClickListener(this);
textToSpeech=new TextToSpeech(getBaseContext(),
new TextToSpeech.OnInitListener() {
@Override
public void onInit(int status) {
if(status!=TextToSpeech.ERROR)
{
Toast.makeText(getBaseContext(),"Success", Toast.LENGTH_LONG).show();
 }
    }
 });
textToSpeech.setLanguage(Locale.UK);
}

public void onClick(View v)
{
        String text=txtSpeak.getText().toString();
        textToSpeech.speak(text,TextToSpeech.QUEUE_FLUSH,null);

 }
}
```

Sample Output

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.



1. Create a New Android Project with Empty Activity.
2. Open activity_main.xml file from res➔ layout folder, check/add ConstraintLayout as the root view.
3. Create the layout design using Drag and Drop framework.
4. Add Listeners to Button Click Event:
   - Create a class which implments OnClickListener interface.
   - Override onClick() method of OnClickListener Interface.
   - Register the button for click event by calling setOnClickListener() method of View class and pass the object of the class that implemented OnClickListener Interface.
5. Declare uses permission android.permission.CALL_PHONE in the manifest file.
6. Use ACTION_CALL intent name and pass the "tel:<phone-number> as URI in intent data and start the call activity.
7. Use intent name and pass the "Telephone Number" and "unknown" as name as intent data call Contacts Save Activity.

Design

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayoutxmlns:android="http://schemas.andr
oid.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_marginTop="50dp"
android:text="PHONE DAILER"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />

<EditText
android:id="@+id/txt_phonenumber"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:ems="10"
android:inputType="textPersonName"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView" />

<Button
android:id="@+id/btn_delete"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="Delete"
app:layout_constraintStart_toEndOf="@+id/txt_phonenumber"
app:layout_constraintTop_toBottomOf="@+id/textView" />

<Button
android:id="@+id/btn_one"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="1"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/txt_phonenumber" />

<Button
android:id="@+id/btn_two"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="30dp"
android:text="2"
```

```
app:layout_constraintEnd_toStartOf="@+id/btn_three"
app:layout_constraintStart_toEndOf="@+id/btn_one"
app:layout_constraintTop_toBottomOf="@+id/txt_phonenumber" />

<Button
android:id="@+id/btn_three"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="30dp"
android:layout_marginEnd="20dp"
android:text="3"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toBottomOf="@+id/txt_phonenumber" />

<Button
android:id="@+id/btn_four"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="4"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/btn_one" />

<Button
android:id="@+id/btn_five"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="30dp"
android:text="5"
app:layout_constraintEnd_toStartOf="@+id/btn_six"
app:layout_constraintStart_toEndOf="@+id/btn_four"
app:layout_constraintTop_toBottomOf="@+id/btn_two" />

<Button
android:id="@+id/btn_six"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="30dp"
android:layout_marginEnd="20dp"
android:text="6"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toBottomOf="@+id/btn_three" />

<Button
android:id="@+id/btn_seven"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="7"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/btn_four" />

<Button
android:id="@+id/btn_eight"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="30dp"
```

```
android:text="8"
app:layout_constraintEnd_toStartOf="@+id/btn_nine"
app:layout_constraintStart_toEndOf="@+id/btn_seven"
app:layout_constraintTop_toBottomOf="@+id/btn_five" />

<Button
android:id="@+id/btn_nine"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="30dp"
android:layout_marginEnd="20dp"
android:text="9"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toBottomOf="@+id/btn_six" />

<Button
android:id="@+id/btn_zero"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="30dp"
android:text="0"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/btn_eight" />

<Button
android:id="@+id/btn_call"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="Call"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/btn_zero" />

<Button
android:id="@+id/btn_save"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="30dp"
android:layout_marginEnd="20dp"
android:text="Save"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toBottomOf="@+id/btn_zero" />

<Button
android:id="@+id/btn_start"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="*"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/btn_seven" />

<Button
android:id="@+id/btn_hash"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
```

```
android:layout_marginTop="30dp"
android:layout_marginEnd="20dp"
android:text="#"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toBottomOf="@+id/btn_nine" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.part_a_program_8;

import androidx.appcompat.app.AppCompatActivity;
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 MainActivityextends AppCompatActivityimplements View.OnClickListener{

Button btnOne,btnTwo,btnThree,btnFour,btnFive;
Button btnSix,btnSeven,btnEight,btnNine,btnZero;
Button btnDel,btnStar,btnHash,btnCall,btnSave;

EditTexttxtPhonenumber;

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

btnOne=(Button)findViewById(R.id.btn_one);
btnOne.setOnClickListener(this);

btnTwo=(Button)findViewById(R.id.btn_two);
btnTwo.setOnClickListener(this);

btnThree=(Button)findViewById(R.id.btn_three);
btnThree.setOnClickListener(this);


btnFour=(Button)findViewById(R.id.btn_four);
btnFour.setOnClickListener(this);

btnFive=(Button)findViewById(R.id.btn_five);
btnFive.setOnClickListener(this);

btnSix=(Button)findViewById(R.id.btn_six);
btnSix.setOnClickListener(this);

btnSeven=(Button)findViewById(R.id.btn_seven);
btnSeven.setOnClickListener(this);

btnEight=(Button)findViewById(R.id.btn_eight);
btnEight.setOnClickListener(this);
```

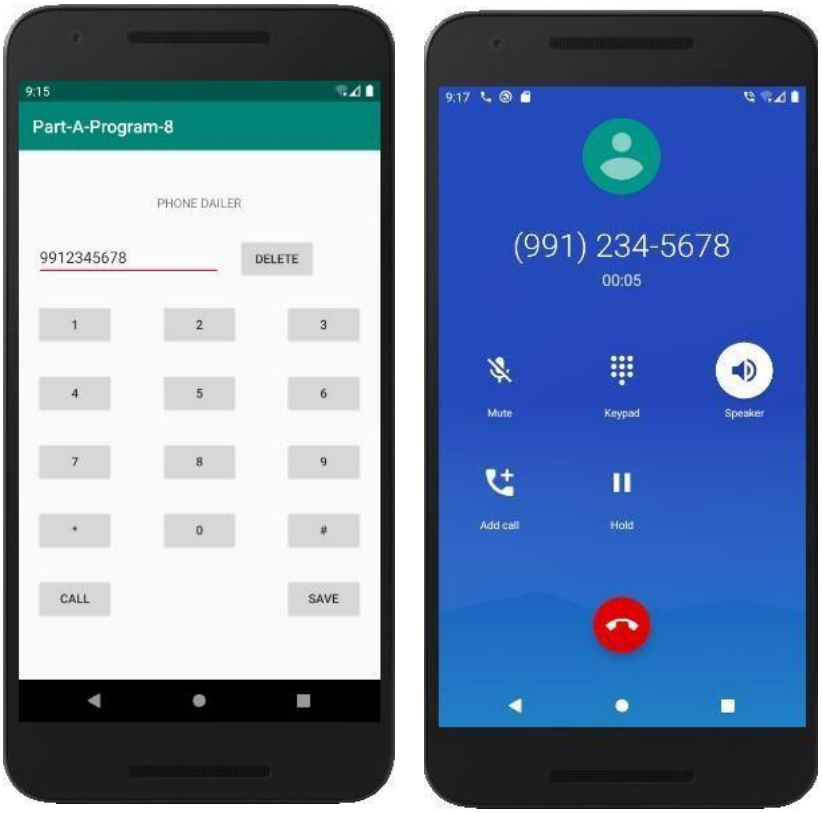
```
btnNine=(Button)findViewById(R.id.btn_nine);
btnNine.setOnClickListener(this);

btnZero=(Button)findViewById(R.id.btn_zero);
btnZero.setOnClickListener(this);

btnStar=(Button)findViewById(R.id.btn_start);
btnStar.setOnClickListener(this);

btnHash=(Button)findViewById(R.id.btn_hash);
btnHash.setOnClickListener(this);

btnCall=(Button)findViewById(R.id.btn_call);
btnCall.setOnClickListener(this);

btnSave=(Button)findViewById(R.id.btn_save);
btnSave.setOnClickListener(this);

btnDel=(Button)findViewById(R.id.btn_delete);
btnDel.setOnClickListener(this);

txtPhonenumber=(EditText)findViewById(R.id.txt_phonenumber);

txtPhonenumber.setText("");
}

public void onClick(View v)
{
if(v.equals(btnOne))
txtPhonenumber.append("1");

else if(v.equals(btnTwo))
txtPhonenumber.append("2");

else if(v.equals(btnThree))
txtPhonenumber.append("3");

else if(v.equals(btnFour))
txtPhonenumber.append("4");

else if(v.equals(btnFive))
txtPhonenumber.append("5");

else if(v.equals(btnSix))
txtPhonenumber.append("6");

else if(v.equals(btnSeven))
txtPhonenumber.append("7");

else if(v.equals(btnEight))
txtPhonenumber.append("8");

else if(v.equals(btnNine))
txtPhonenumber.append("9");


else if(v.equals(btnZero))
txtPhonenumber.append("0");
```

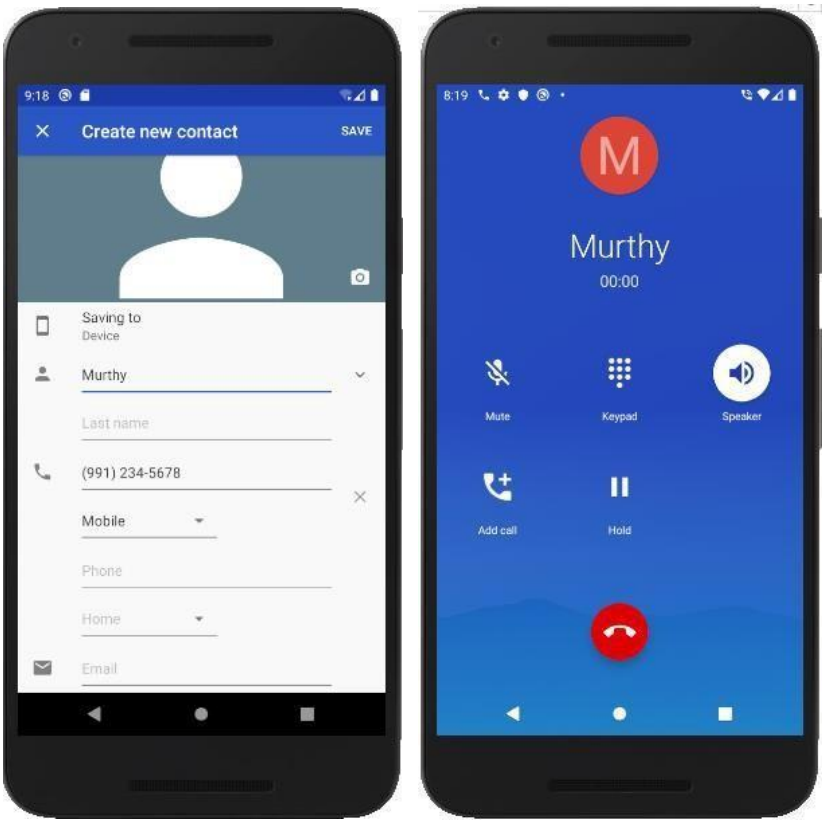
```
else if(v.equals(btnStar))
txtPhonenumber.append("*");

else if(v.equals(btnHash))
txtPhonenumber.append("#");

else  if(v.equals(btnSave))
{
Intent contactIntent= new Intent
                    (ContactsContract.Intents.Insert.ACTION);
contactIntent.setType
                    (ContactsContract.RawContacts.CONTENT_TYPE);

contactIntent
.putExtra(ContactsContract.Intents.Insert.NAME,"Unknown");
contactIntent.putExtra(ContactsContract.Intents.Insert.PHONE,
txtPhonenumber.getText().toString());

startActivity(contactIntent);

}

else  if(v.equals(btnDel))
{
String data=txtPhonenumber.getText().toString();
if(data.length()>0)
{
txtPhonenumber.setText
                        (data.substring(0,data.length()-1));

}
else
{
txtPhonenumber.setText("");
}
}

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

String data = txtPhonenumber.getText().toString();
Intent intent=new Intent(Intent.ACTION_DIAL);
intent.setData(Uri.parse("tel:"+ data));
startActivity(intent);
                        }
                }
        );

    }

}
```

AndriodManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.part_a_program_8">

<uses-permission  android:name="android.permission.CALL_PHONE"/>

<application
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:roundIcon="@mipmap/ic_launcher_round"
android:supportsRtl="true"
android:theme="@style/AppTheme">
<activity android:name=".MainActivity">
<intent-filter>
<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
</application>

</manifest>
```

Sample Output



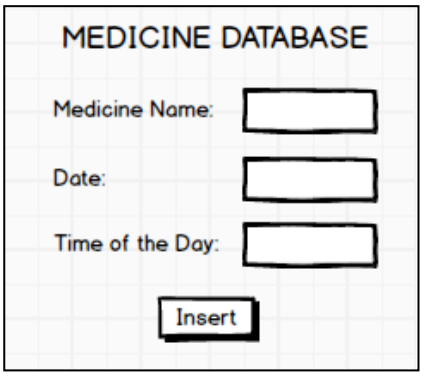


PART B

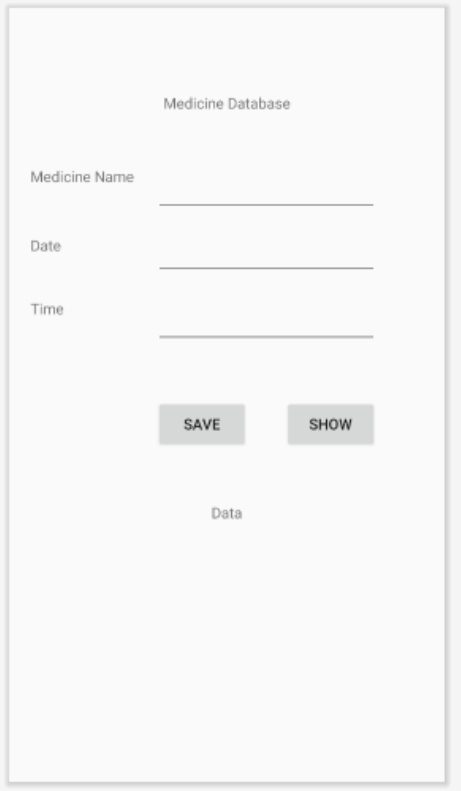
Part B programs should be developed as an application and are to be demonstrated as a mini project in a group by adding extra features or the students can also develop their application and demonstrate it as a mini-project. (Projects/programs are not limited to the list given in Part B).

Program 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 Evening or Night. Trigger an alarm based on the Date and Time of the Day and display the Medicine Name.



Design



activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayoutxmlns:android="http://schemas.andr
oid.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/textView2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="80dp"
android:text="Medicine Database"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />

<TextView
android:id="@+id/textView3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="20dp"
android:text="Medicine Name"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="@+id/txt_medicine_name" />
```

```
<TextView
android:id="@+id/textView4"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="20dp"
android:text="Date"
app:layout_constraintBottom_toBottomOf="@+id/txt_date"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/txt_medicine_name" />

<TextView
android:id="@+id/textView5"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="20dp"
android:text="Time"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="@+id/txt_time" />

<EditText
android:id="@+id/txt_medicine_name"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="20dp"
android:layout_marginTop="50dp"
android:ems="10"
android:inputType="textPersonName"
app:layout_constraintStart_toEndOf="@+id/textView3"
app:layout_constraintTop_toBottomOf="@+id/textView2" />

<EditText
android:id="@+id/txt_date"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="15dp"
android:ems="10"
android:inputType="textPersonName"
app:layout_constraintStart_toStartOf="@+id/txt_medicine_name"
app:layout_constraintTop_toBottomOf="@+id/txt_medicine_name" />

<EditText
android:id="@+id/txt_time"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="20dp"
android:ems="10"
android:inputType="textPersonName"
app:layout_constraintStart_toStartOf="@+id/txt_date"
app:layout_constraintTop_toBottomOf="@+id/txt_date" />

<Button
android:id="@+id/btn_save"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="50dp"
android:text="Save"
app:layout_constraintStart_toStartOf="@+id/txt_time"
app:layout_constraintTop_toBottomOf="@+id/txt_time" />
```

```
<Button
android:id="@+id/btn_show"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="50dp"
android:text="Show"
app:layout_constraintEnd_toEndOf="@+id/txt_time"
app:layout_constraintTop_toBottomOf="@+id/txt_time" />

<TextView
android:id="@+id/lbl_data"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="50dp"
android:text="Data"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/btn_save" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MyDatabase.java

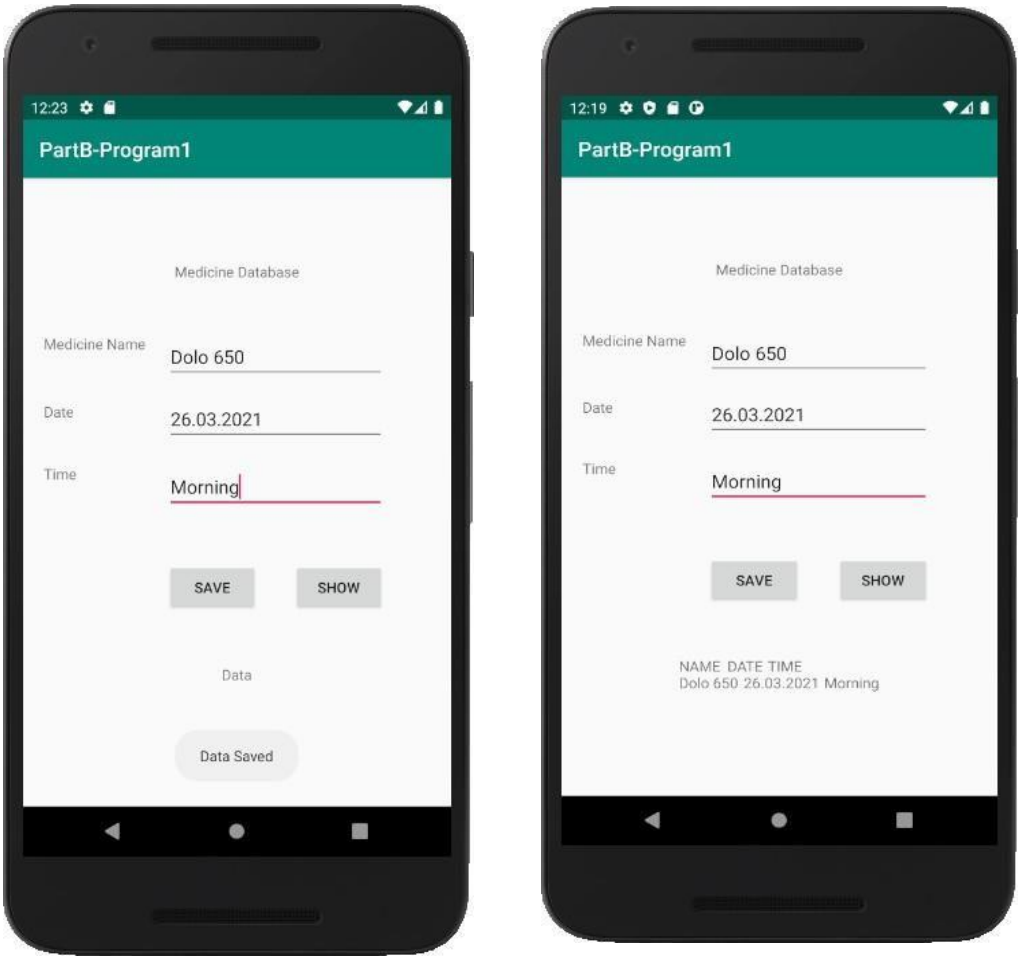
```
package com.example.partb_program1;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

import androidx.annotation.Nullable;

public class MyDatabaseextends SQLiteOpenHelper {

public static String DATABASE_NAME="medicine.db";

public MyDatabase(@Nullable Context context, @Nullable String name, @Nullable
SQLiteDatabase.CursorFactory factory, int version) {
super(context, name, factory, version);
    }

@Override
public void onCreate(SQLiteDatabasedb) {

db.execSQL("CREATE TABLE MEDICINE_NAMES (NAME TEXT,MDATE TEXT,MTIME TEXT)");

    }

@Override
public void onUpgrade(SQLiteDatabasedb, int oldVersion, int newVersion) {

    }
}
```

MainActivity.java

```
package com.example.partb_program1;

import androidx.appcompat.app.AppCompatActivity;

import android.content.ContentValues;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import org.w3c.dom.Text;

public class MainActivityextends AppCompatActivityimplements View.OnClickListener {

EditTexttxtMedicineName,txtDate,txtTime;
    Button btnSave,btnShow;
TextViewlblData;

MyDatabasemyDatabase;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
txtMedicineName=(EditText)findViewById(R.id.txt_medicine_name);
txtDate=(EditText)findViewById(R.id.txt_date);
txtTime=(EditText)findViewById(R.id.txt_time);

btnSave=(Button)findViewById(R.id.btn_save);
btnSave.setOnClickListener(this);
btnShow=(Button)findViewById(R.id.btn_show);
btnShow.setOnClickListener(this);
lblData=(TextView)findViewById(R.id.lbl_data);

myDatabase=new MyDatabase(getBaseContext(),
MyDatabase.DATABASE_NAME,null,1);

}

public void onClick(View v)
{
if(v.equals(btnSave))
{
String medicineName= txtMedicineName.getText().toString();
String date=txtDate.getText().toString();
String time=txtTime.getText().toString();

SQLiteDatabase database=myDatabase.getWritableDatabase();
ContentValues cv=new ContentValues();
cv.put("NAME",medicineName);
cv.put("MDATE",date);
cv.put("MTIME",time);
```

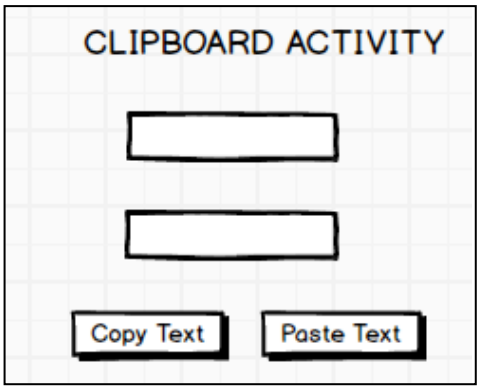
```
database.insert("MEDICINE_NAMES",null,cv);
Toast.makeText(getBaseContext(),"Data Saved",Toast.LENGTH_LONG).show();
}
else if(v.equals(btnShow))
{

SQLiteDatabase database=myDatabase.getReadableDatabase();
Cursor cursor= database.query("MEDICINE_NAMES",
new String[]{"NAME","MDATE","MTIME"},null,null,null,null,null);
lblData.setText("NAME\tDATE\tTIME\n");
while(cursor.moveToNext())
{
lblData.append(cursor.getString(0)+"\t");
lblData.append(cursor.getString(1)+"\t");
lblData.append(cursor.getString(2)+"\n");
        }
     }
  }
}
```

Sample Output



Program 7

Develop an application that makes use of the clipboard framework for copying and pasting of thetext. The activity consists of two EditText controls and two Buttons to trigger the copy and pastefunctionality.



Design

activity_main.xml

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


<Button
android:id="@+id/btn_create"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="10dp"
android:layout_marginTop="40dp"
android:text="Create"
app:layout_constraintEnd_toStartOf="@+id/textView2"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView2" />

<Button
android:id="@+id/btn_open"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="40dp"
android:layout_marginEnd="10dp"
android:text="Open"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toEndOf="@+id/textView2"
app:layout_constraintTop_toBottomOf="@+id/textView2" />

<TextView
android:id="@+id/textView2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="50dp"
android:text="File Application"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />

<EditText
android:id="@+id/txt_content"
android:layout_width="272dp"
android:layout_height="138dp"
android:layout_marginTop="50dp"
android:ems="10"
android:inputType="textPersonName"
app:layout_constraintTop_toBottomOf="@+id/btn_create"
tools:layout_editor_absoluteX="65dp" />

<Button
android:id="@+id/btn_save"
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
android:layout_marginTop="50dp"
android:text="Save"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/txt_content" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

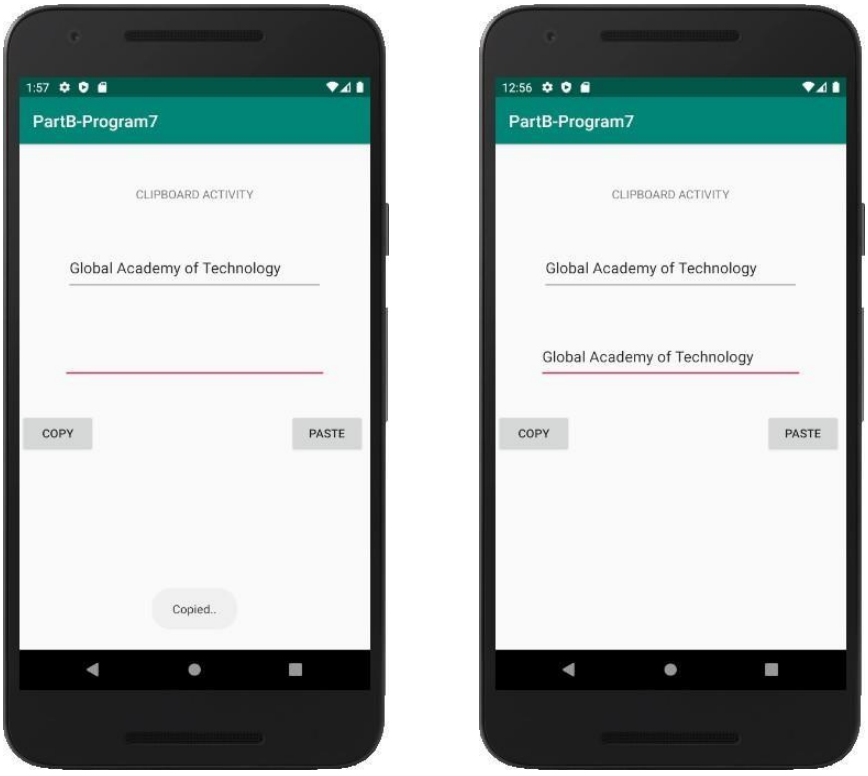
```
package com.example.partbprogram7;

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

public class MainActivityextends AppCompatActivityimplements View.OnClickListener {
EditTexttxtCopy,txtPaste;
Button btnCopy,btnPaste;

ClipboardManagermyClipboard;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
txtCopy=(EditText)findViewById(R.id.txt_copy);
txtPaste=(EditText)findViewById(R.id.txt_paste);

btnCopy=(Button)findViewById(R.id.btn_copy);
btnCopy.setOnClickListener(this);

btnPaste=(Button)findViewById(R.id.btn_paste);
btnPaste.setOnClickListener(this);

myClipboard= (ClipboardManager)getSystemService(CLIPBOARD_SERVICE);
}

@Override
public void onClick(View v) {
if(v.equals(btnCopy))
{
ClipDatamyClip;
String data = txtCopy.getText().toString();
myClip = ClipData.newPlainText("text", data);
myClipboard.setPrimaryClip(myClip);
Toast.makeText(getBaseContext(),"Copied..",Toast.LENGTH_LONG).show();
}
```

```
else if(v.equals(btnPaste))
{
ClipDataabc = myClipboard.getPrimaryClip();
ClipData.Item item = abc.getItemAt(0);
txtPaste.setText(item.getText().toString());
     }
  }
}
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.partbprogram7">

<application
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:roundIcon="@mipmap/ic_launcher_round"
android:supportsRtl="true"
android:theme="@style/AppTheme">
<activity android:name=".MainActivity">
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
</application>
</manifest>
```

Sample Output



Program 8

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

$E = P * (r(1+r)^n)/((1+r)^n-1)$

where

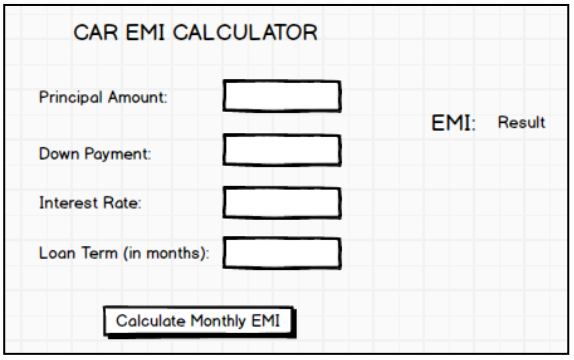
E = The EMI payable on the car loan amount

P = The Car loan Principal Amount

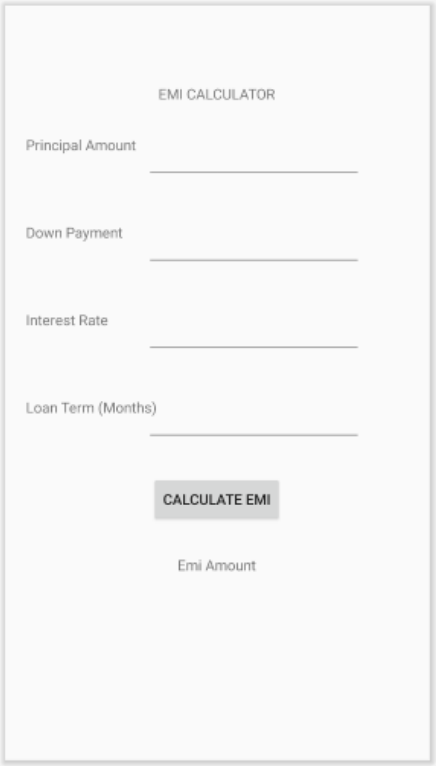
r = The interest rate value computed on a monthly basis

n = The loan tenure in the form of months

The down payment amount has to be deducted from the principal amount paid towards buying theCar. Develop an application that makes use of this AIDL service to calculate the EMI. Thisapplication 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 andInterest Rate values.



Design



activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayoutxmlns:android="http://schemas.andr
oid.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/lblpayment"
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:text="EMI CALCULATOR"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
tools:layout_editor_absoluteY="76dp" />

<TextView
android:id="@+id/textView2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="Principal Amount"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView" />

<EditText
android:id="@+id/txt_principal"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="10dp"
android:layout_marginTop="30dp"
android:ems="10"
android:inputType="textPersonName"
app:layout_constraintStart_toEndOf="@+id/textView2"
app:layout_constraintTop_toBottomOf="@+id/textView" />

<TextView
android:id="@+id/downpayment"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Down Payment"
app:layout_constraintStart_toStartOf="@+id/textView2"
app:layout_constraintTop_toTopOf="@+id/txt_downnpayment" />

<EditText
android:id="@+id/txt_downnpayment"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="40dp"
android:ems="10"
android:inputType="textPersonName"
app:layout_constraintStart_toStartOf="@+id/txt_principal"
```

```
app:layout_constraintTop_toBottomOf="@+id/txt_principal" />

<TextView
android:id="@+id/textView4"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Interest Rate"
app:layout_constraintStart_toStartOf="@+id/downpayment"
app:layout_constraintTop_toTopOf="@+id/txt_interestrate" />


<EditText
android:id="@+id/txt_interestrate"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="40dp"
android:ems="10"
android:inputType="textPersonName"
app:layout_constraintStart_toStartOf="@+id/txt_downnpayment"
app:layout_constraintTop_toBottomOf="@+id/txt_downnpayment" />

<TextView
android:id="@+id/textView5"
android:layout_width="130dp"
android:layout_height="33dp"
android:layout_marginTop="8dp"
android:text="Loan Term (Months)"
app:layout_constraintStart_toStartOf="@+id/textView4"
app:layout_constraintTop_toTopOf="@+id/txt_termmonths" />

<EditText
android:id="@+id/txt_termmonths"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="20dp"
android:layout_marginTop="32dp"
android:ems="10"
android:inputType="textPersonName"
app:layout_constraintStart_toStartOf="@+id/txt_interestrate"
app:layout_constraintTop_toBottomOf="@+id/txt_interestrate" />

<Button
android:id="@+id/btn_calculate"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="30dp"
android:text="Calculate EMI"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/txt_termmonths" />

<TextView
android:id="@+id/lbl_emiamount"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="30dp"
android:text="Emi Amount"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
```

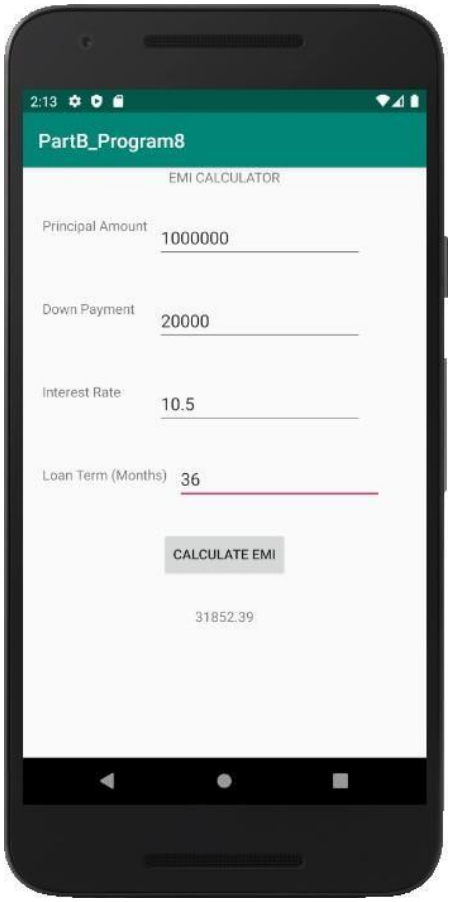
```
app:layout_constraintTop_toBottomOf="@+id/btn_calculate" />

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

MainActivity.java

```
package com.example.partb_program8;

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

import java.text.DecimalFormat;
import java.util.logging.SimpleFormatter;

public class MainActivityextends AppCompatActivityimplements View.OnClickListener {

EditTexttxtPrinicple,txtDownPayment,txtInterestRate,txtLoanTerm;

Button btnCalculate;

TextViewlblResult;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
txtPrinicple=(EditText)findViewById(R.id.txt_principal);
txtDownPayment=(EditText)findViewById(R.id.txt_downnpayment);
txtInterestRate=(EditText)findViewById(R.id.txt_interestrate);
txtLoanTerm=(EditText)findViewById(R.id.txt_termmonths);

btnCalculate=(Button)findViewById(R.id.btn_calculate);
btnCalculate.setOnClickListener(this);

lblResult=(TextView)findViewById(R.id.lbl_emiamount);

}

public void onClick(View v)
{
try
{

DecimalFormat formatter = new
DecimalFormat("#0.00");

double prinicipleAmount=
Double.parseDouble(txtPrinicple.
getText().toString());
double downPayment=Double.parseDouble(txtDownPayment.getText().toString());
```

```
prinicipleAmount=prinicipleAmount-downPayment;
double interestRate=Double.parseDouble(txtInterestRate.getText().toString());
interestRate=interestRate/(12*100);
double loanTerm=Double.parseDouble(txtLoanTerm.getText().toString());

double emi=prinicipleAmount*
                    (interestRate*Math.pow((1+interestRate),loanTerm))
                    /(Math.pow((1+interestRate),loanTerm)-1);
lblResult.setText(String.valueOf(formatter.format(emi)));
}

catch(Exception e)
{
Toast.makeText(getBaseContext(),"Invalid Input",Toast.LENGTH_LONG).show();
}
}
}
```

Sample Output



Reference 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)
2. Erik Hellman, "Android Programming – Pushing the Limits", 1st Edition, Wiley India Pvt Ltd, 2014. ISBN-13: 978-8126547197
3. Dawn Griffiths and David Griffiths, "Head First Android Development", 1st Edition, O"Reilly SPD Publishers, 2015. ISBN-13: 978-9352131341
4. 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

1. **What is JAVA?**

   Java is a high-level programming language and is platform-independent.

2. **What do you mean by Platform Independence?**

   Platform Independence you can run and compile program in one platform and can execute in any other platform.

3. **What are the features of JAVA?**

   Object-oriented
   Inheritance
   Encapsulation
   Polymorphism
   Abstraction

4. **Name the Java IDE's?**

   Eclipse and NetBeans are the IDE's of JAVA.

5. **What is a Class?**

   All Java codes are defined in a Class. It has variables and methods.

   - Variables are attributes which define the state of a class.
   - Methods are the place where the exact business logic has to be done. It contains a set of statements (or) instructions to satisfy the particular requirement.

6. **What is an Object?**

   An instance of a class is called an object. The object has state and behavior.Whenever the JVM reads the "new()" keyword then it will create an instance of that class.

7. **What is meant by the Local variable and the Instance variable?**

   - Local variables are defined in the method and scope of the variables that exist inside the method itself.
   - Instance variable is defined inside the class and outside the method and the scope of the variables exists throughout the class.

8. **What do you mean by Constructor?**

   When a new object is created in a program a constructor gets invoked corresponding to the class.The constructor is a method which has the same name as the class name.If a user doesn't create a constructor implicitly a default constructor will be created.The constructor can be overloaded.

9. What is meant by Overloading?

a. Method overloading happens for different classes or within the same class.
b. For method overloading, sub-class method should satisfy the below conditions with the Super-class method (or) methods in the same class itself:

Same method name
Different argument types
There may be different return types

10. **What is Inheritance?**

a. Inheritance means one class can extend to another class. So that the codes can be reused from one class to another class. The existing class is known as the Super class whereas the derived class is known as a sub class.

11. **What is meant by Method Overriding?**

Method overriding happens if the sub-class method satisfies the below conditions with the Super-class method:

- Method name should be the same
- The argument should be the same
- Return type should also be the same

The key benefit of overriding is that the Sub-class can provide some specific information about that sub-class type than the super-class.

12. What is abstraction?

Abstraction is a process of hiding the implementation details and showing only functionality to the user.

13. What is abstract class?

A class that is declared as abstract is known as abstract class. It needs to be extended and its method implemented. It cannot be instantiated.

14. What is interface?

Interface is a blueprint of a class that have static constants and abstract methods.It can be used to achieve fully abstraction and multiple inheritance.

15. What is Thread in Java?

In Java, the flow of execution is called Thread. Every java program has at least one thread called the main thread, the main thread is created by JVM. The user can define their own threads by extending the Thread class (or) by implementing the Runnable interface. Threads are executed concurrently.

16. What is the difference between Thread and Process in Java?

The thread is a subset of Process, in other words, one process can contain multiple threads. Two process runs on different memory space, but all threads share same memory space.

17. **How do you make a thread in Java?**

There are two ways available to make a thread.

- Extend Thread class: Extending a Thread class and override the run method. The thread is available in java.lang.thread.
- Implement Runnable interface: Another way is by implementing the runnable interface. For that, we should provide the implementation for the run () method which is defined in the interface.

18. **Explain about wait () method.**

Wait () method is used to make the thread to wait in the waiting pool. When the wait () method is executed during a thread execution then immediately the thread gives up the lock on the object and goes to the waiting pool.

Wait () method tells the thread to wait for a given amount of time.Then the thread will wake up after notify () (or) notify all () method is called.

19. **How to stop a thread in java? Explain about sleep () method in a thread?**

We can stop a thread by using the following thread methods:

- Sleeping
- Waiting
- Blocked

Sleep: Sleep () method is used to sleep the currently executing thread for the given amount of time. Once the thread is wake up it can move to the runnable state. So sleep () method is used to delay the execution for some period.

20. **What is Multi-threading?**

Multiple threads are executed simultaneously. Each thread starts its own stack based on the flow (or) priority of the threads.

21. Explain the thread life cycle in Java.

Thread has the following states:

- New
- Runnable
- Running
- Non-runnable (Blocked)
- Terminated

22. What is meant by Exception?

An Exception is a problem that can occur during the normal flow of execution. A method can throw an exception when something wails at runtime. If that exception couldn't be handled, then the execution gets terminated before it completes the task.

23. What are the different ways to handle exceptions?

Using try/catch

By declaring throws keyword

24. What are the Exception handling keywords in Java?

- **try**
- **catch**
- **finally**
- **throw**
- **throws**

25. What is the final keyword in Java?

Final variable: Once a variable is declared as final, then the value of the variable could not be changed. It is like a constant.

26. What is a IO stream?

It is a stream of data that flows from source to destination. Good example is file copying. Two streams are involved – input stream and output stream. An input stream reads from the file and stores the data in the process (generally in a temporary variable). The output stream reads from the process and writes to the destination file

27. What are FileInputStream and FileOutputStream?

These two are general purpose classes used by the programmer very often to copy file to file. These classes work well with files containing less data of a few thousand bytes as by performance these are very poor. For larger data, it is preferred to use BufferedInputStream (or BufferedReader) and BufferedOutputStream (or BufferedWriter).

28. What is an applet?

A program that a Java enabled browser can download and run is an Applet.

29. What are the attributes of Applet tags? Explain the purposes?

height: Defines height of applet, width: Defines width of applet

30. How will you initialize an Applet?

Write my initialization code in the applets init method or applet constructor.

31. What is the sequence for calling the methods by AWT for applets?

When an applet begins, the AWT calls the following methods, in this sequence:

- init()
- start()
- paint()

When an applet is terminated, the following sequence of method calls takes place

- stop()
- destroy()

32. What is nested class?

A class which is declared inside another class is known as nested class. There are 4 types of nested class member inner class, local inner class, annonymous inner class and static nested class.

33. What is servlet?

A servlet is an extension to a server. It is a Java class that is loaded to expand the functionality of the server. It helps extend the capability of web servers by providing support for dynamic response and data persistence.

A servlet runs inside a Java Virtual Machine (JVM) on the server, and hence it is safe and portable. Servlets operate only within the domain of the server. These do not require support for Java in the web browser