AMC Engineering College

NAAC Accredited with ‘B+’ Grade, Kalkere , Bengaluru-560083
(Affiliated to Visvesvaraya Technological University, Belagavi)

MOBILE APPLICATION DEVELOPMENTLABORATORY MANUAL MASTER COPY

VI Semester
CourseCode:18CSMP6 8

[As per the Choice Based Credit System Scheme]
Scheme:2018

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	Prof. Deivapraba T
Approved by	HOD, Dept. of CSE

Table of Content

Vision of the Institute

Become a premier institution imparting quality education in engineering and managementto meet the changing needs of society

Mission of the Institute

- Create environment conducive for continuous learning through quality teaching andlearning processes supported by modern infrastructure
- Promote Research and Innovation through collaboration with industries
- Inculcate ethical values and environmental consciousness through holistic educationprograms

Vision of the Department

“Be a premier department in the field of Computer Science & Engineering tomeet the technological challenges of the society”

Mission of the Department

MD 1 To provide state of the art infrastructure facilities

MD 2 To provide exposure to the latest tools in the area of computer hardware and software

MD 3 To strive for academic excellence through research in Computer Science and Engineering with creative teaching-learning pedagogy

MD 4 To establish Industry Institute Interaction and make students ready for the Industrial environment

MD 5 To transform students into entrepreneurial, technically competent, socially responsible and ethical computer science professional

Program Educational Objectives (PEOs)

After the course completion, CSE graduates will be able to:

PEO1: Succeed in engineering/management positions with professional ethics.

PEO2: Engage in improving professional knowledge through certificate/post-graduate programsin engineering or management.

PEO3: Establish themselves as entrepreneurs and contribute to the Society.

Program Specific Outcomes (PSOs)

PSO1: Design, implement and test System Software and Application Software to meet thedesired needs.

PSO2: Develop solutions in the area of Communication Networks, Database Systems andComputing Systems.

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 anunderstanding 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 professionalengineeringpractice.

7. **Environment and sustainability**: Understand the impact of the professional engineering solutions insocietal 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 report sand designdocumentation, 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-longlearning**: 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

CourseCode:18CSMP68

Course Prerequisite: Core Java

Course Objectives

Upon completion of this course, students are expected to:

1. Learn and acquire the art of AndroidProgramming.
2. Configure Android studio to run theapplications.
3. Understand and implement Android's User interfacefunctions.
4. Create, modify and query on SQLitedatabase.
5. Inspect different methods of sharing data usingservices.

Course Outcomes

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

18CSMP68.1	Create, test and debug Android application by setting up Android development environment.
18CSMP68.2	Implement adaptive, responsive user interfaces that work across a wide range of devices.
18CSMP68.3	Demonstrate methods in storing, sharing and retrieving data in Android applications.
18CSMP68.4	Infer the role of permissions and security for Android applications.

SYLLABUS

MOBILE APPLICATION DEVELOPMENT

SubjectCode:18CSMP68 **IA Marks:40**

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

Practical Hrs:3Hours/Week **Exam Hours: 03No. of Credits:02**

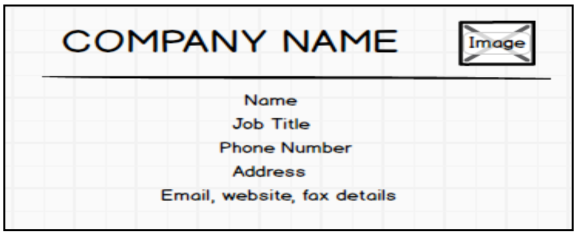
Descriptions (if any):

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 theseprograms. 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 thelist given in PartB).

PART A

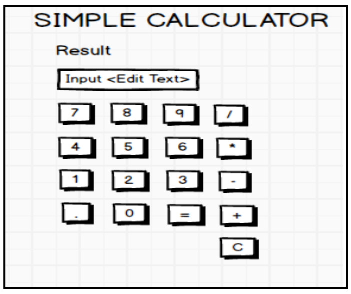
Program 1

Create an application to design a Visiting Card. The Visiting card should have a company logo atthe 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.



Program 2

Develop an Android application using controls like Button, TextView, EditText for designing aCalculator having basic functionality

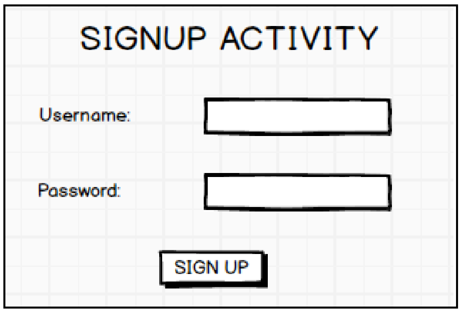


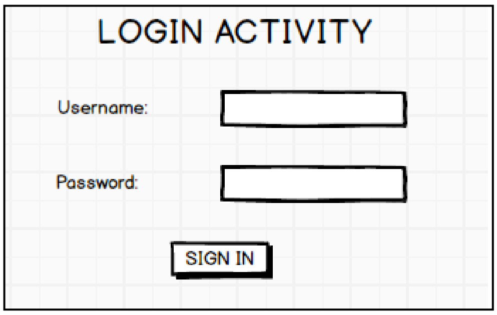
Program 3

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

- Password should contain uppercase and lower case letters.
- Password should contain letters and numbers.
- Password should contain special characters.
- Minimum length of the password (the default value is8).

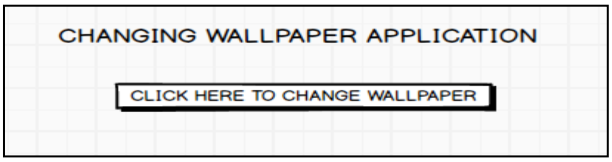
On successful **SIGN UP** proceed to the next Login activity. Here the user should **SIGN IN** usingthe Username and Password created during signup activity. If the Username and Password are matched then navigate to the next activity which displays a message saying “Successful Login” or else display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





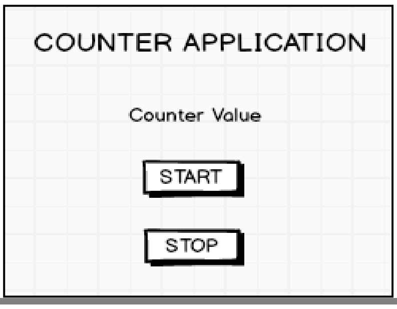
Program 4

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


CHANGING WALLPAPER APPLICATION
CLICK HERE TO CHANGE WALLPAPER

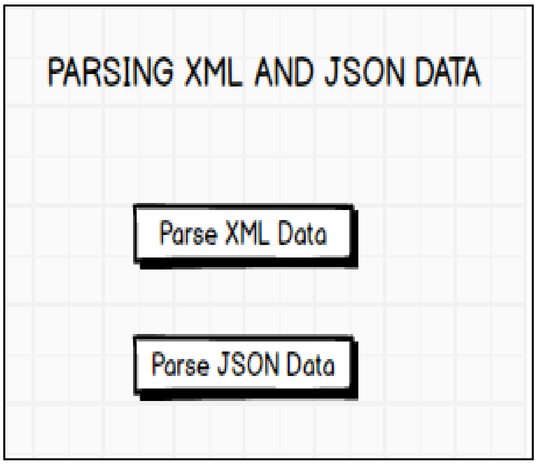
Program 5

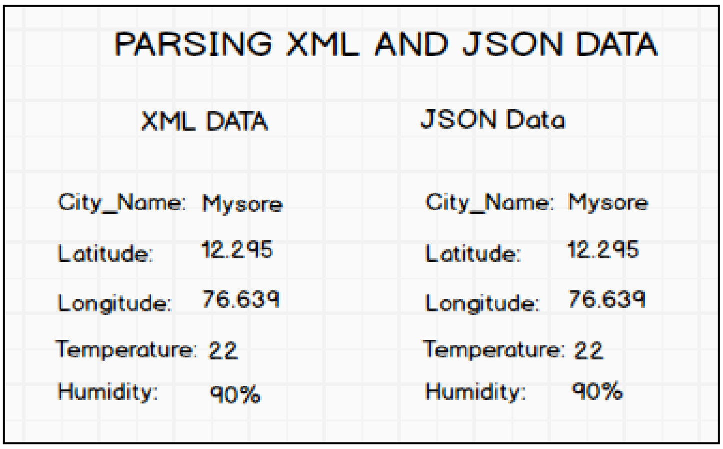
Write a program to create an activity with two buttons START and STOP. On Pressing of the START button, the activity must start the counter by displaying the numbers from One and the counter must keep on counting until the STOP button is pressed. Display the counter value in a TextView control.



Program 6

Create two files of XML and JSON type with values for City_Name, Latitude, Longitude, Temperature,andHumidity.Developanapplicationtocreateanactivitywithtwobuttonstop arse theXML and JSON files which when clicked should display the data in their respective layouts side by side.



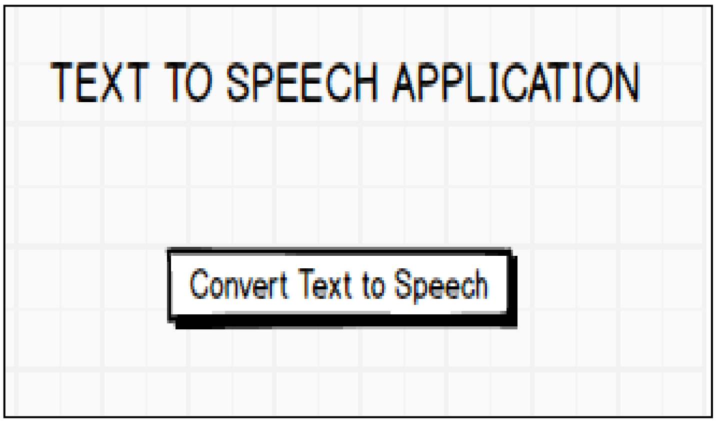


City_Name: Mysore
Latitude: 12.295
Longitude: 76.639
Temperature: 22
Humidity: 90%

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

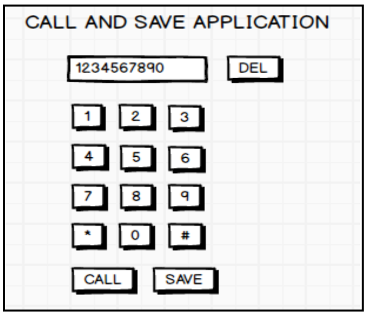
Program 7

Develop a simple application with one Edit Text so that the user can write some text in it. Createa button called "Convert Text to Speech" that converts the user input text into voice.



Program 8

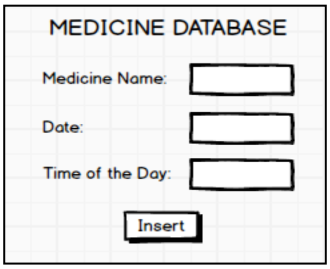
Create an activity like a phone dialer with CALL and SAVE buttons. On pressing the CALL button, it must call the phone number and on pressing the SAVE button it must save the number to the phone contacts.



PART B

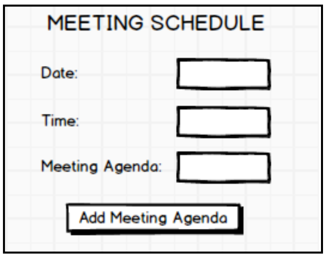
Program 1

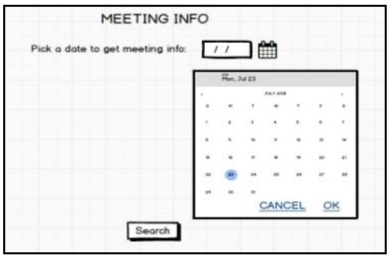
Write a program to enter Medicine Name, Date and Time of the Day as input from the user and store it in the SQLite database. Input for Time of the Day should be either Morning or Afternoonor Evening or Night. Trigger an alarm based on the Date and Time of the Day and display the Medicine Name.



Program 2

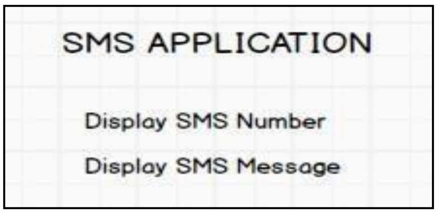
Develop a content provider application with an activity called “Meeting Schedule” which takesDate ,Time and Meeting Agenda as input from the user and store this information into the SQLite database. Create another application with an activity called “Meeting Info” having Date Pickercontrol, which on the selection of a date should display the Meeting Agenda information for thatparticular date, else it should display a toast message saying “No Meeting on this Date”.





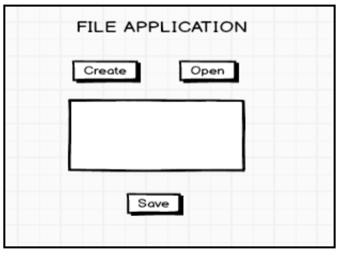
Program 3

Create an application to receive an incoming SMS which is notified to the user. On clicking thisSMS 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.



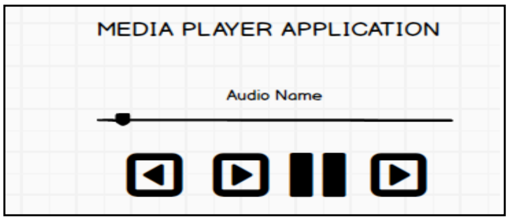
Program 4

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 displaythe 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”.



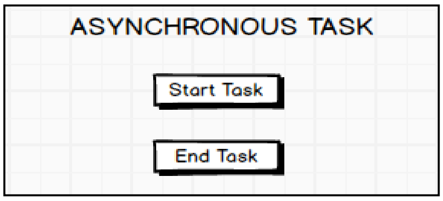
Program 5

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.



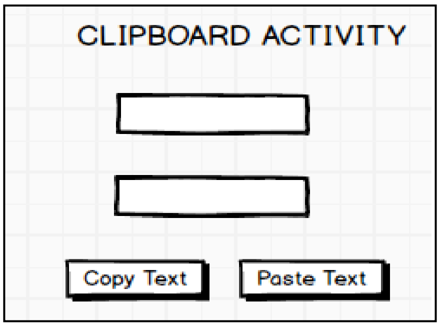
Program 6

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 **StartTask** 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 ofthetext. The activity consists of two EditText controls and two Buttons to trigger the copy and pastefunctionality.



Program 8

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

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

Where

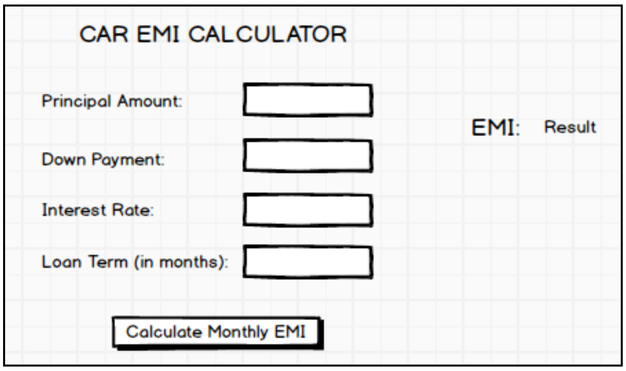
E = The EMI payable on the car loan amount

P = The Car loan Principal Amount

r = The interest rate value computed on a monthly basis

n = The loan tenure in the form of months

The down payment amount has to be deducted from the principal amount paid towards buying the Car. Develop an application that makes use of this AIDL service to calculate the EMI. This application should have four EditText to read the Principal Amount, DownPayment, InterestRate, LoanTerm (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 InterestRate values.



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	3	2	2	-	2	1	-	2	1	1	-	1

LAB EVALUATION PROCESS

WEEK WISE EVALUATION OF EACH PROGRAM

ACTIVITY	MARKS
Manual	19
Viva	05
TOTAL	**24**

INTERNAL ASSESSMENT EVALUATION (End of Semester)

SI. No	ACTIVITY	MARKS
01	Write-up	15
02	Conduction	70
03	Viva Voce	15
	Total	**100**

FINAL INTERNAL ASSESSMENT CALCULATION

Sl.No	ACTIVITY	MARKS
01	Average of weekly Entries	24
02	Internal Assessment	16
	TOTAL	**40**

Lab Rubrics

Rubrics for Evaluation of Observation Book

Attribute		EXCELLENT	GOOD	SATISFACTORY	NOT SATISFACTORY
	Max Marks	3	2	1	0
Write-Up	**3**	1. Written complete program without any errors 2. Written input and expected output for all the test cases. 3. Written program with proper indentation	1. Written Program with few logical errors. 2. Written input and expected output for few test cases. 3. Written program with moderate indentation	1. Written Program with logical and syntax errors. 2. Written input and expected output for one test case. 3. Written program with moderate indentation	1. Written Incomplete Program 2. Input and expected output is not written
	Max Marks	9-10	5-8	3-4	0-2
Execution	**10**	1. Execution of code with excellent debugging skill. 2. Got desired output for all required test case	1. Execution of code with good debugging skill. 2. Got desired output for few test cases	1. Execution of code with average debugging skill. 2. Got desired output for only one test case	No Execution
	Max Marks	3	2	1	0
Viva Voce	**3**	1. Exhibits strong hold on basic of Data Structure concepts 2. Able to demonstrate given program 3. Presents all concepts clearly	1. Exhibits good hold on basic of Data Structure concepts 2. Able to demonstrate given program 3. Presents only a few concepts clearly	1. Able to exhibits average knowledge on basic of Data Structure concepts 2. Able to demonstrate given program partially 3. Lack of confidence to answer the questions	1. Unable to explain concepts. 2. Unable to explain the program. 3. Unable to communicate ideas.

Rubrics for Evaluation of Record Book

Attribute		EXCELLENT	GOOD	NOT SATISFACTORY
	Max Marks	4	2-3	1
Write-Up	**4**	1. Written complete program without any errors 2. Written input and expected output for all the test cases. 3.Written program with proper indentation	1. Written Program with few logical errors. 2. Written input and expected output for few the test cases. 3.Written program with moderate indentation	1. Written Incomplete Program 2. Input and expected output is not written
	Max Marks	2	1	0
Results	**2**	Written desired output for all required test cases	Written desired output for few test cases	Not able to write output
	Max Marks	2	1	0
Time Management	**2**	Written Complete program with desired output and submitted on time	Written Complete program with desired output and submitted by taking extra time	Not submitted

Rubrics for Evaluation of Internal Test

Attribute		EXCELLENT	GOOD	SATISFACTORY	NOT SATISFACTORY
	Max Marks	13-15	8-12	5-7	0-4
Write-Up	15	1. Written complete program without any errors 2.Written input and expected output for all the test cases. 3.Written program with proper indentation	1.Written Program with few logical errors. 2.Written input and expected output for few test cases. 3.Written program with moderate indentation	1.Written Program with logical and syntax errors. 2.Written input and expected output for one test case 3.Written program with reasonable indentation	1.Written Incomplete Program 2.Input and expected output is not written
	Max Marks	61-70	36-60	21-35	0-20
Execution	70	1.Execution of code with excellent debugging skill. 2.Got desired output for all required test cases	1. Execution of code with good debugging skill. 2. Got desired output for few test cases	1.Execution of code with average debugging skill. 2.Got desired output for only one test case	No Execution
	Max Marks	13-15	8-12	5-7	0-4
Viva Voce	15	1. Exhibits strong hold on basic of Data Structure concepts 2. Able to demonstrate given program 3. Presents all concepts clearly	1. Exhibits good hold on basic of Data Structure concepts 2. Able to demonstrate given program 3. Presents all concepts clearly	1.Able to exhibits average knowledge on basic of Data Structure concepts 2. Able to demonstrate given program partially 3. Lack of confidence to answer the questions	1.Unable to explain concepts. 2.Unable to explain the program. 3.Unable to communicate ideas.

Evaluation Sheet

S. No	Date	Particulars	Page No.	Marks		Total (Max 24M)	Signature
				O	R		
				W+C+V (Max. 16)	W+R+T (Max. 8M)		
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
Average (Max. 24) **(A)**							
Internal Assessment Marks (Max. 16) **(B)** ***Scheme: As per VTU guidelines***							
Total Marks (A + B)							
Note:- **W:** Write-up **C :**Conduction **V :**Viva Voce **O:**Observation **R :**Record							

Procedure to Conduct Practical Examination

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

- **Change of experiment is allowed only once and marks allotted for procedure to bemade 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 PARTB
    i) Part A – Procedure + Execution + Viva = 6 + 28 + 6 = 40Marks
    ii) Part B – Procedure + Execution + Viva = 9 + 42 + 9 = 60Marks

CHAPTER 1

INTRODUCTION TO MOBILE APPLICATION DEVELOPMENT

Every day the new devices are incoming to the market with innovative options thanks to growing technology. The evolution of Mobile Application Development technology with new devices made our lives much easier.

In the smartphone world, simply having a running web site is not enough. Regarding arecent study, it has shown that about 45% and more of Google search happens using smartphones.The number is spectacular and there is a growth within the mobile business. Being obtainable on an internet-enabled device is needed for every and each business which has given the kicking start to mobile application development.

What is Mobile Development?

Mobile development, which is not about building phone apps, though it is a huge part of it. Actually, it’s doing any reasonably development for any kind of mobile devices such as developing apps for phones, tablets, smartwatches, and every form of wearable devices that run anykind of mobile operating system.

Mobile development presents a reasonably distinctive chance for a one-person development team to build an actual, usable, significant app end-to-end during a comparatively short period. However, Mobile Apps Development represents more than just a chance for the solo-developer to create their own project as it is arguably the longer term of development, as mobile devices are getting larger and bigger parts of our lives.

Android is the dominant player in mobile development platforms space, it was a bit later participant to the game, first being released in Sept 2008, virtually a year later than iOS but it has managed to achieve a reasonably massive share of the mobile market.

Technically, Android is the mobile OS with the largest most dominant share of the marketwith around 80% share compared to iOS’s 18 % share. Those numbers are a bit deceivingsince android may be a fragmented market consisting of the many different devices created by different manufacturers, running completely different versions of the Android OS.

Types of Applications

1. Native Applications:

These are applications developed to be used on a particular platform or operating system such as Android, iOS etc. Native apps are usually written in languages that the platform accepts. They arealso built using the specific Integrated Development Environment (IDE) for the given operating systems, such as Android Studio for Android Apps and XCode for iOS Apps.

The principal advantage of native apps is that they optimize the user experience. By being designedand developed specifically for that platform, they look and perform better.

Examples of some popular Native Applications are Instagram for Android, VLC media player for Android, WordPress for iOS, and 2048 game for iOS,

2. Hybrid Applications:

These are applications developed to be used across multiple platforms i.e can be deployed on bothiOS and Android platforms. Hybrid mobile applications are built in a similar manner as websites. Both use a combination of technologies like HTML, CSS, and JavaScript. However, instead of targeting a mobile browser, hybrid applications target a WebView hosted inside a native container.This enables them to do things like access hardware capabilities of the mobile device.

Today, most hybrid mobile applications leverage Apache Cordova, a platform that provides a consistent set of JavaScript APIs to access device capabilities through plug-ins, which are built withnative code.

Examples of some popular Hybrid Applications are MarketWatch, Untappd, FanReact, and TripCase.

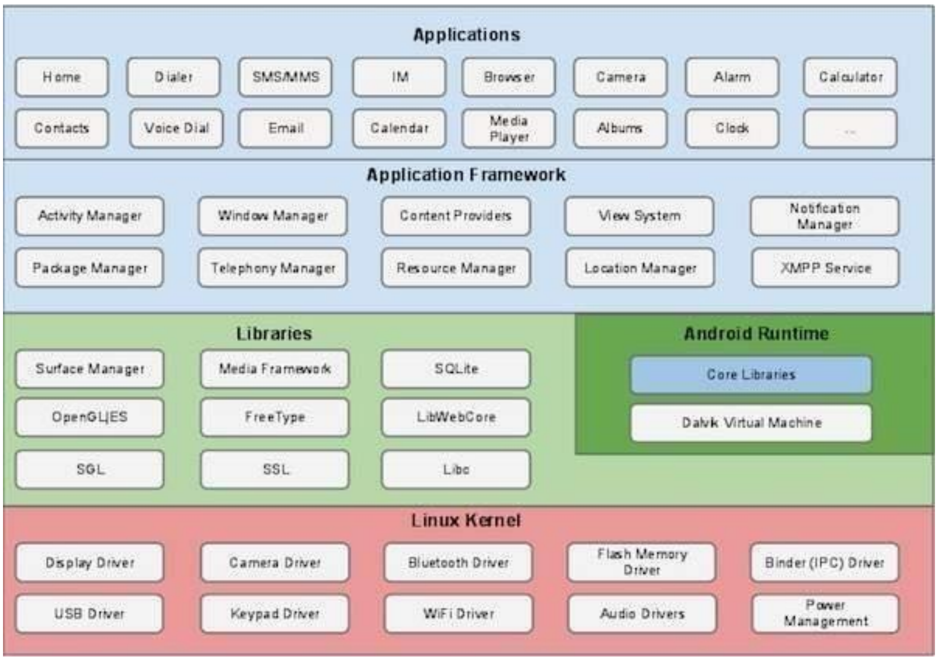
3. Progressive Web Applications (PWAs):

A Progressive Web App (PWA) is a web app that uses modern web capabilities to deliver an app- like experience to users without requiring them to install an app from the AppStore/PlayStore. Theyare usually accessible by a web URL which can always be pinned or saved on your phone's home screen. PWAs are usually built using HTML, CSS, JavaScript also.

Examples of some popular Progressive Web Applications are AliExpress's PWA, Financial TimesPWA, NASA's PWA, and the recently just launched PayStack's PWA.

Architecture

Android operating system is a stack of software components which is roughly divided into fivesections and four main layers as shown below in the architecture diagram.



Linux kernel

At the bottom of the layers is Linux - Linux 3.6 with approximately 115 patches. This provides a level of abstraction between the device hardware and it contains all the essential hardware driverslike camera, keypad, display etc. Also, the kernel handles all the things that Linux is really good atsuch as networking and a vast array of device drivers, which take the pain out of interfacing to peripheral hardware.

Libraries

On top of Linux kernel there is a set of libraries including open-source Web browser engine WebKit, well known library libc, SQLite database which is a useful repository for storage and sharing of application data, libraries to play and record audio and video, SSL libraries responsiblefor Internet security etc.

Android Libraries

This category encompasses those Java-based libraries that are specific to Android development. Examples of libraries in this category include the application framework libraries in addition to those that facilitate user interface building, graphics drawing and database access. A summary of some key core Android libraries available to the Android developer is as follows −

- **android.app** − Provides access to the application model and is the cornerstone of all Android applications.
- **android.content** − Facilitates content access, publishing and messaging between applicationsand application components.

- **android.database** – Used to access data published by content providers and includes SQLitedatabase management classes.
- **android.opengl** – A Java interface to the OpenGL ES 3D graphics rendering API.
- **android.os** – Provides applications with access to standard operating system services includingmessages, system services and inter-process communication.
- **android.text** – Used to render and manipulate text on a device display.
- **android.view** – The fundamental building blocks of application user interfaces.
- **android.widget** – A rich collection of pre-built user interface components such as buttons,labels,list views, layout managers, radio buttons etc.
- **android.webkit** – A set of classes intended to allow web-browsing capabilities to be built into applications.

Having covered the Java-based core libraries in the Android runtime, it is now time to turn our attention to the C/C++ based libraries contained in this layer of the Android software stack.

Android Runtime

This is the third section of the architecture and available on the second layer from the bottom. Thissection provides a key component called **Dalvik Virtual Machine** which is a kind of Java VirtualMachine specially designed and optimized for Android.

The Dalvik VM makes use of Linux core features like memory management and multi-threading, which is intrinsic in the Java language. The Dalvik VM enables every Android application to run in its own process, with its own instance of the Dalvik virtual machine.

The Android runtime also provides a set of core libraries which enable Android application developers to write Android applications using standard Java programming language.

Application Framework

The Application Framework layer provides many higher-level services to applications in the form of Java classes. Application developers are allowed to make use of these services in their applications.

The Android framework includes the following key services –

- **Activity Manager** – Controls all aspects of the application lifecycle and activity stack.
- **Content Providers** – Allows applications to publish and share data with other applications.

- **Resource Manager** − Provides access to non-code embedded resources such as strings, colorsettings and user interface layouts.
- **Notifications Manager** − Allows applications to display alerts and notifications to the user.
- **View System** − An extensible set of views used to create application user interfaces.
- **Applications**

  You will find all the Android application at the top layer. You will write your application to be installed on this layer only. Examples of such applications are Contacts Books, Browser, Games etc.

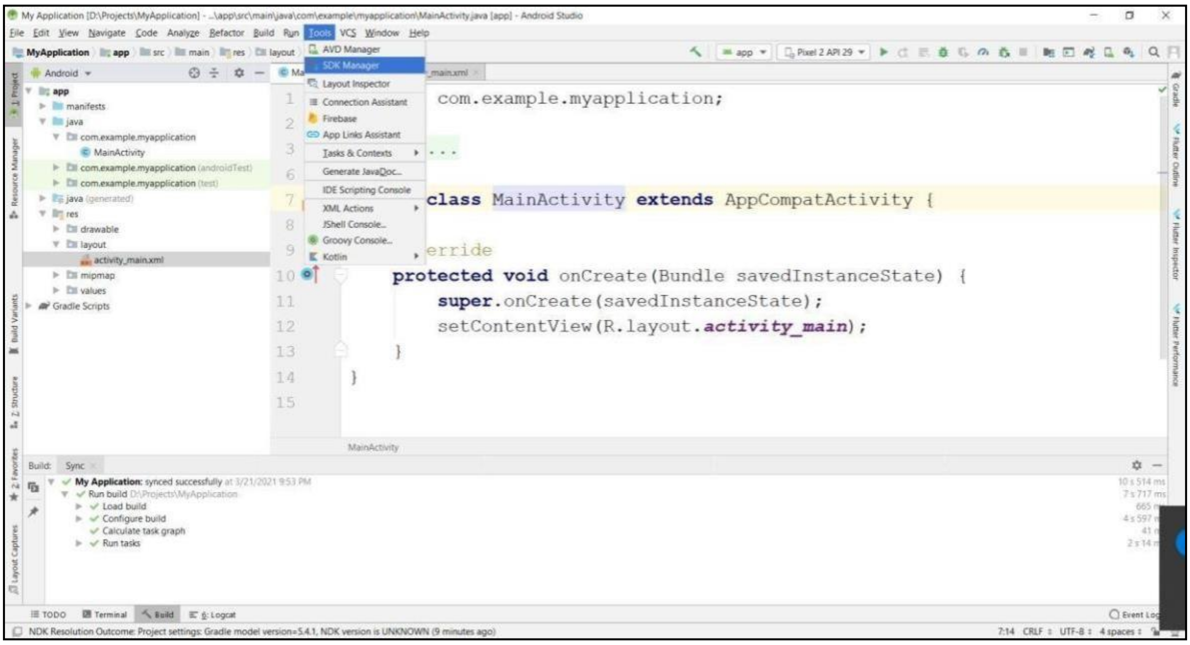
1. Android StudioTutorials

Install Android Studio andPackages:

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 SDKpackages:

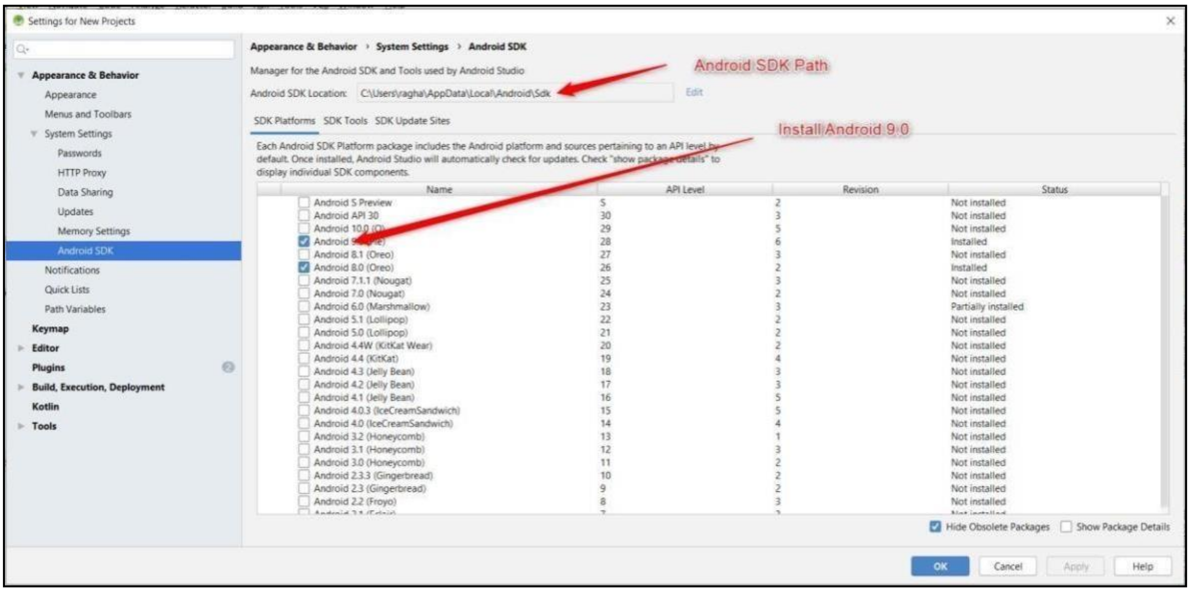
Go to Tools ➐SDK Manager



```
com.example.myapplication;

class MainActivity extends AppCompatActivity {

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

```
My Application: synced successfully at 3/21/2021 9:53 PM
  Run build D:\Projects\MyApplication
    Load build
    Configure build
    Calculate task graph
    Run tasks
```

NDK Resolution Outcome: Project settings: Gradle model version=5.4.1, NDK version is UNKNOWN (9 minutes ago)



Appearance & Behavior › System Settings › Android SDK

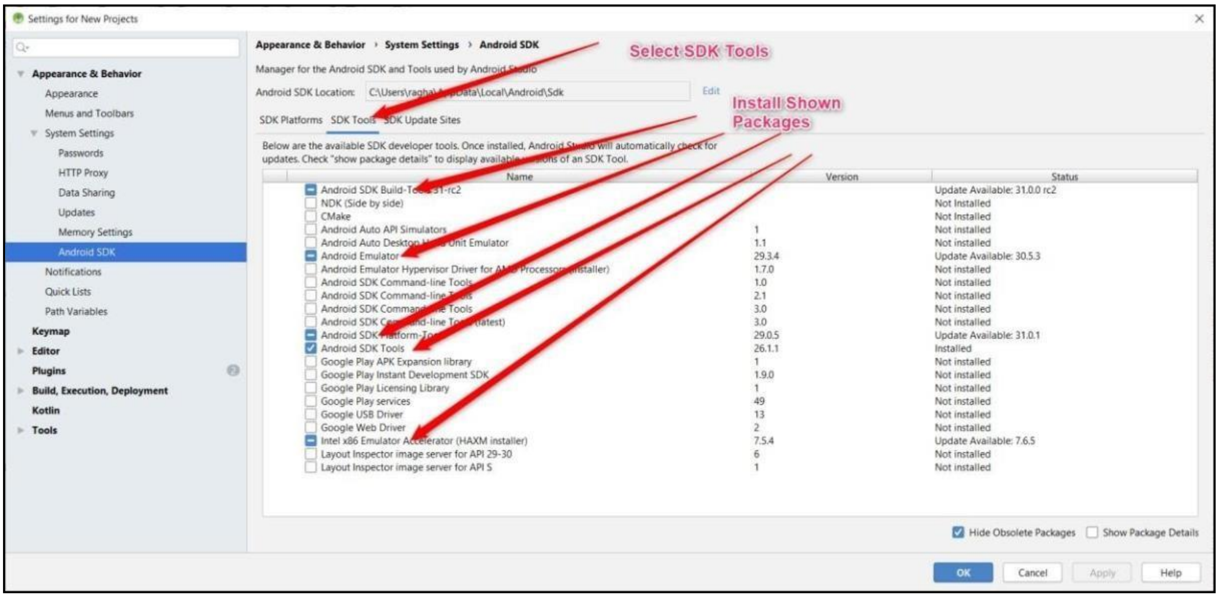
Manager for the Android SDK and Tools used by Android Studio

Android SDK Location: C:\Users\ragha\AppData\Local\Android\Sdk Edit

Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components.

Name	API Level	Revision	Status
□ Android S Preview	S	2	Not installed
□ Android API 30	30	3	Not installed
□ Android 10.0 (Q)	29	5	Not installed
☑ Android 9.0 (Pie)	28	6	Installed
□ Android 8.1 (Oreo)	27	3	Not installed
☑ Android 8.0 (Oreo)	26	2	Installed
□ Android 7.1.1 (Nougat)	25	3	Not installed
□ Android 7.0 (Nougat)	24	2	Not installed
□ Android 6.0 (Marshmallow)	23	3	Partially installed
□ Android 5.1 (Lollipop)	22	2	Not installed
□ Android 5.0 (Lollipop)	21	2	Not installed
□ Android 4.4W (KitKat Wear)	20	2	Not installed
□ Android 4.4 (KitKat)	19	4	Not installed
□ Android 4.3 (Jelly Bean)	18	3	Not installed
□ Android 4.2 (Jelly Bean)	17	3	Not installed
□ Android 4.1 (Jelly Bean)	16	5	Not installed
□ Android 4.0.3 (IceCreamSandwich)	15	5	Not installed
□ Android 4.0 (IceCreamSandwich)	14	4	Not installed
□ Android 3.2 (Honeycomb)	13	1	Not installed
□ Android 3.1 (Honeycomb)	12	3	Not installed
□ Android 3.0 (Honeycomb)	11	2	Not installed
□ Android 2.3.3 (Gingerbread)	10	2	Not installed
□ Android 2.3 (Gingerbread)	9	2	Not installed
□ Android 2.2 (Froyo)	8	3	Not installed

☑ Hide Obsolete Packages □ Show Package Details



Appearance & Behavior › System Settings › Android SDK

Manager for the Android SDK and Tools used by Android Studio

Android SDK Location: C:\Users\ragha\AppData\Local\Android\Sdk Edit

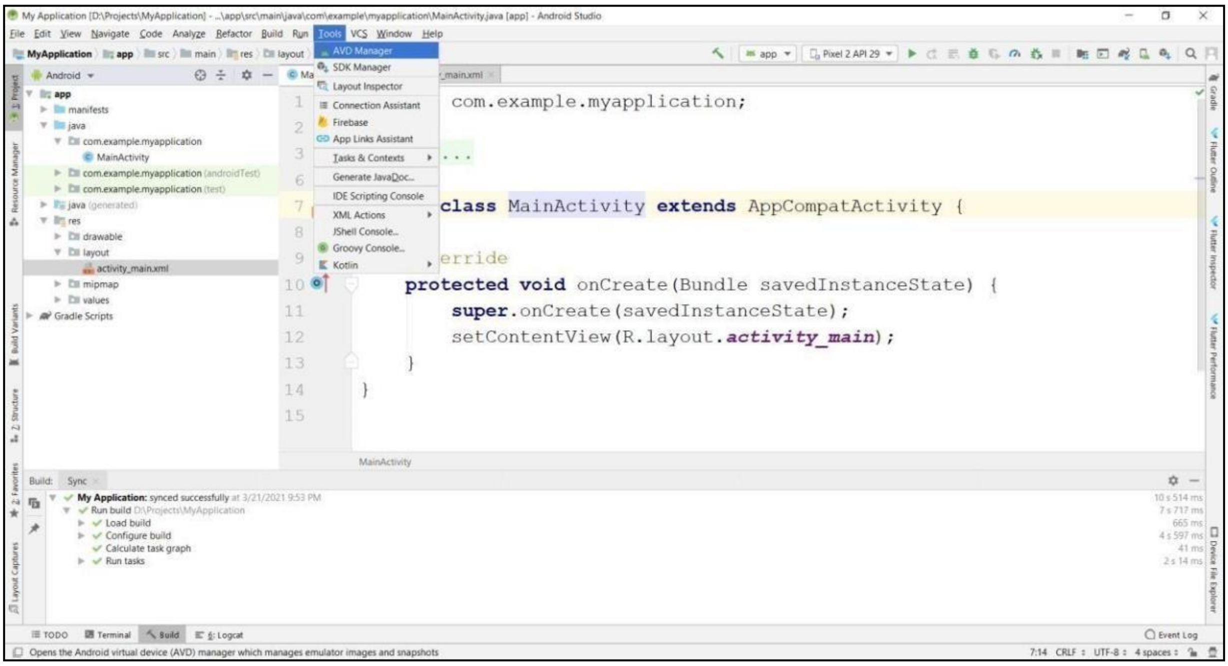
Below are the available SDK developer tools. Once installed, Android Studio will automatically check for updates. Check "show package details" to display available versions of an SDK Tool.

Name	Version	Status
▣ Android SDK Build-Tools 31-rc2		Update Available: 31.0.0 rc2
□ NDK (Side by side)		Not Installed
□ CMake		Not Installed
□ Android Auto API Simulators	1	Not installed
□ Android Auto Desktop Head Unit Emulator	1.1	Not installed
▣ Android Emulator	29.3.4	Update Available: 30.5.3
□ Android Emulator Hypervisor Driver for AMD Processors (installer)	1.7.0	Not installed
□ Android SDK Command-line Tools	1.0	Not installed
□ Android SDK Command-line Tools	2.1	Not installed
□ Android SDK Command-line Tools	3.0	Not installed
□ Android SDK Command-line Tools (latest)	3.0	Not installed
▣ Android SDK Platform-Tools	29.0.5	Update Available: 31.0.1
☑ Android SDK Tools	26.1.1	Installed
□ Google Play APK Expansion library	1	Not installed
□ Google Play Instant Development SDK	1.9.0	Not installed
□ Google Play Licensing Library	1	Not installed
□ Google Play services	49	Not installed
□ Google USB Driver	13	Not installed
□ Google Web Driver	2	Not installed
▣ Intel x86 Emulator Accelerator (HAXM installer)	7.5.4	Update Available: 7.6.5
□ Layout Inspector image server for API 29-30	6	Not installed
□ Layout Inspector image server for API S	1	Not installed

☑ Hide Obsolete Packages □ Show Package Details

CreatingEmulator

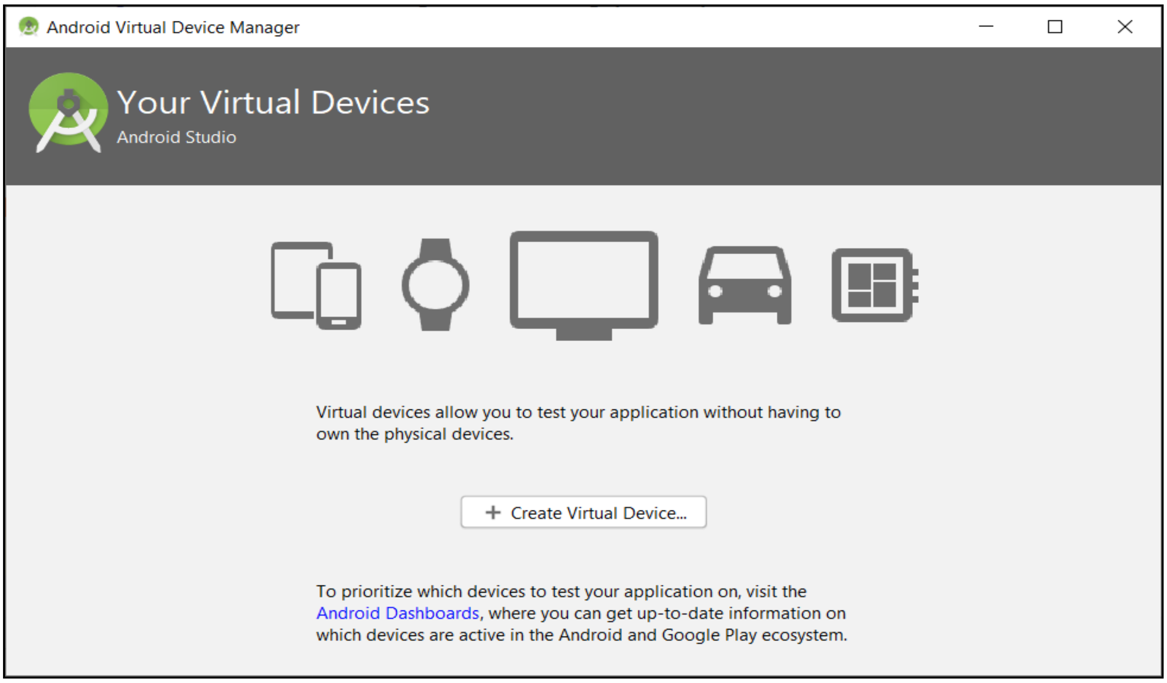
GotoTools ➐ SelectAVDManager



```
com.example.myapplication;

...

class MainActivity extends AppCompatActivity {

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

```
My Application: synced successfully at 3/21/2021 9:53 PM    10 s 514 ms
  Run build D:\Projects\MyApplication                         7 s 717 ms
    Load build                                                  665 ms
    Configure build                                           4 s 597 ms
    Calculate task graph                                         41 ms
    Run tasks                                                  2 s 14 ms
```

Opens the Android virtual device (AVD) manager which manages emulator images and snapshots



Android Virtual Device Manager

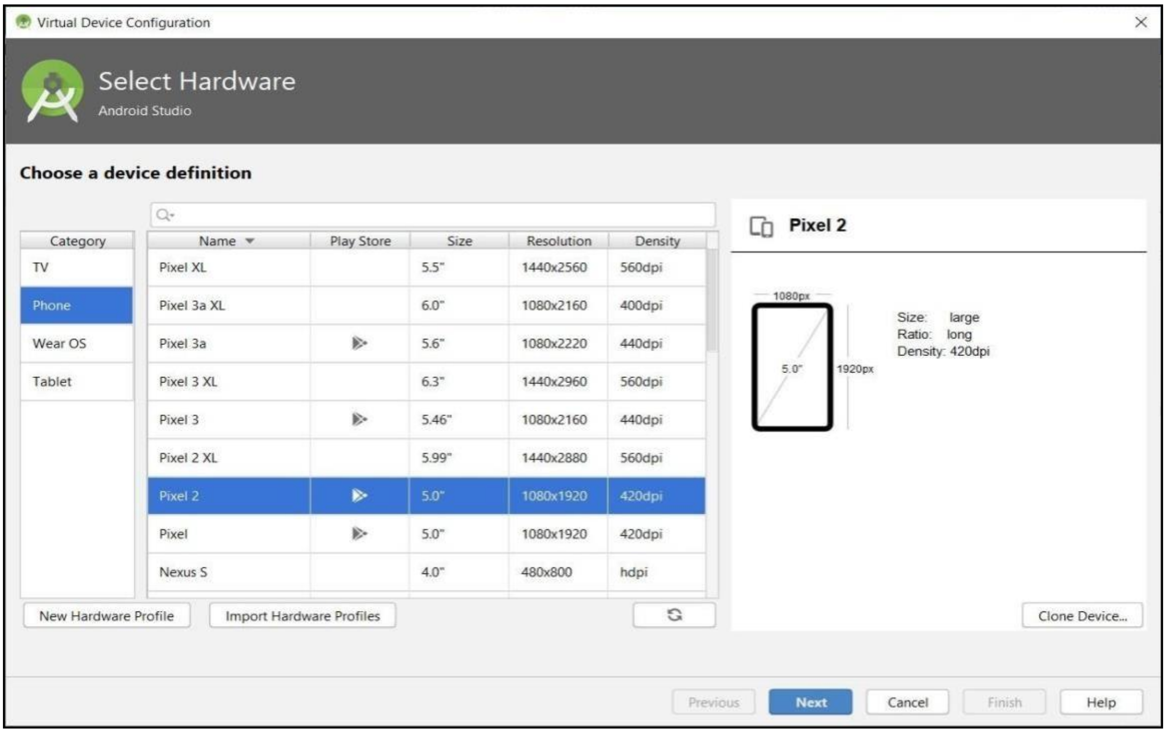
Your Virtual Devices

Android Studio

Virtual devices allow you to test your application without having to own the physical devices.

To prioritize which devices to test your application on, visit the Android Dashboards, where you can get up-to-date information on which devices are active in the Android and Google Play ecosystem.

Select CreateVirtual Device ➔ Select Phone ➔ Pixel 2 ➔ PressNext



Virtual Device Configuration

Select Hardware

Android Studio

Choose a device definition

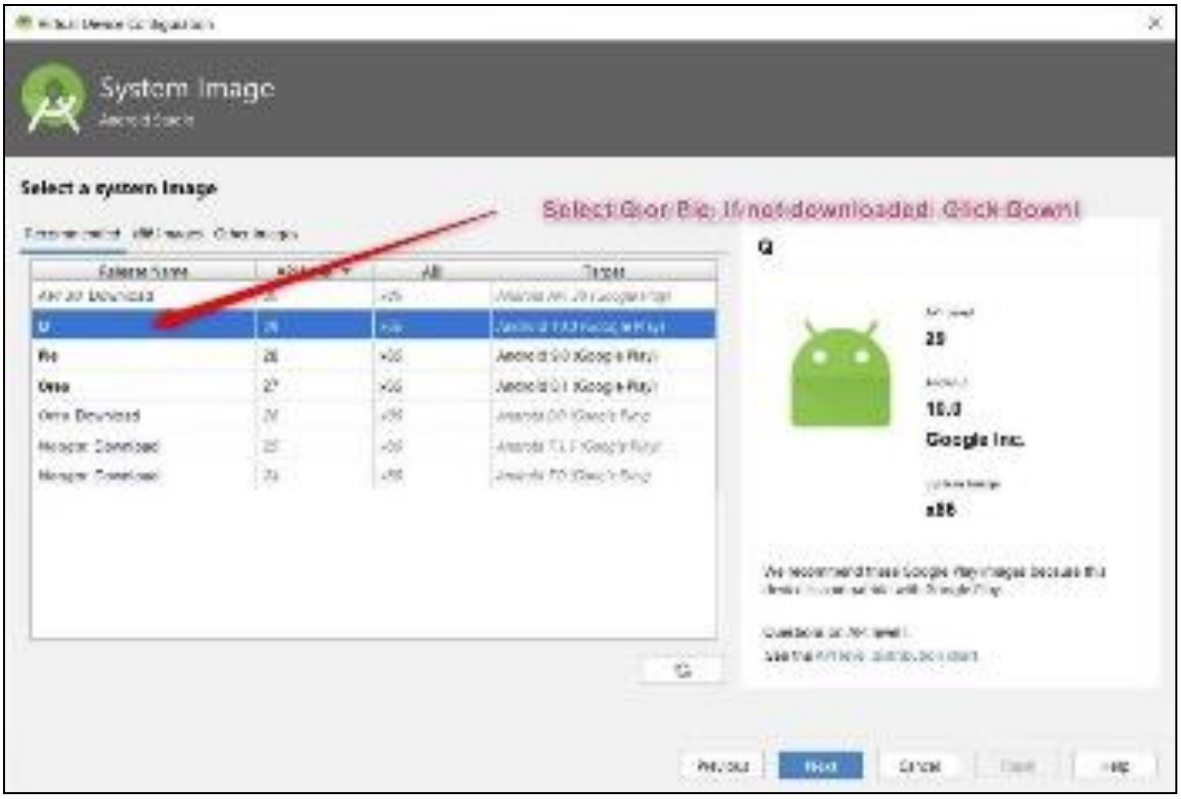
Category
TV
Phone
Wear OS
Tablet

Name	Play Store	Size	Resolution	Density
Pixel XL		5.5"	1440x2560	560dpi
Pixel 3a XL		6.0"	1080x2160	400dpi
Pixel 3a	▶	5.6"	1080x2220	440dpi
Pixel 3 XL		6.3"	1440x2960	560dpi
Pixel 3	▶	5.46"	1080x2160	440dpi
Pixel 2 XL		5.99"	1440x2880	560dpi
Pixel 2	▶	5.0"	1080x1920	420dpi
Pixel	▶	5.0"	1080x1920	420dpi
Nexus S		4.0"	480x800	hdpi

Pixel 2

Size: large
Ratio: long
Density: 420dpi

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



System Image

Select a system image

Select Q or Pie: If not downloaded: Click Down!

Release Name	API Level	ABI	Target
[illegible] Download	[illegible]	x86	[illegible]
Q	29	x86	Android 10.0 (Google Play)
Pie	28	x86	Android 9.0 (Google Play)
Oreo	27	x86	Android 8.1 (Google Play)
Oreo Download	26	x86	Android 8.0 (Google Play)
Nougat Download	25	x86	Android 7.1.1 (Google Play)
Nougat Download	24	x86	Android 7.0 (Google Play)

Q

API Level: 29

Android: 10.0

Google Inc.

System Image: x86

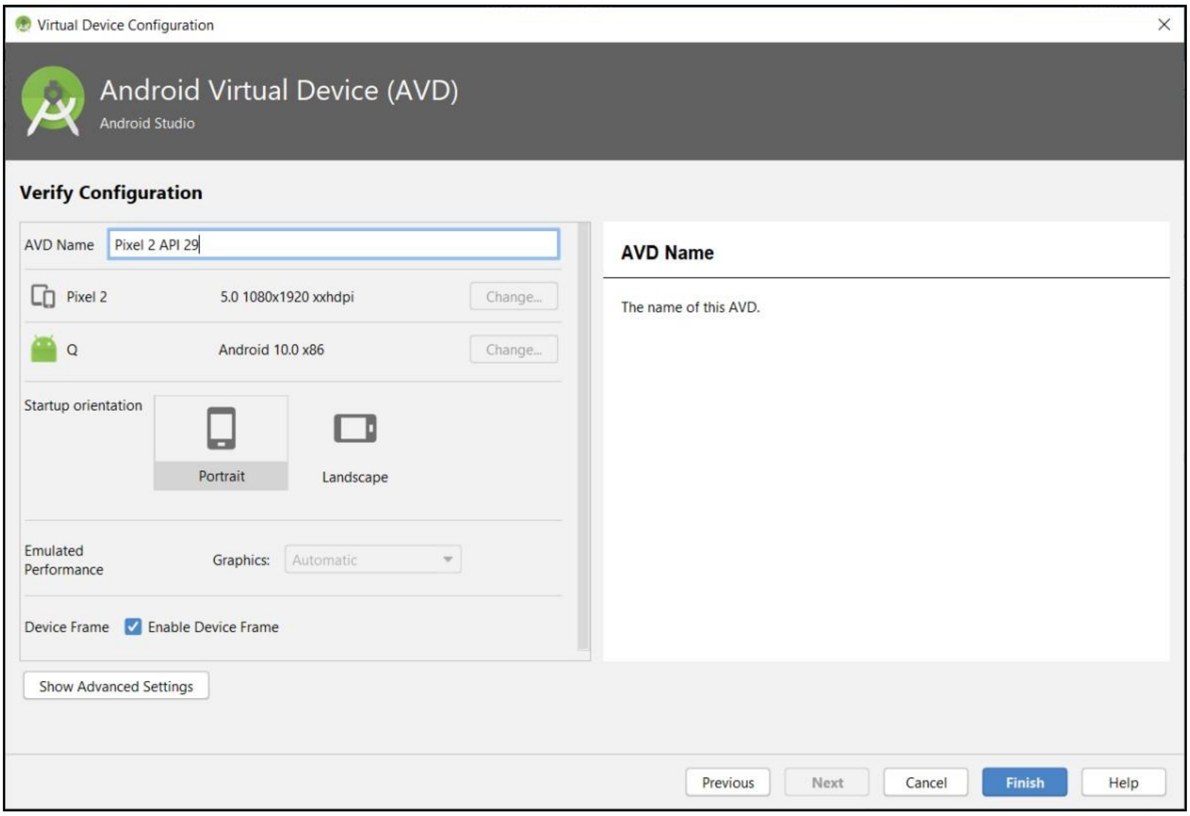
We recommend these Google Play images because this device is compatible with Google Play.

Questions on API level?

See the API level distribution chart

Enter AVD Name and PressFinish.

Press Play Button to StartEmulator



Virtual Device Configuration

Android Virtual Device (AVD)

Android Studio

Verify Configuration

AVD Name: Pixel 2 API 29

Pixel 2 — 5.0 1080x1920 xxhdpi

Q — Android 10.0 x86

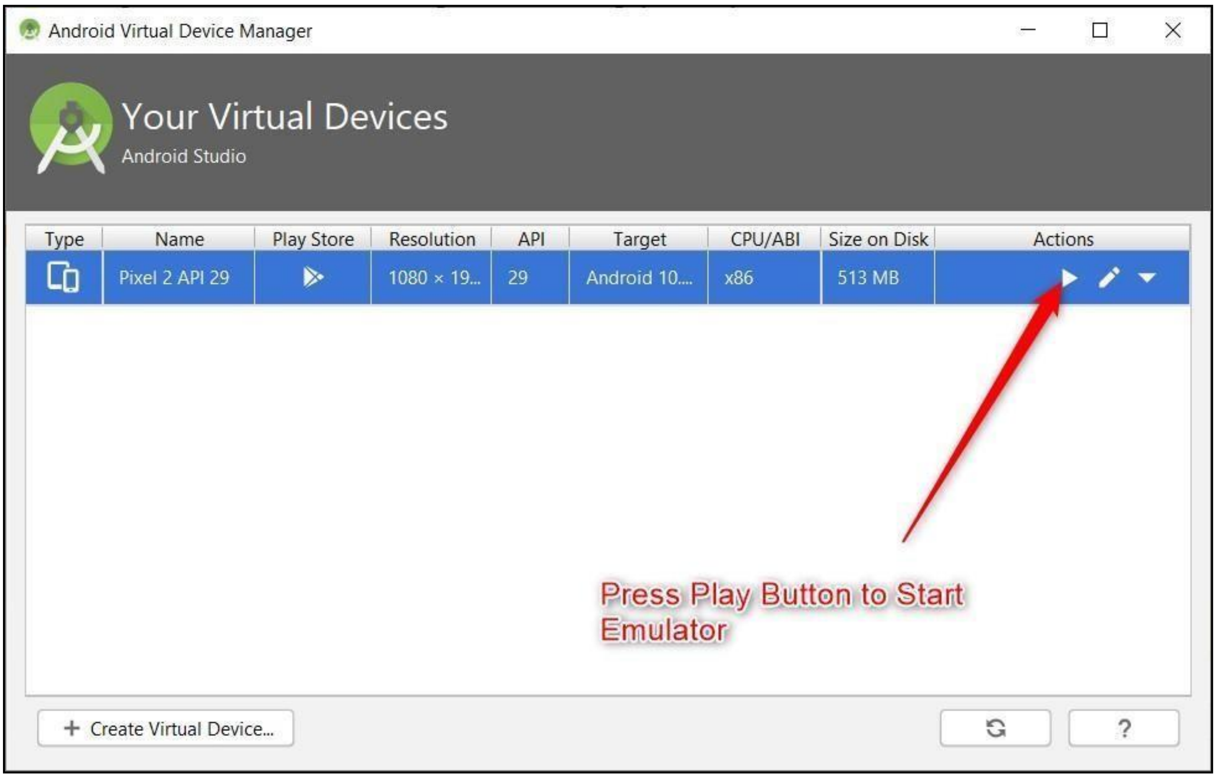
Startup orientation: Portrait

Emulated Performance — Graphics: Automatic

Device Frame: ☑ Enable Device Frame

AVD Name

The name of this AVD.



Android Virtual Device Manager

Your Virtual Devices

Android Studio

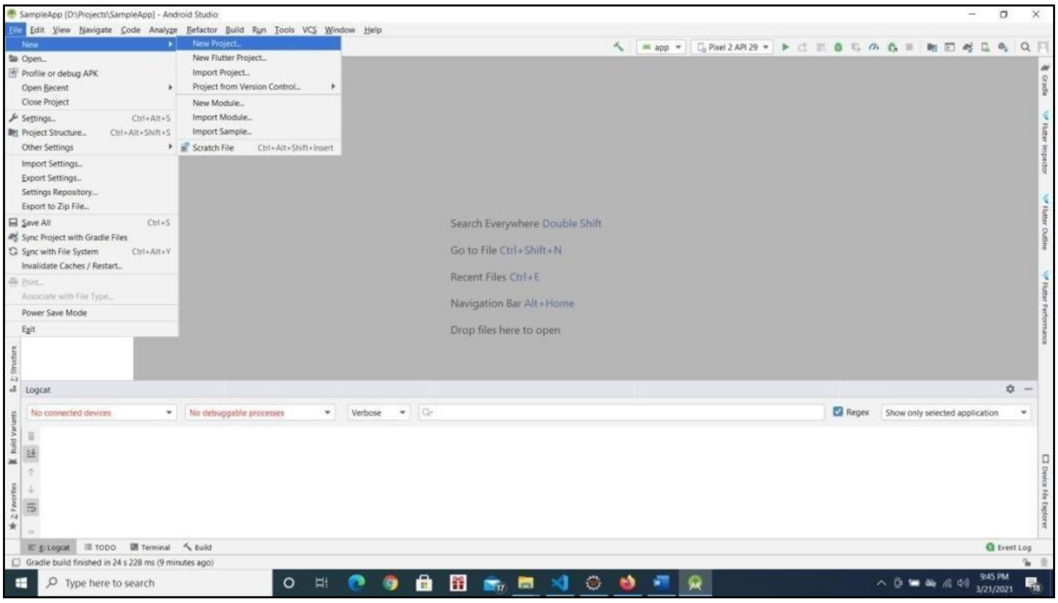
Type	Name	Play Store	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
	Pixel 2 API 29		1080 × 19...	29	Android 10....	x86	513 MB	

Press Play Button to Start Emulator

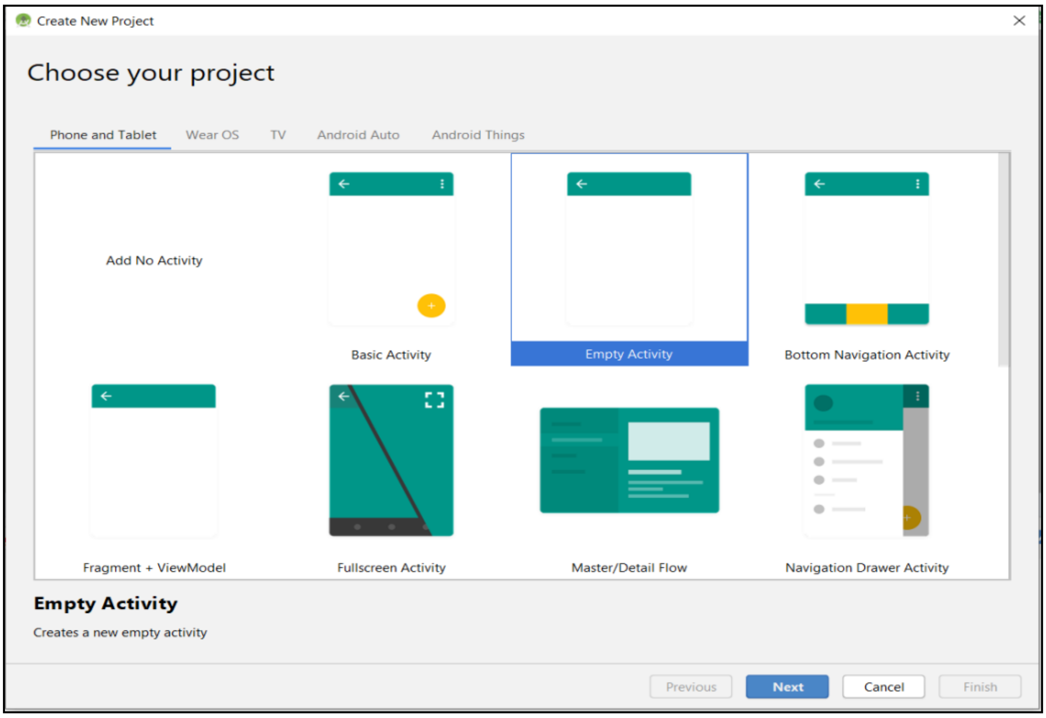
Creating a New Project inAndroid

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 ❼ New ❼ NewProject



Choose Phone and Tablet ❼ Empty Activity ❼ Press Next



Create New Project

Choose your project

Empty Activity

Creates a new empty activity

In Configure your Project Screen, Enter below details and Press Finish Button. EnterNameoftheApplication ❼ This will be application name this will be visible with HomeScreen Icon.

PackageName❼Enter package name atleast two identifier(Eg:com.example).Best Practice is 3 or more identifier (Eg:com.example.firstapp).

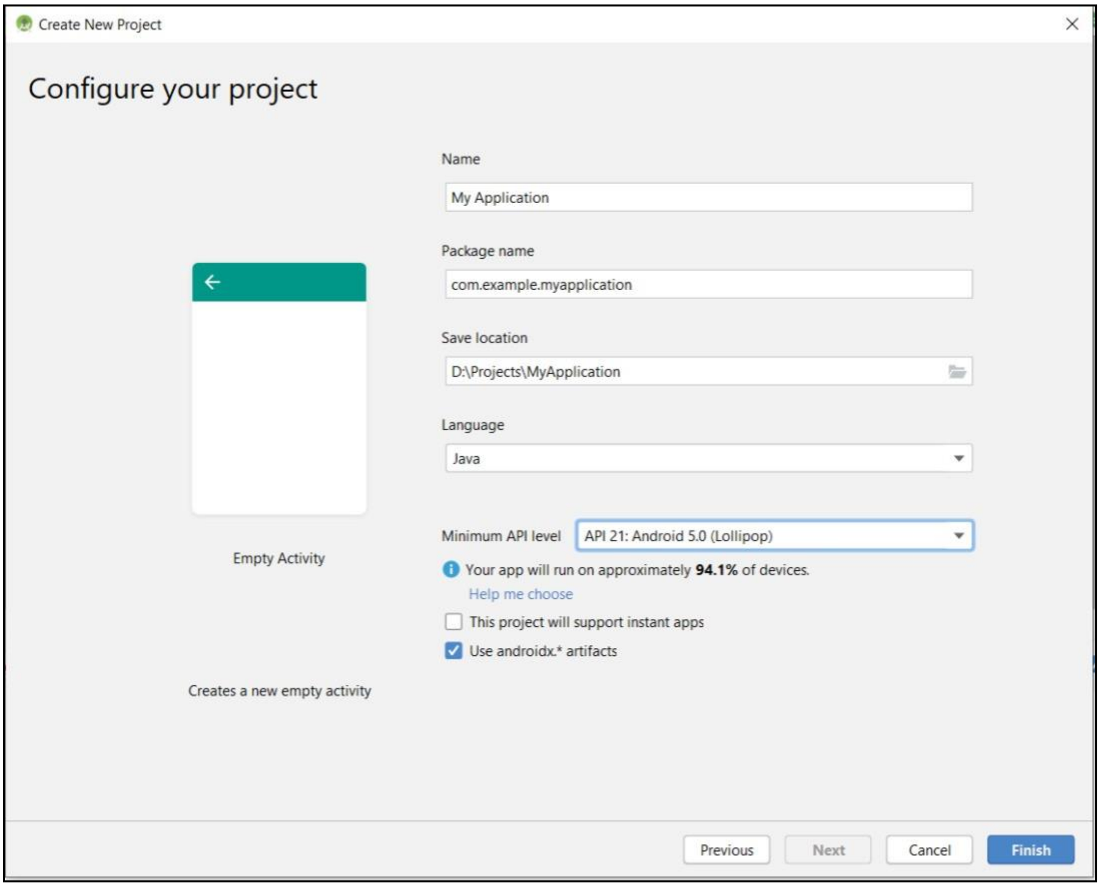
Save Location ❼ Location where

to save the ProjectLanguage

❼ Choose Java

Minimum API Level ❼ Android 5.0

Select Checkbox Use androidx.artifacts folder as below screenshot.



Name: My Application

Package name: com.example.myapplication

Save location: D:\Projects\MyApplication

Language: Java

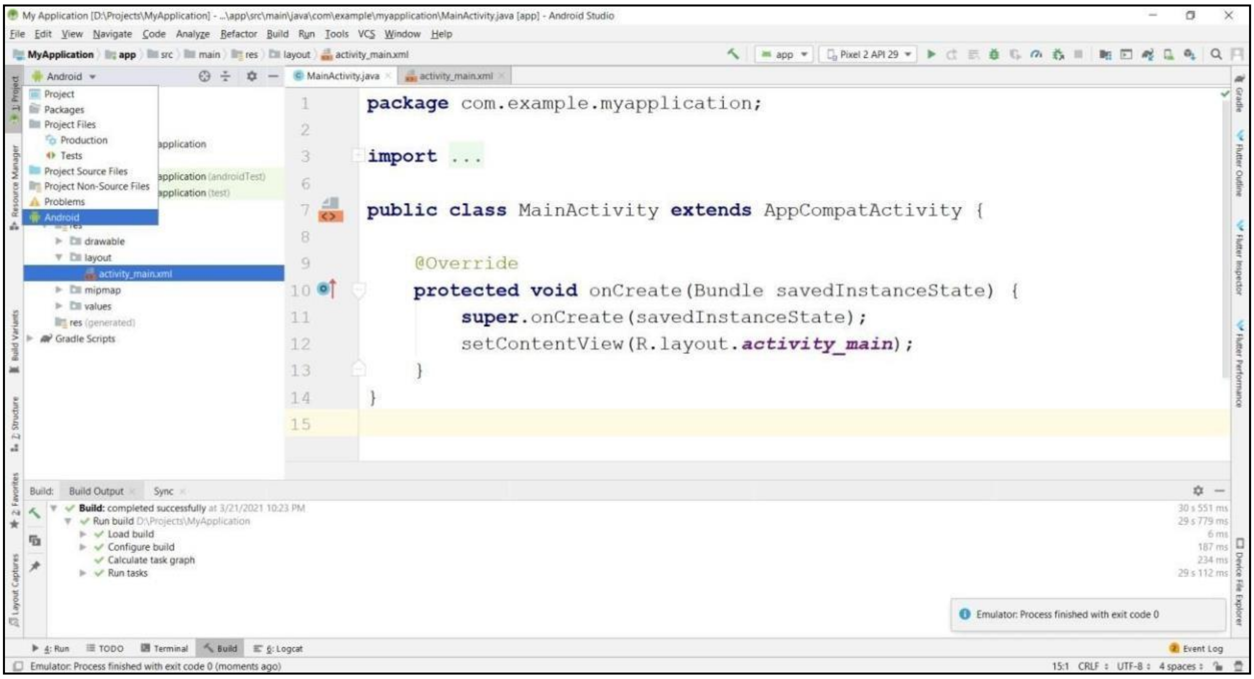
Minimum API level: API 21: Android 5.0 (Lollipop)

Your app will run on approximately **94.1%** of devices.

□ This project will support instant apps

☑ Use androidx.* artifacts

Android Project Structure:



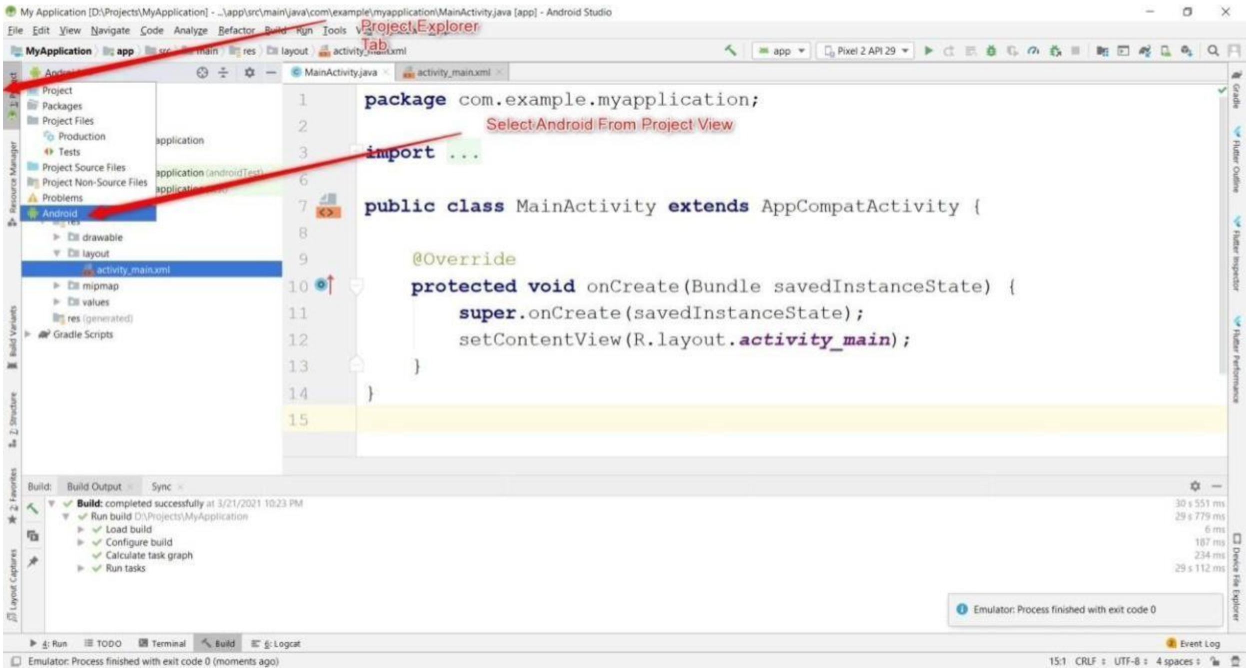
```
package com.example.myapplication;

import ...

public class MainActivity extends AppCompatActivity {

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

```
Build: completed successfully at 3/21/2021 10:23 PM      30 s 551 ms
  Run build D:\Projects\MyApplication                    29 s 779 ms
    Load build                                                 6 ms
    Configure build                                          187 ms
    Calculate task graph                                     234 ms
    Run tasks                                            29 s 112 ms
```

Emulator: Process finished with exit code 0

Emulator: Process finished with exit code 0 (moments ago)

Select Project Explorer and Select Android from Project View



Project Explorer Tab

Select Android From Project View

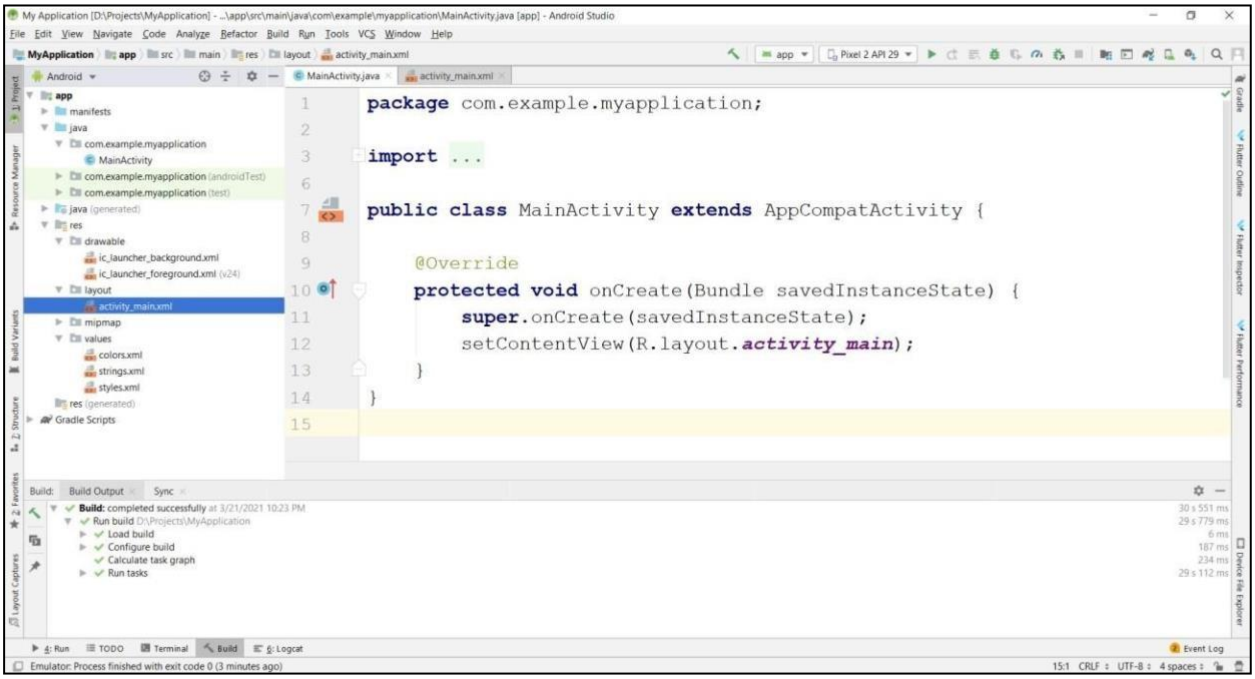
My Application [D:\Projects\MyApplication] - ...\app\src\main\java\com\example\myapplication\MainActivity.java [app] - Android Studio

```java
package com.example.myapplication;

import ...

public class MainActivity extends AppCompatActivity {

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

```
Build: Build Output  Sync
Build: completed successfully at 3/21/2021 10:23 PM    30 s 551 ms
  Run build D:\Projects\MyApplication                  29 s 779 ms
    Load build                                              6 ms
    Configure build                                       187 ms
    Calculate task graph                                  234 ms
    Run tasks                                          29 s 112 ms
```

Emulator: Process finished with exit code 0

Emulator: Process finished with exit code 0 (moments ago)



My Application [D:\Projects\MyApplication] - ...\app\src\main\java\com\example\myapplication\MainActivity.java [app] - Android Studio

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
MyApplication app src main res layout activity_main.xml
app
Pixel 2 API 29
Android
app
manifests
java
com.example.myapplication
MainActivity
com.example.myapplication (androidTest)
com.example.myapplication (test)
java (generated)
res
drawable
ic_launcher_background.xml
ic_launcher_foreground.xml (v24)
layout
activity_main.xml
mipmap
values
colors.xml
strings.xml
styles.xml
res (generated)
Gradle Scripts
MainActivity.java
activity_main.xml

```java
package com.example.myapplication;

import ...

public class MainActivity extends AppCompatActivity {

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

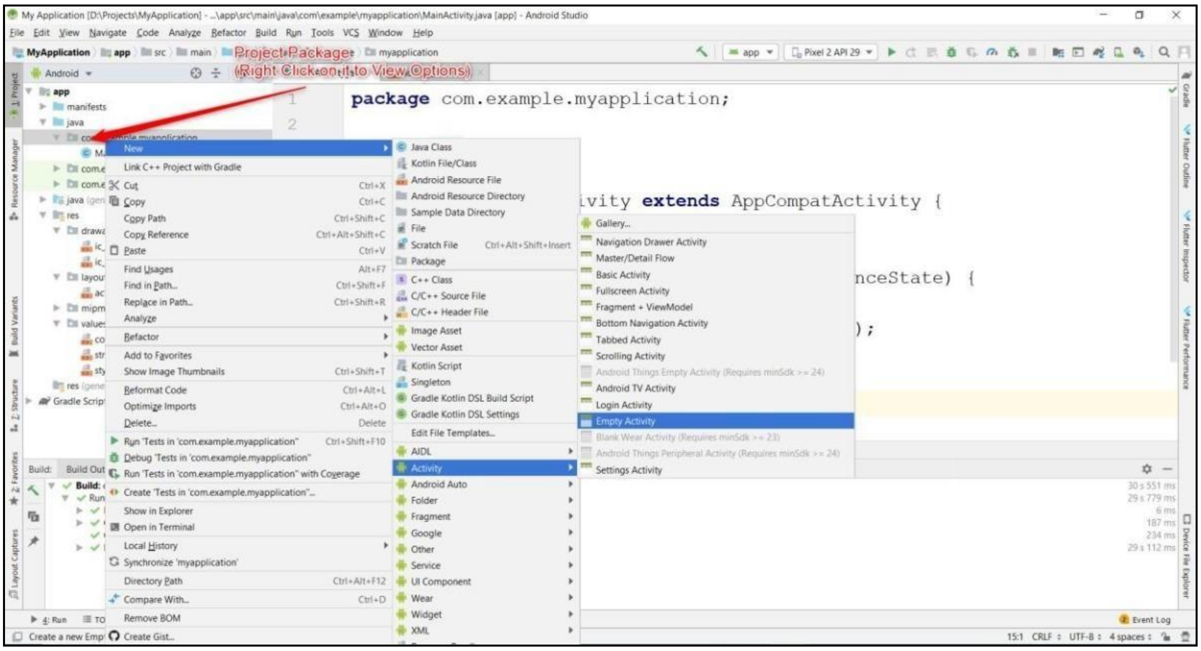
```
Build: Build Output  Sync
Build: completed successfully at 3/21/2021 10:23 PM    30 s 551 ms
  Run build D:\Projects\MyApplication                  29 s 779 ms
    Load build                                              6 ms
    Configure build                                       187 ms
    Calculate task graph                                  234 ms
    Run tasks                                          29 s 112 ms
```

Emulator: Process finished with exit code 0 (3 minutes ago)

2. Creating an Activity inAndroid

- Right Click on Package ❼ New ❼ Activity ❼ Empty Activity

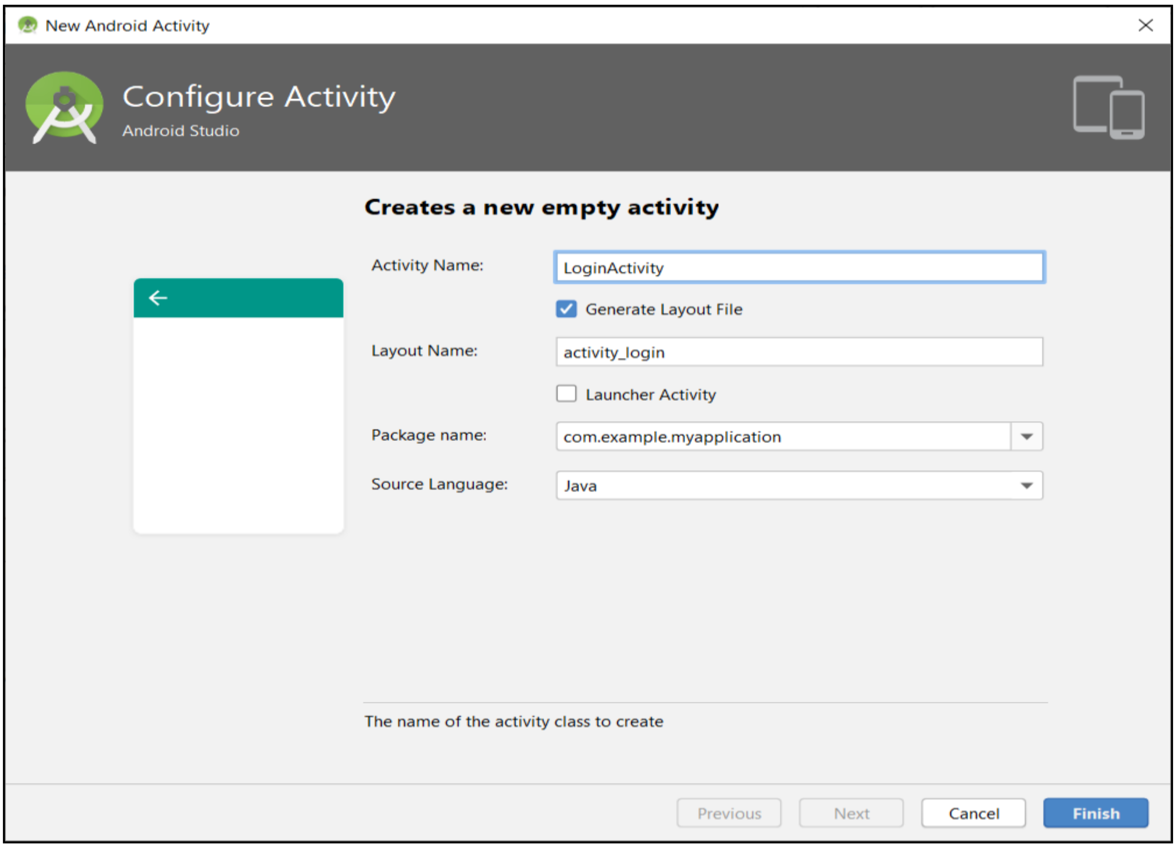
Enter Activity Name and Press Finish



Project Package
(Right Click on it to View Options)

```
package com.example.myapplication;

ivity extends AppCompatActivity {

nceState) {

);
```

My Application [D:\Projects\MyApplication] - ...\app\src\main\java\com\example\myapplication\MainActivity.java [app] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
MyApplication app src main myapplication
Android
app
manifests
java
res
Gradle Scripts
New
Link C++ Project with Gradle
Cut Ctrl+X
Copy Ctrl+C
Copy Path Ctrl+Shift+C
Copy Reference Ctrl+Alt+Shift+C
Paste Ctrl+V
Find Usages Alt+F7
Find in Path... Ctrl+Shift+F
Replace in Path... Ctrl+Shift+R
Analyze
Refactor
Add to Favorites
Show Image Thumbnails Ctrl+Shift+T
Reformat Code Ctrl+Alt+L
Optimize Imports Ctrl+Alt+O
Delete... Delete
Run 'Tests in 'com.example.myapplication'' Ctrl+Shift+F10
Debug 'Tests in 'com.example.myapplication''
Run 'Tests in 'com.example.myapplication'' with Coverage
Create 'Tests in 'com.example.myapplication''...
Show in Explorer
Open in Terminal
Local History
Synchronize 'myapplication'
Directory Path Ctrl+Alt+F12
Compare With... Ctrl+D
Remove BOM
Create Gist...
Java Class
Kotlin File/Class
Android Resource File
Android Resource Directory
Sample Data Directory
File
Scratch File Ctrl+Alt+Shift+Insert
Package
C++ Class
C/C++ Source File
C/C++ Header File
Image Asset
Vector Asset
Kotlin Script
Singleton
Gradle Kotlin DSL Build Script
Gradle Kotlin DSL Settings
Edit File Templates...
AIDL
Activity
Android Auto
Folder
Fragment
Google
Other
Service
UI Component
Wear
Widget
XML
Gallery...
Navigation Drawer Activity
Master/Detail Flow
Basic Activity
Fullscreen Activity
Fragment + ViewModel
Bottom Navigation Activity
Tabbed Activity
Scrolling Activity
Android Things Empty Activity (Requires minSdk >= 24)
Android TV Activity
Login Activity
Empty Activity
Blank Wear Activity (Requires minSdk >= 23)
Android Things Peripheral Activity (Requires minSdk >= 24)
Settings Activity
Event Log
15:1 CRLF UTF-8 4 spaces



New Android Activity

Configure Activity

Android Studio

Creates a new empty activity

Activity Name: LoginActivity

☑ Generate Layout File

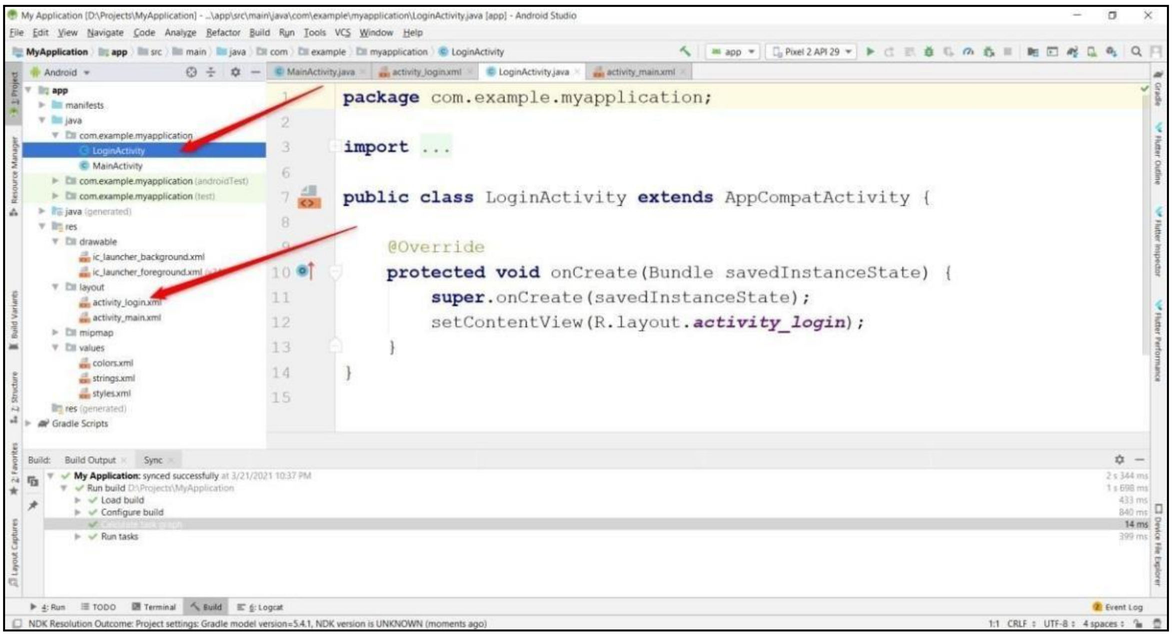
Layout Name: activity_login

□ Launcher Activity

Package name: com.example.myapplication

Source Language: Java

The name of the activity class to create



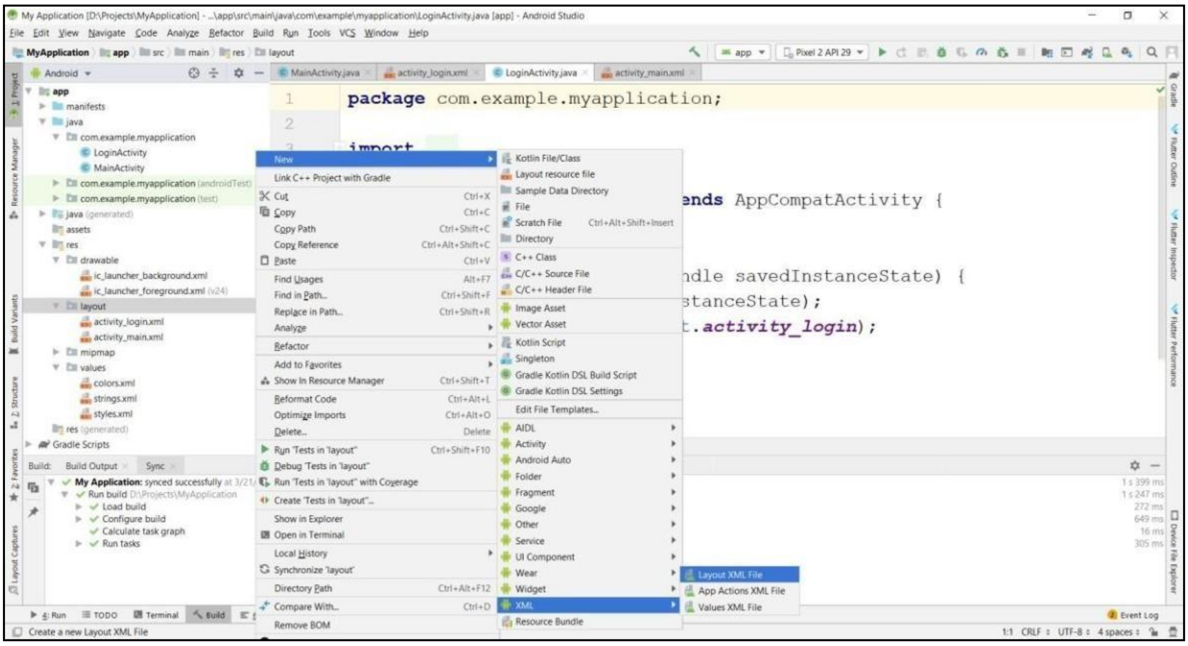
```
package com.example.myapplication;

import ...

public class LoginActivity extends AppCompatActivity {

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

```
My Application: synced successfully at 3/21/2021 10:37 PM
  Run build D:\Projects\MyApplication
    Load build
    Configure build
    Calculate task graph
    Run tasks
```

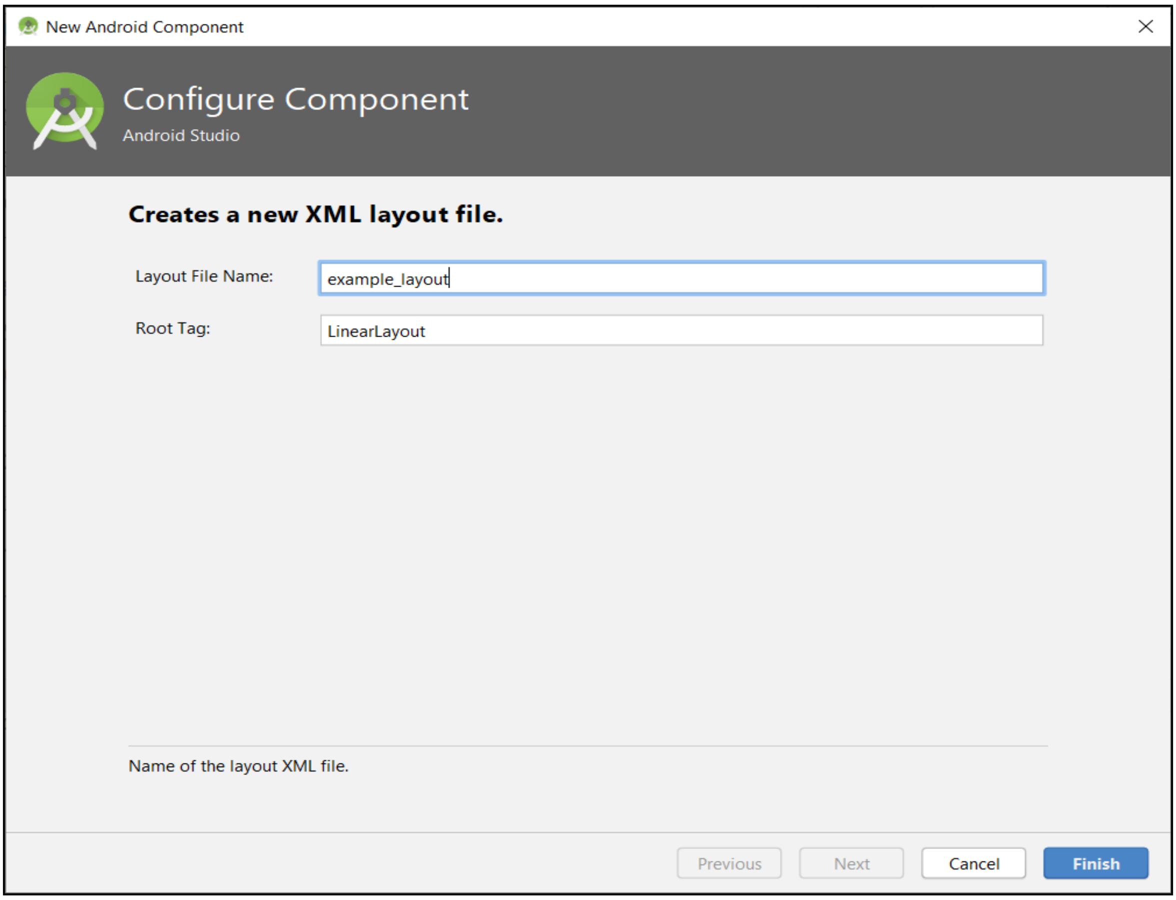
My Application [D:\Projects\MyApplication] - ...\app\src\main\java\com\example\myapplication\LoginActivity.java [app] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
app
manifests
java
com.example.myapplication
LoginActivity
MainActivity
res
drawable
layout
activity_login.xml
activity_main.xml
mipmap
values
colors.xml
strings.xml
styles.xml
Gradle Scripts
MainActivity.java
activity_login.xml
LoginActivity.java
activity_main.xml
NDK Resolution Outcome: Project settings: Gradle model version=5.4.1, NDK version is UNKNOWN (moments ago)

Creating a Layout inAndroid

Right Click on Layout Folder ➐ New ➐ XML ➐ Layout XMLFile



Enter xml file name and press Finish



Creates a new XML layout file.

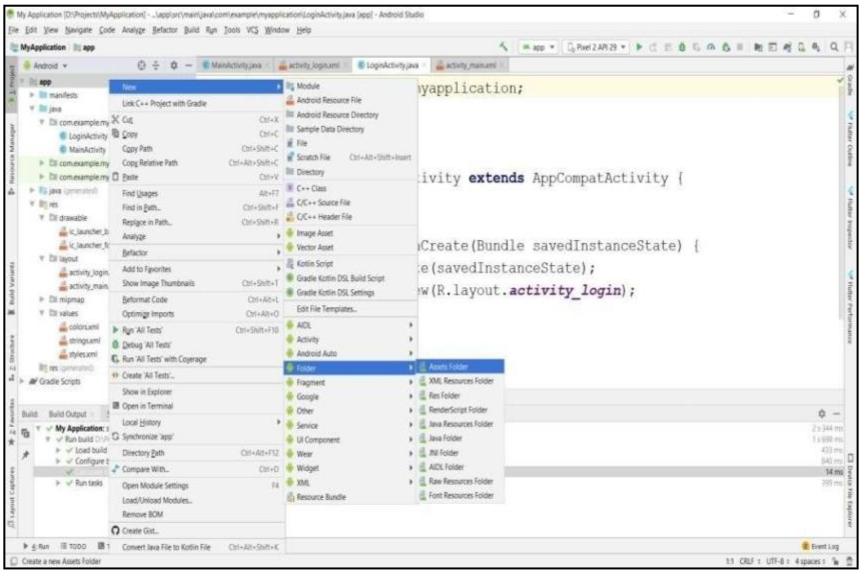
Layout File Name: example_layout

Root Tag: LinearLayout

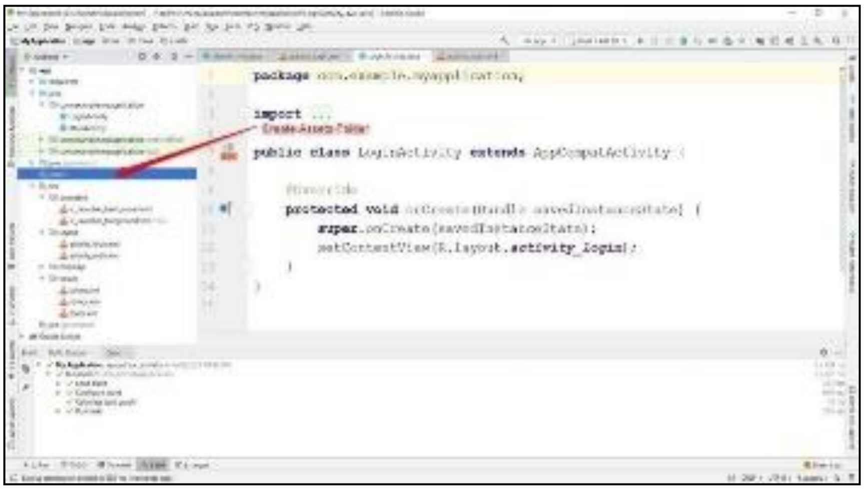
Name of the layout XML file.

Creating Assets Folder inAndroid

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



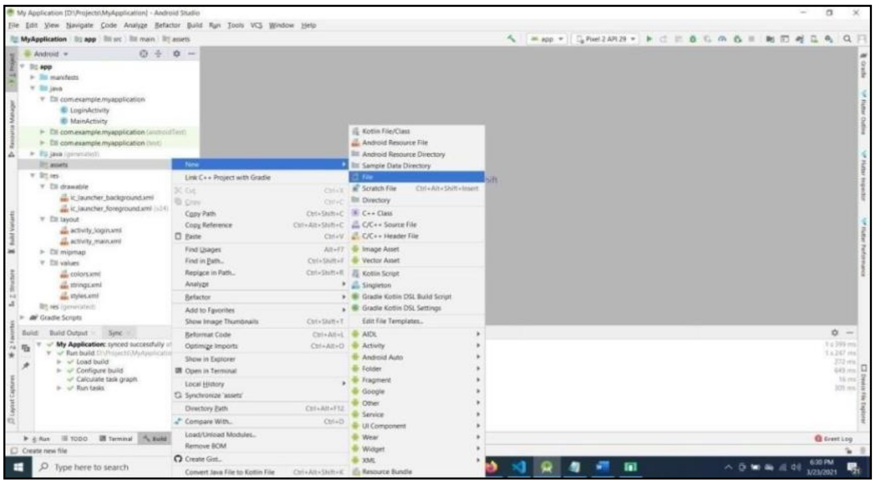
```
yapplication;
ivity extends AppCompatActivity {
Create(Bundle savedInstanceState) {
e(savedInstanceState);
w(R.layout.activity_login);
```



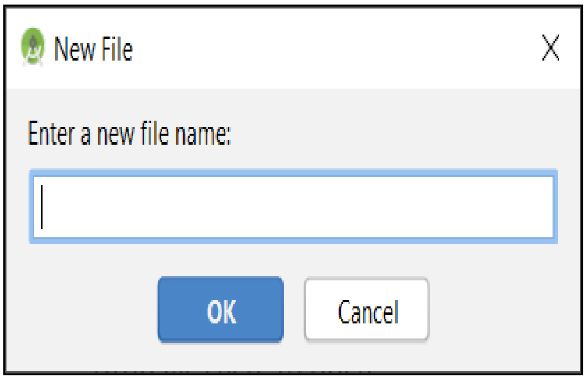
```
package com.example.myapplication;

import ...

public class LoginActivity extends AppCompatActivity {

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

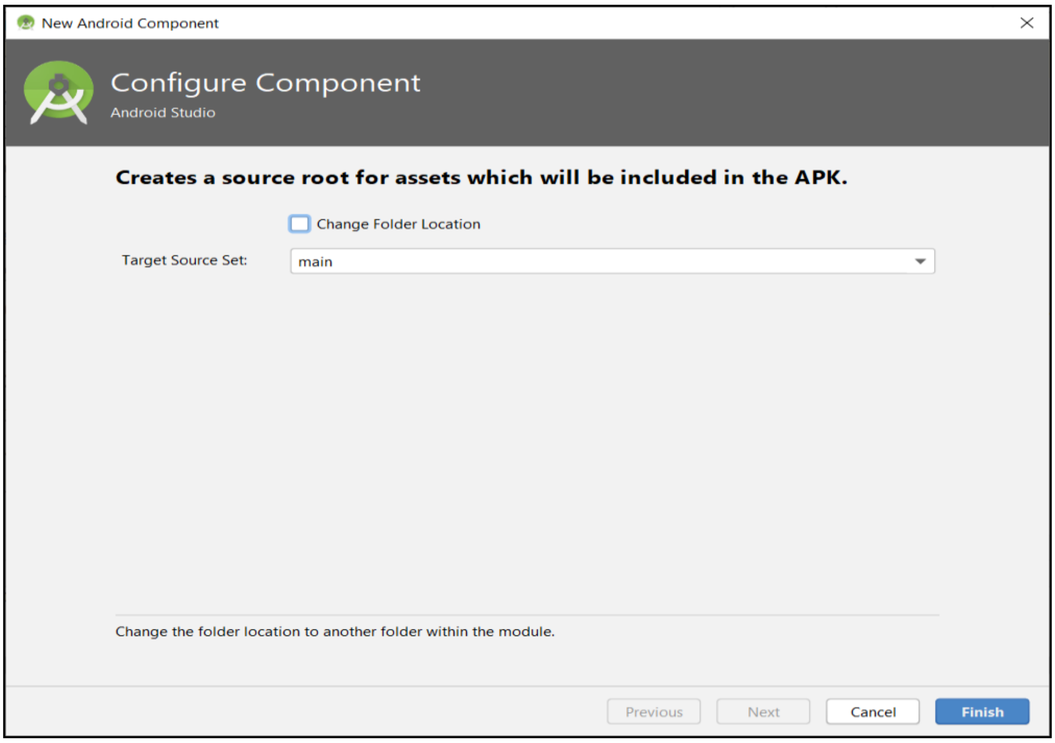
Creating File in assetsFolder:

RightClickonassetsfolder ❼ New ❼ File



Enter filename with extension (Eg: abc.xml)





Creates a source root for assets which will be included in the APK.

☐ Change Folder Location

Target Source Set: main

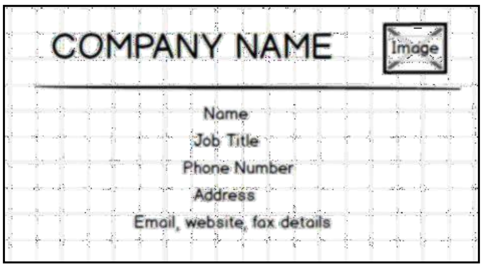
Change the folder location to another folder within the module.

Programs

PART A

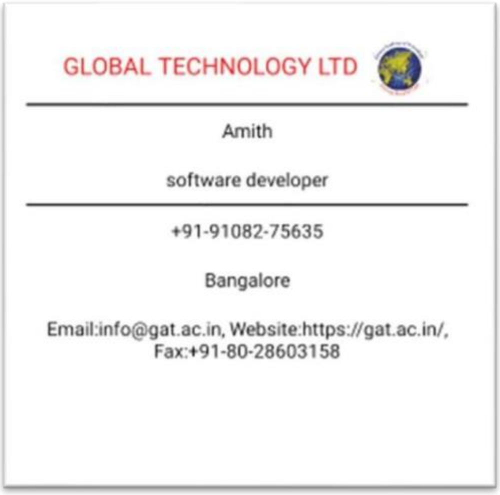
Program 1

CreateanapplicationtodesignaVisitingCard.TheVisitingcardshouldhaveacompanylogo 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 thewebsite address is to be displayed. Insert a horizontal line between the job title and the phonenumber.



1. Create a New Android Project with EmptyActivity.
2. Open activity_main.xml file from res ➐ layoutfolder,check/add Linear Layout as.
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



GLOBAL TECHNOLOGY LTD

Amith

software developer

+91-91082-75635

Bangalore

Email:info@gat.ac.in, Website:https://gat.ac.in/,
Fax:+91-80-28603158

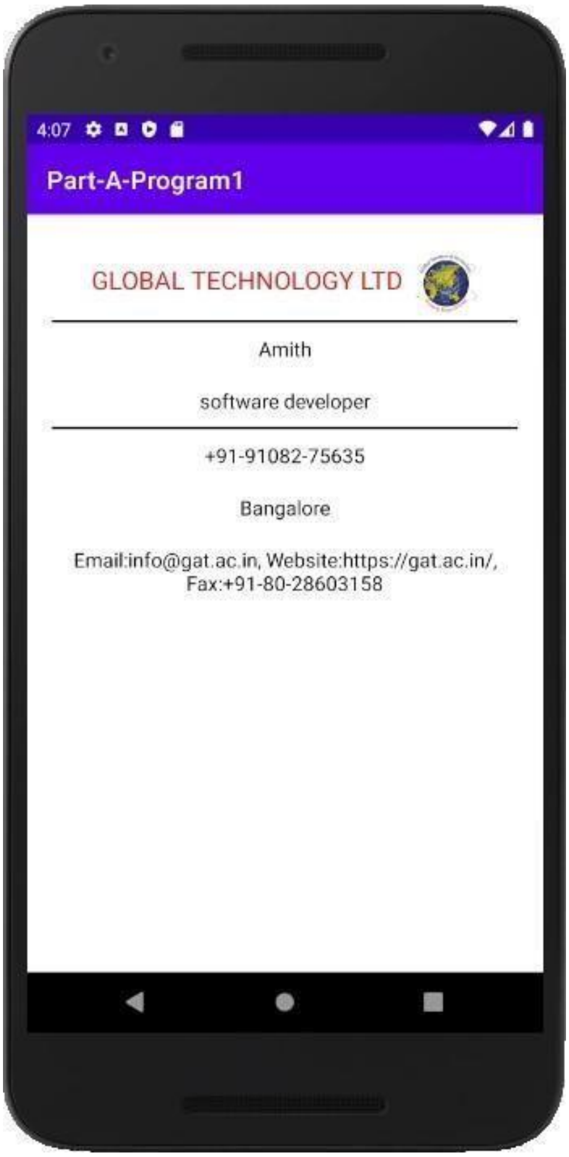
activity_main.xml

```
<?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:layout_marginStart="11dp"
android:layout_marginLeft="11dp"
android:layout_marginTop="50dp"
android:layout_marginEnd="0dp"
```

```
android:layout_marginRight="0dp"
android:layout_marginBottom="10dp"
android:orientation="vertical"
tools:context=".MainActivity">

<RelativeLayout
android:layout_width="382dp"
android:layout_height="60dp"
android:layout_marginStart="25dp"
android:layout_marginLeft="25dp"
android:layout_marginTop="80dp"
android:layout_marginEnd="20dp"
android:layout_marginRight="20dp">

<ImageView
android:id="@+id/imageView2"
android:layout_width="60dp"
android:layout_height="match_parent"
android:layout_alignWithParentIfMissing="false"
android:layout_marginStart="39dp"
android:layout_marginLeft="39dp"
android:layout_toEndOf="@+id/textView"
android:layout_toRightOf="@+id/textView"
app:srcCompat="@android:drawable/btn_star_big_on" />

<TextView
android:id="@+id/textView"
android:layout_width="wrap_content"
android:layout_height="44dp"
android:layout_alignParentBottom="true"
android:layout_marginStart="31dp"
android:layout_marginLeft="-29dp"
android:layout_marginBottom="8dp"
android:gravity="center"
android:text="TIMZ INFOTECH PVT LTD"
android:textAllCaps="true"
android:textColor="#090808"
android:textSize="20sp"
android:textStyle="bold" />

</RelativeLayout>

<View
android:id="@+id/view"
android:layout_width="match_parent"
android:layout_height="10dp"
android:background="#0C0C0C" />

<TextView
android:id="@+id/textView2"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="10dp"
```

```
android:layout_marginBottom="10dp"
android:gravity="center"
android:text="Sanjay "
android:textColor="#971414"
android:textSize="20sp" />

<TextView
android:id="@+id/textView3"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginStart="10dp"
android:layout_marginLeft="10dp"
android:layout_marginTop="5dp"
android:layout_marginBottom="10dp"
android:gravity="center"
android:paddingLeft="10dp"
android:text="Software Developer"
android:textColor="#931212"
android:textSize="16sp" />

<View
android:id="@+id/view2"
android:layout_width="match_parent"
android:layout_height="5dp"
android:background="#0C0C0C" />

<TextView
android:id="@+id/textView4"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="10dp"
android:layout_marginBottom="10dp"
android:gravity="center"
android:text="Email id:xxxx@gmail.com"
android:textColor="#971414"
android:textSize="20sp" />

<TextView
android:id="@+id/textView5"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="10dp"
android:layout_marginBottom="10dp"
android:gravity="center"
android:text="Phone:1234566789"
android:textColor="#971414"
android:textSize="20sp" />

</LinearLayout>
```

Sample Output



4:07

Part-A-Program1

GLOBAL TECHNOLOGY LTD

Amith

software developer

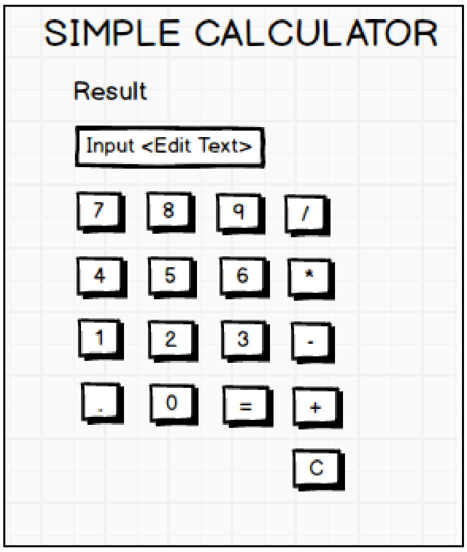
+91-91082-75635

Bangalore

Email:info@gat.ac.in, Website:https://gat.ac.in/,
Fax:+91-80-28603158

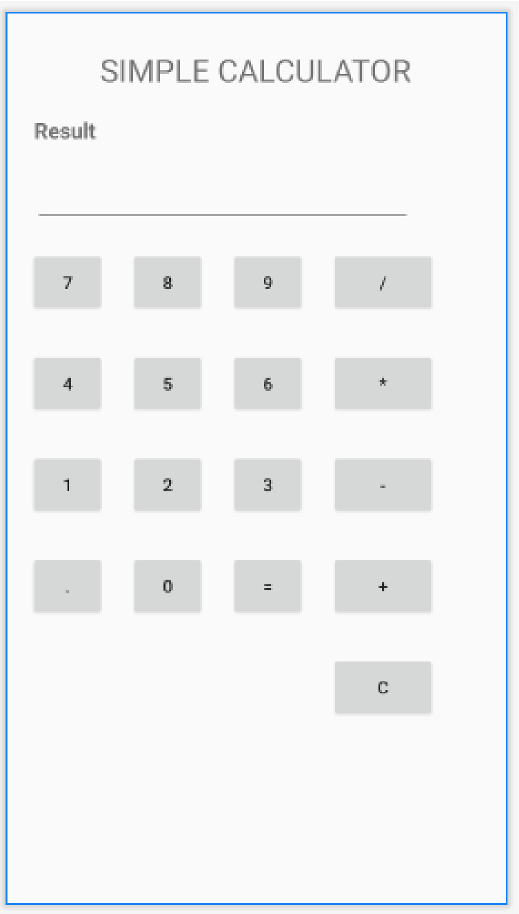
Program 2

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



1. Create a New Android Project with EmptyActivity.
2. Open activity_main.xml filefromres ➐ layout folder, check/add ConstraintLayoutastherootview.
3. Create Layout using Drag and Dropframework.
4. OpenMainActivty.javafile,OverrideonCreate()methodandbringactivity_main. xmlfile onscreen using setContentView() and bring the view references using findViewById()method.
5. Add Listeners to Button ClickEvent:
6. Create a class which implements OnClickListenerinterface.
7. Override onClick() method of OnClickListenerInterface.
8. Register the button for click event by calling setOnClickListener() method of View class and pass the object of the class that implemented OnClickListenerInterface.
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.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">


    <Button
        android:id="@+id/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"
        android:layout_marginLeft="20dp" />

    <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"
        android:layout_marginLeft="20dp"/>

    <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"
        android:layout_marginLeft="20dp"/>

    <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="SIMPLECALCULATOR"
    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>
```

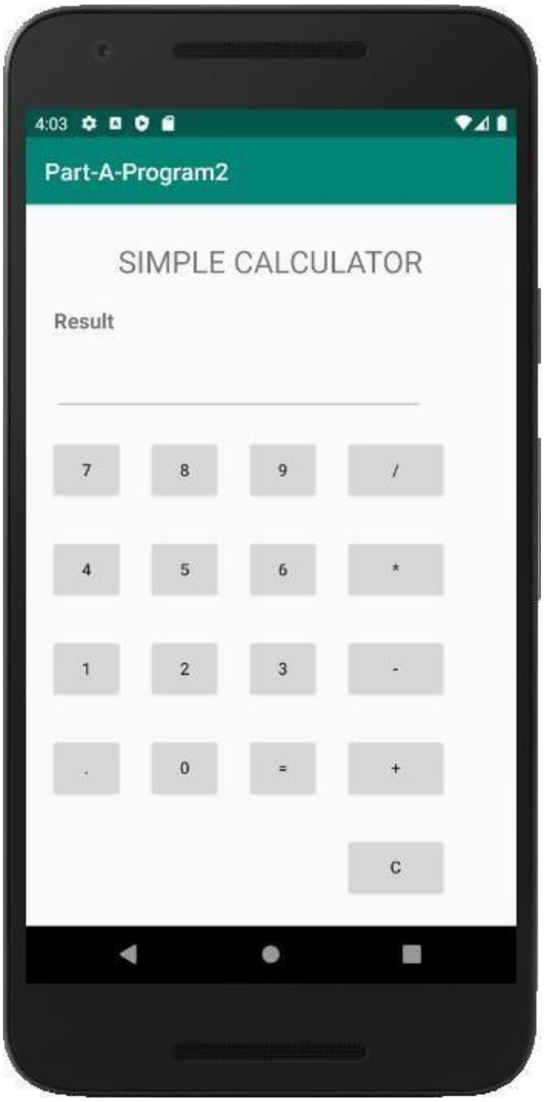
MainActivity.java

```
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 MainActivity extends AppCompatActivity implements View.OnClickListener
        {
            Button btnOne,btnTwo,btnThree,btnFour,btnFive,btnSix;
        Button     btnSeven,       btnEight,        btnNine, btnZero;
        Button btnAdd,btnSub,btnMul,btnDiv;
        Button btnClear,btnEqual,btnDot;
```

```
EditText txtResult;
   @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(), "InvalidInput", 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(), "InvalidInput", 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(), "InvalidInput", 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(), "InvalidInput", Toast.LENGTH_LONG).show();

                  }
              }
```

```
            } catch (Exception e) {
                Toast.makeText(getBaseContext(), "InvalidInput", 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

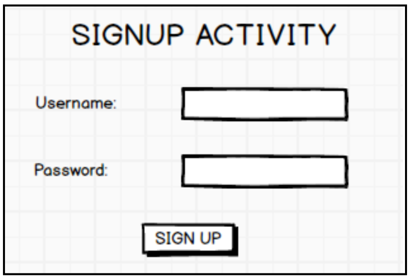


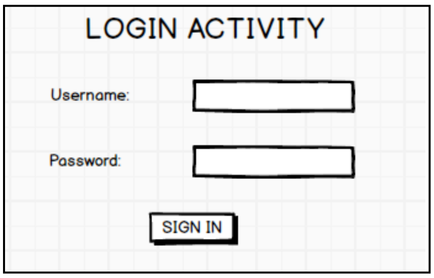
Program 3

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

- Password should contain uppercase and lowercaseletters.
- Password should contain letters andnumbers.
- Password should contain specialcharacters.
- Minimum length of the password (the default value is8).

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 naviAMCECe 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 fromone activity to another.

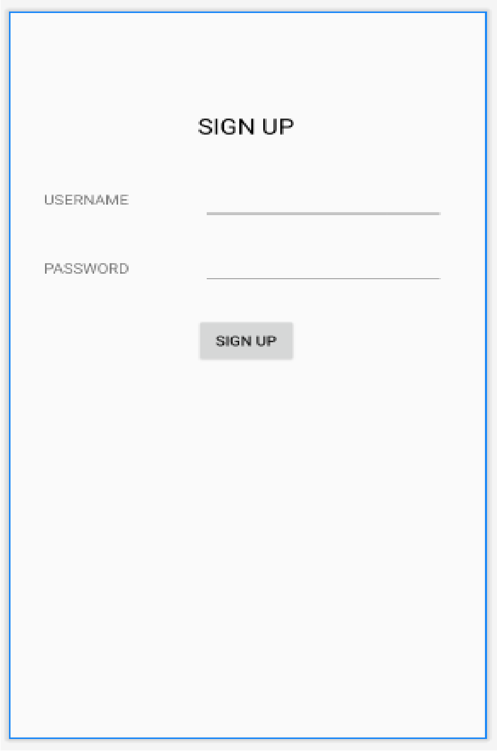


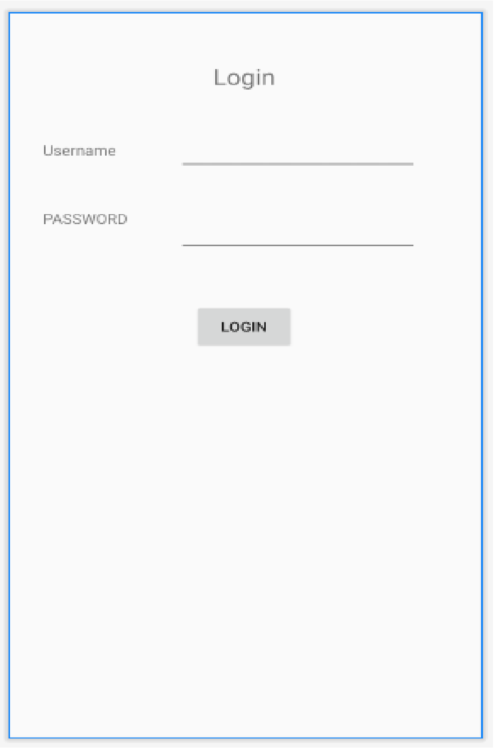


1. Create a New Android Project with EmptyActivity.
2. Open activity_main.xml filefromres ➐ layout folder, check/add ConstraintLayoutastherootview.
3. Create Signup Layout using Drag and Drop framework design thelayout.
4. Create One more Empty Activity LoginActivity using Android Studio Create ActivityFlow (Refer Android StudioTutorial)
5. Openactivity_login.xmlfilefromres ➐ layoutfolder,check/addConstraintLayout as the rootview.
6. Create Login Layout using Drag and Dropframework.
7. Add Listeners to Button ClickEvent:
   - Create a class which implements OnClickListenerinterface.
   - Override onClick() method of OnClickListenerInterface.
   - Register the button for click event by calling setOnClickListener() method of Viewclass and pass the object of the class that implemented OnClickListenerInterface.

8. Use Regular Expression "^(?=.*[A-Z])(?=.*[a-z])(?=.*\\d)(?=.*[@$!])[A-Za-z\\d@$!]{8,}$” to validate thepassword.

Design





activity_main.xml:

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

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="100dp"
        android:text="SIGNUP"
        android:textColor="@android:color/background_dark"
        android:textSize="22sp"
        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"
        android:layout_marginLeft="30dp" />

    <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_marginLeft="40dp"
        android:layout_marginRight="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"/>

    <Button
        android:id="@+id/btn_signup"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="30dp"
        android:text="SignUp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/txt_password" />

    <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"/>

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

MainActivity.java:

```
package com.example. partAprogram3;

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 androidx.appcompat.app.AppCompatActivity;

import java.util.regex.Matcher;
import java.util.regex.Pattern;
public class MainActivity extends AppCompatActivity implements View.OnClickListener
{
    EditText txtUsername;
    EditText txtPassword;
    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(), "InvalidPassword", Toast.LENGTH_LONG).show();
        }
    }


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

    }
}
```

....

activity_login.xml:

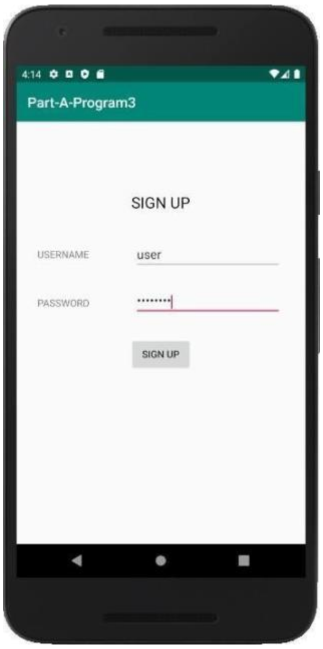
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".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="22sp"
        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_marginLeft="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_marginLeft="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>
```

....

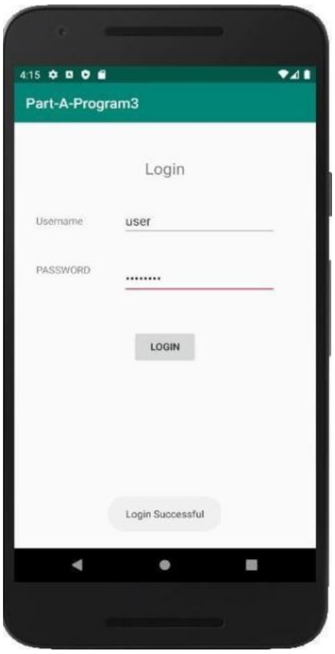
LoginActivity.java

```
package com.example.partAprogram3;

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 LoginActivity extends AppCompatActivity implements View.OnClickListener
{
    EditText txtLoginUsername;
    EditText txtLoginPassword;
    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,"LoginSuccessful",Toast.LENGTH_LONG).show();
        }
        else
        {
            count++;
            if(count==3)
        {
            btnLogin.setEnabled(false);
            Toast.makeText(this, "FailedLoginAttempts",Toast.LENGTH_LONG).show();
        }
        else
        {
            Toast.makeText(this,"LoginFailed"+count,Toast.LENGTH_LONG).show();
        }
        }
    }
}
```

Sample Output



USERNAME user
PASSWORD ••••••••

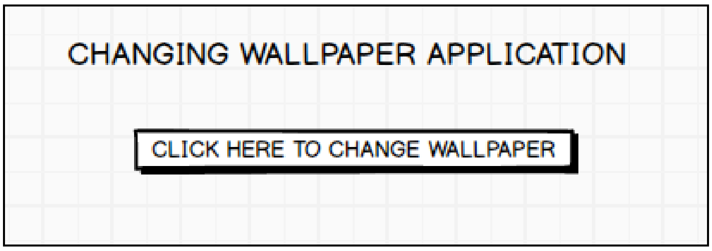


Username user
PASSWORD ••••••••

Login Successful

Program 4

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


CHANGING WALLPAPER APPLICATION
CLICK HERE TO CHANGE WALLPAPER

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

AndriodManifest.xml

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

    <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/Theme.MyDial">
        <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>
```

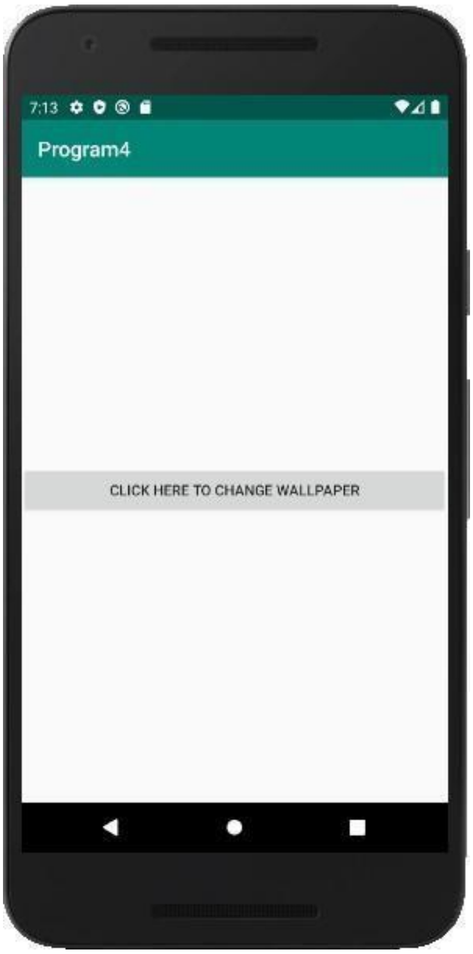
Activity_main.xml :

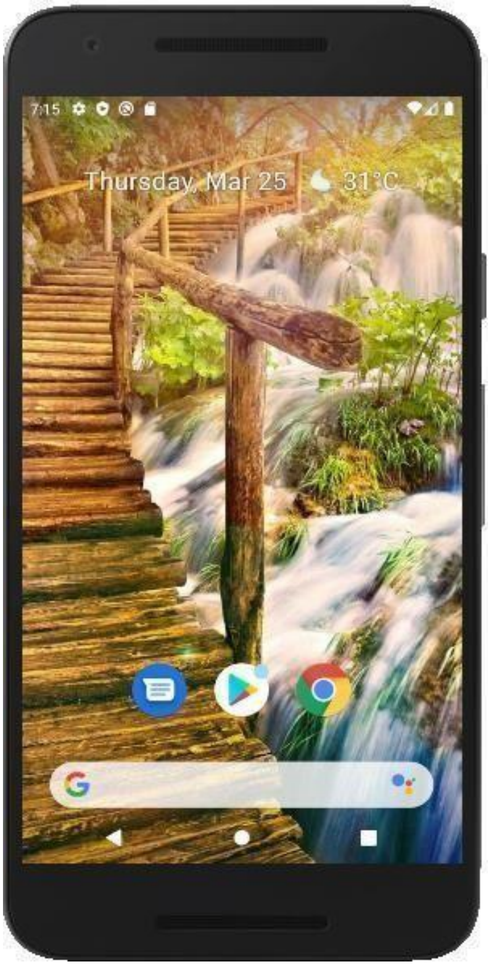
```
<?xml version="1.0"encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    tools:context=".MainActivity">
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="ClickheretoChangeWallpaper"
        android:id="@+id/btn_start_change_wallpaper"/>
</LinearLayout>
```

MainActivity.java

```
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.app.AppCompatActivity;
import android.app.WallpaperManager;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.drawable.AnimationDrawable;
import android.graphics.drawable.BitmapDrawable;
import android.graphics.drawable.Drawable;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import java.io.IOException;
import java.util.Timer;
import java.util.TimerTask;
public class MainActivity extends AppCompatActivity {
    Button changewallpaper;
    Timer mytimer;
    Drawable drawable;
    boolean running;
    WallpaperManager wpm;
    int prev=1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mytimer = new Timer();
        wpm = WallpaperManager.getInstance(this);
        changewallpaper = findViewById(R.id.button);
        changewallpaper.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                setWallpaper();
            }
        });
    }
    private void setWallpaper() {
        new Timer().schedule(new MyTimer(), 0, 6000);
        running = true;
    }
    class MyTimer extends TimerTask
    {
            @Override
            public void run() {
                if(prev==1) {
                    drawable = getResources().getDrawable(R.drawable.one);
                    prev = 2;
                }
                else if(prev==2) {
                    drawable = getResources().getDrawable(R.drawable.two);
                    prev=3;
                }
                else if(prev==3) {
                    drawable = getResources().getDrawable(R.drawable.three);
                    prev=4;
                }
                else if(prev==4) {
                    drawable = getResources().getDrawable(R.drawable.four);
                    prev=1;
                }

                Bitmap wallpaper = ((BitmapDrawable)drawable).getBitmap();
                try {
                    wpm.setBitmap(wallpaper);
                }
                catch (IOException e) {
                    e.printStackTrace();
                }
            }

    }
}
```

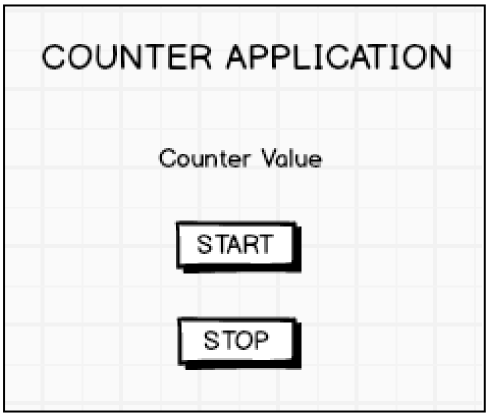
Sample Output





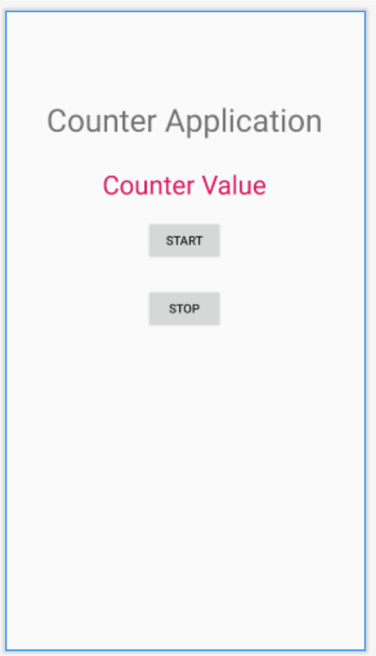
Program 5

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 EmptyActivity.
2. Openactivity_main.xmlfilefromres ➐ layoutfolder,check/addConstraintLa youtasherootv iew.
3. Create the layout design using Drag and Dropframework.
4. Add Listeners to Button ClickEvent:
   - Create a class which implements OnClickListenerinterface.
   - Override onClick() method of OnClickListenerInterface.
   - Register the button for click event by calling setOnClickListener() method of Viewclass and pass the object of the class that implemented OnClickListenerInterface.
5. Create a Thread to start the counterlogic.
6. Steps to Create aThread
   - Create a class that extends ThreadClass.
   - Override run method of Thread Class.
   - Use start() method of thread class to start thethread.
7. Create Handler class to receive message from child thread, Handler executes in MainThread.
8. Steps to CreateHandler
   - Create Object of typeHandler.
   - OverridhandleMessage() of handlerclass.
9. Pass the counter value to be displayed to thehandler.
10. Update the UI to display the counter value received fromthread.

Design



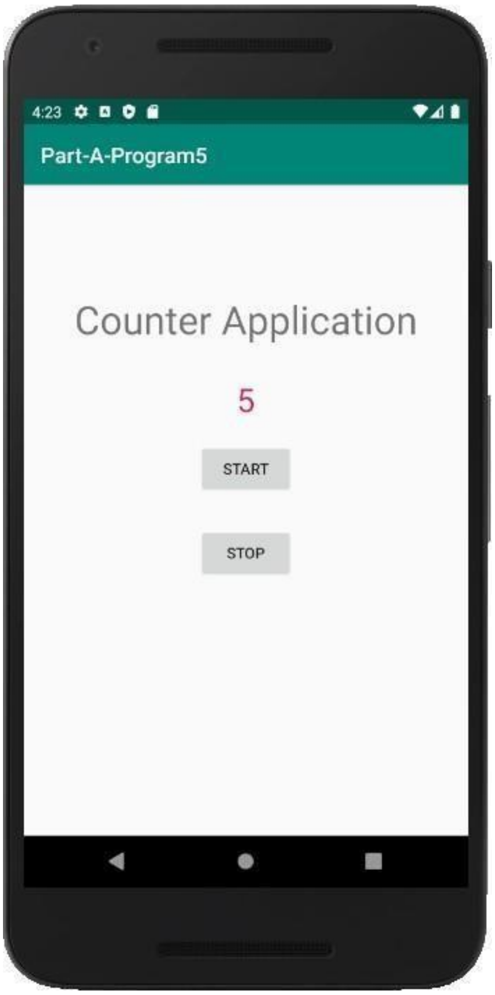
Activity_main.xml

```
<?xmlversion="1.0"encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android: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="CounterApplication"
        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="CounterValue"
        android:textColor="#6E3030"
        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.partAprogram5;

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 MainActivity extends AppCompatActivity implements View.OnClickListener
{
    TextView lblCounter;
    Button btnStart,btnStop;

    int counter=0;
    boolean running=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;
        }
    }

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

    };


    class MyCounter extends Thread
    {
        public void run()
        {
            while(running)
            {
                counter++;
                handler.sendEmptyMessage(counter);

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

Sample Output

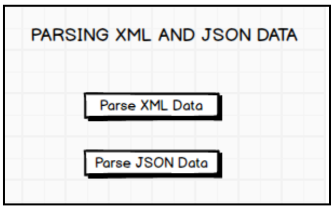


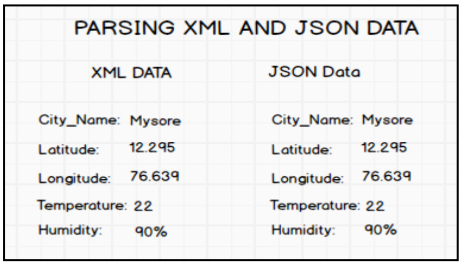
Counter Application

5

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 byside.





XML DATA		JSON Data	
City_Name:	Mysore	City_Name:	Mysore
Latitude:	12.295	Latitude:	12.295
Longitude:	76.639	Longitude:	76.639
Temperature:	22	Temperature:	22
Humidity:	90%	Humidity:	90%

1. Create a New Android Project with EmptyActivity.
2. Openactivity_main.xmlfilefromres ➐ layoutfolder,check/addConstraintLa youtastherootv iew.
3. Create the layout design using Drag and Dropframework.
4. Add Listeners to Button ClickEvent:
   - Create a class which implements OnClickListenerinterface.
   - Override onClick() method of OnClickListener Interface.
   - Register the button for click event by calling setOnClickListener() method of Viewclass and pass the object of the class that implemented OnClickListener Interface.
5. Create assets folder (Refer Section Android StudioTutorial)
6. Create **input.xml** file inside assets folder and paste the below XmlData

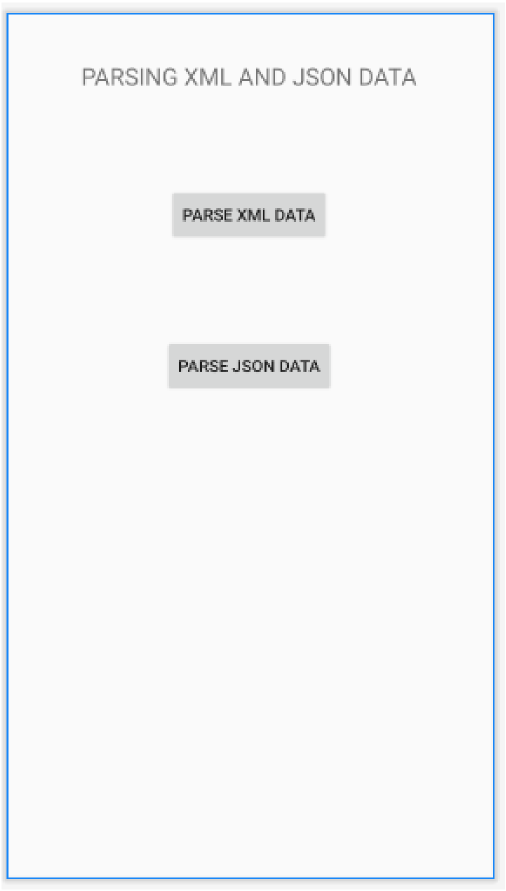
```xml
<?xml version="1.0"?>
<records>
<employee>
<city_name>Mysore</city_name>
<Latitude>12.295</Latitude>
<Longitude>76.639</Longitude>
<Temperature>22</Temperature>
<Humidity>90%</Humidity>
</employee>
</records>
```

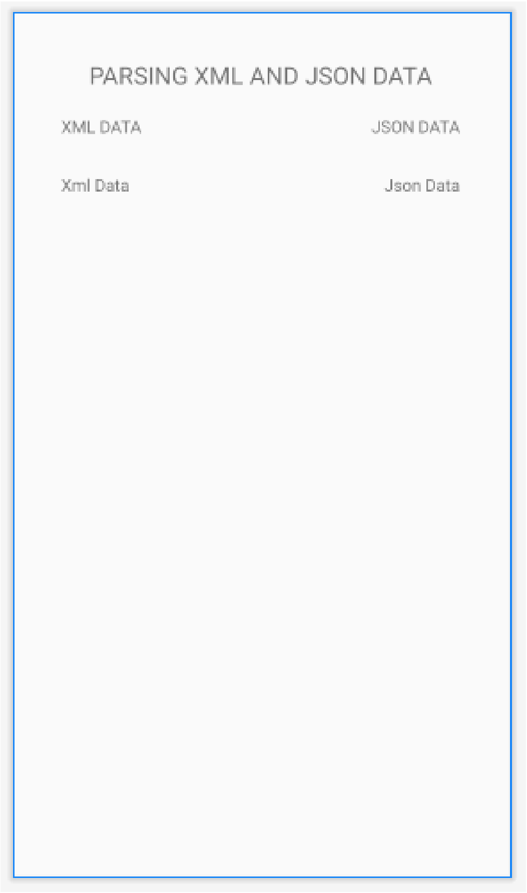
7. Create **input.json** file inside assets folder and paste the below JsonData

```json
{
"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 onscreen

Design





Activity_main.xml:

```
<?xmlversion="1.0"encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".MainActivity">
   <Button
      android:id="@+id/btn_parsexml"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginTop="80dp"
      android:text="ParseXMLData"
      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="ParseJsonData"
        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="PARSINGXMLANDJSONDATA"
        android:textSize="20sp"
        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.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ViewActivity">
    <TextView
        android:id="@+id/lbl_xml_data"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="30dp"
        android:text="XmlData"
        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="PARSINGXMLANDJSONDATA"
        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="XMLDATA"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView"
        android:layout_marginLeft="40dp" />
    <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="JSONDATA"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView"
        android:layout_marginRight="40dp" />
    <TextView
        android:id="@+id/lbl_json_data"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="30dp"
        android:text="JsonData"
        app:layout_constraintEnd_toEndOf="@+id/textView3"
        app:layout_constraintTop_toBottomOf="@+id/textView3"/>
</androidx.constraintlayout.widget.ConstraintLayout>
.................................................................
```

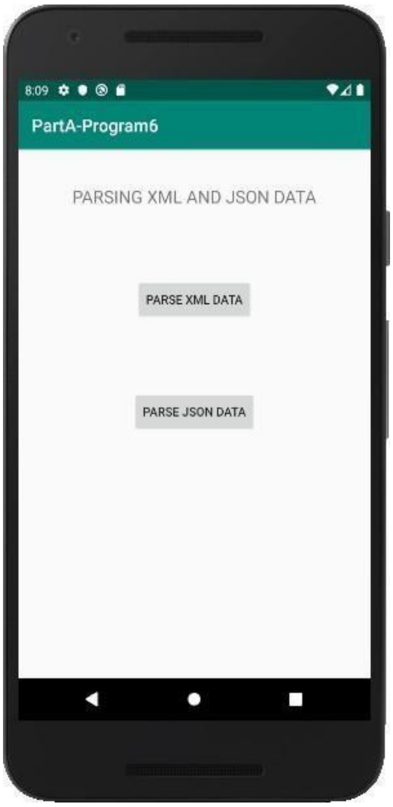
MainActivity.java

```
package com.example.programsix;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
public class MainActivity extends AppCompatActivity implements View.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);
        }
    }
}
```

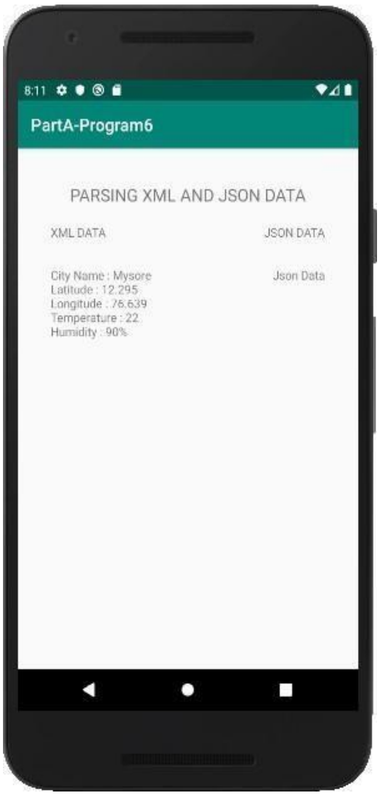
ViewActivity.java

```
package com.example.programsix;

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 ViewActivity extends AppCompatActivity
{
    TextView lblXmlData,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");
            DocumentBuilderFactory dbFactory=DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuilder;
            dBuilder = dbFactory.newDocumentBuilder();
            Document doc=dBuilder.parse(is);
            Element element=doc.getDocumentElement();
            element.normalize();
            NodeList nList=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("CityName:"+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)
    {
        NodeList nodeList=element.getElementsByTagName(tag).item(0).getChildNodes();
        Node node=nodeList.item(0);
        return node.getNodeValue();
    }
    public void parseJson()
    {
        try
        {
            InputStream inputStream=getAssets().open("input.json");
            byte[]data=new byte[inputStream.available()];
            inputStream.read(data);
                String readData=new String(data);
                JSONObject jsonObject=new JSONObject(readData);
                JSONObject jsonObject1=jsonObject.getJSONObject("employee");
                lblJsonData.setText("CityName:"+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();
        }
    }
}
```

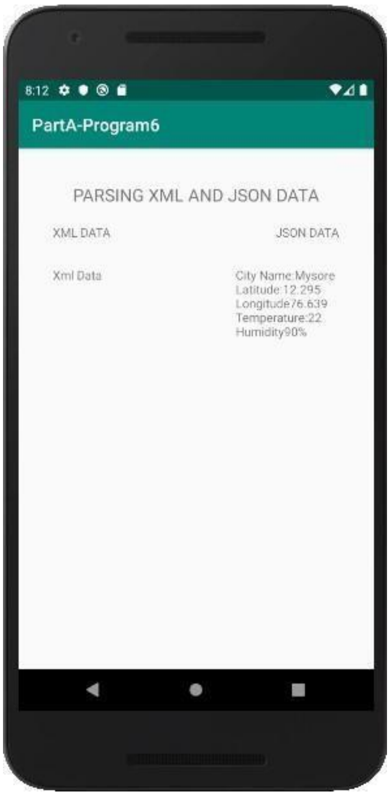
Sample Output



```
PARSING XML AND JSON DATA
```



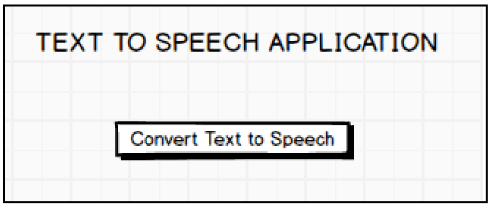
```
PARSING XML AND JSON DATA

XML DATA                        JSON DATA

City Name : Mysore              Json Data
Latitude : 12.295
Longitude : 76.639
Temperature : 22
Humidity : 90%
```



```
PARSING XML AND JSON DATA

XML DATA                        JSON DATA

Xml Data                        City Name:Mysore
                                Latitude:12.295
                                Longitude76.639
                                Temperature:22
                                Humidity90%
```

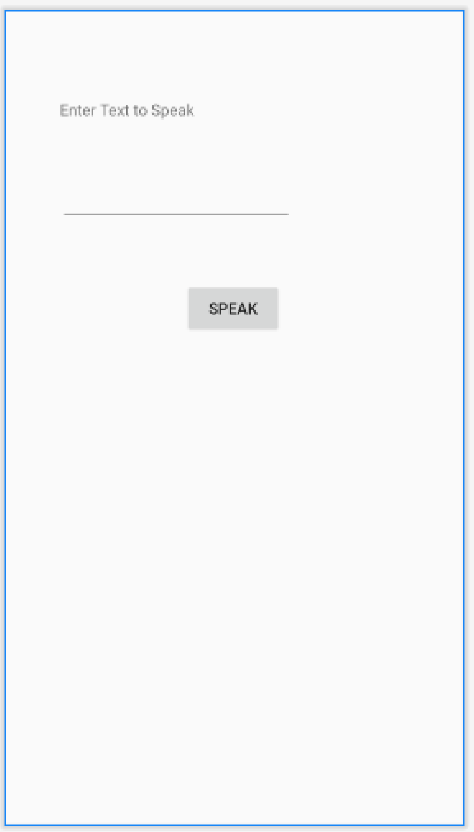
Program 7

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



1. Create a New Android Project with EmptyActivity.
2. Openactivity_main.xmlfilefromres ❼ layoutfolder,check/addConstraintLa youtasherootv iew.
3. Create the layout design using Drag and Dropframework.
4. Add Listeners to Button ClickEvent:
   - Create a class which implments OnClickListenerinterface.
   - Override onClick() method of OnClickListenerInterface.
   - Register the button for click event by calling setOnClickListener() method of Viewclass and pass the object of the class that implemented OnClickListenerInterface.
5. Initialize TextToSpeech Engine and the Language to Speak using setLanguage()method
6. Use Speak() method to speak the text passed toit.

Design



Activity_main.xml:

```
<?xmlversion="1.0"encoding="utf-8"?>
```

```
<androidx.constraintlayout.widget.ConstraintLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android: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="EnterTexttoSpeak"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="parent"
      android:layout_marginLeft="50dp" />
   <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>
```

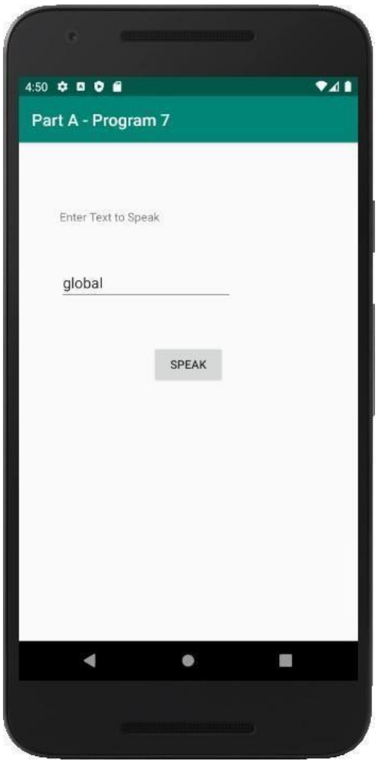
....

MainActivity.java

```
package com.example.partAprogram7;
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 MainActivity extends AppCompatActivity implements View.OnClickListener
{
   EditText txtSpeak;
```

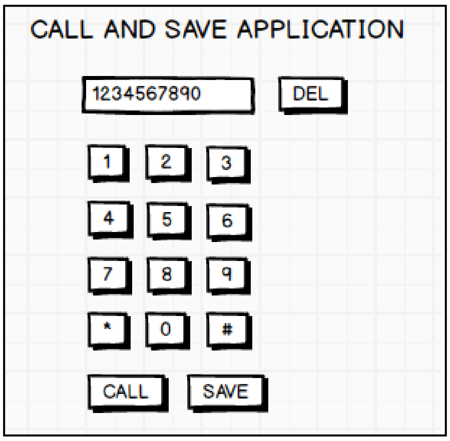
```
    Button btnSpeak;
    TextToSpeech textToSpeech;
    @Override protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        txtSpeak=(EditText)findViewById(R.id.editText);
        btnSpeak=(Button)findViewById(R.id.btn_speak);
        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



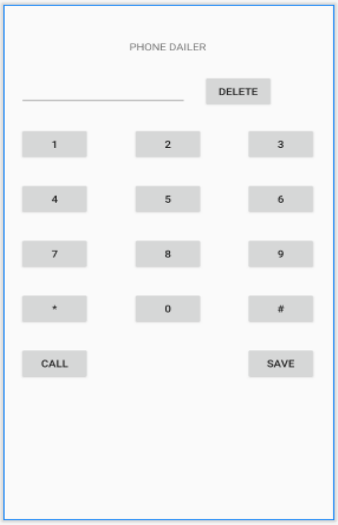
Program 8

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 EmptyActivity.
2. Open activity_main.xml filefromres ❼ layout folder, check/add ConstraintLayout astherootview.
3. Create the layout design using Drag and Dropframework.
4. Add Listeners to Button ClickEvent:
   - Create a class which implments OnClickListenerinterface.
   - Override onClick() method of OnClickListenerInterface.
   - Register the button for click event by calling setOnClickListener() method of Viewclass and pass the object of the class that implemented OnClickListenerInterface.
5. Declare uses permission android.permission.CALL_PHONE in the manifestfile.
6. Use ACTION_CALL intent name and pass the “tel:<phone-number> as URI in intentdata and start the callactivity.
7. Use intent name and pass the “Telephone Number” and “unknown” as name as intentdata call Contacts SaveActivity.

Design



Activity_main.xml:

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

<TextView android:id="@+id/textView" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:layout_marginTop="50dp" android:text="PHONEDAILER"
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"
    android:layout_marginLeft="20dp" />

<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"
    android:layout_marginLeft="20dp" />

<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"
    android:layout_marginLeft="20dp" />

<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"
    android:layout_marginRight="20dp" />
```

```
<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"
   android:layout_marginLeft="20dp" />

<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"
   android:layout_marginRight="20dp" />

<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"
   android:layout_marginLeft="20dp" />

<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"
   android:layout_marginRight="20dp" />

<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"
```

```
    android:layout_marginLeft="20dp" />

<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"
    android:layout_marginRight="20dp" />

<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"
    android:layout_marginLeft="20dp" />

<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"
    android:layout_marginRight="20dp" />
    </androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.mydial;


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 MainActivity extends AppCompatActivity implements View.OnClickListener
        {
        Button btnOne,btnTwo,btnThree,btnFour,btnFive;
        Button btnSix,btnSeven,btnEight,btnNine,btnZero;
        Button btnDel,btnStar,btnHash,btnCall,btnSave;
        EditText txtPhonenumber;
        @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");
```

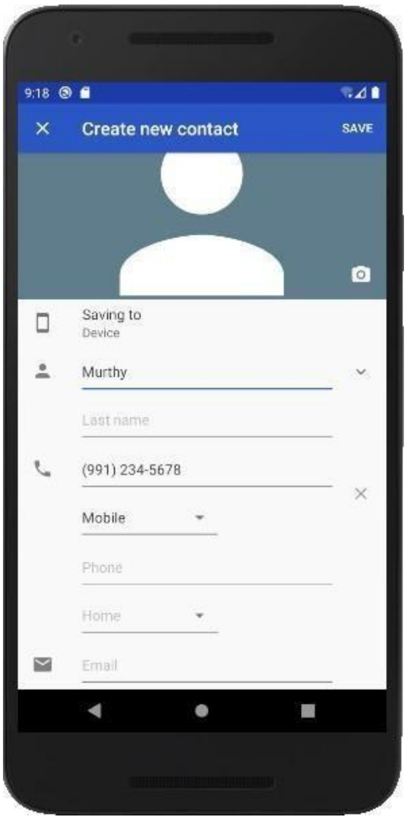
```
    else        if(v.equals(btnZero)) txtPhonenumber.append("0");
    else if(v.equals(btnStar)) txtPhonenumber.append("*");

    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.mydial">
   <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/Theme.MyDial">
        <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



Create new contact

Saving to
Device

Murthy

Last name

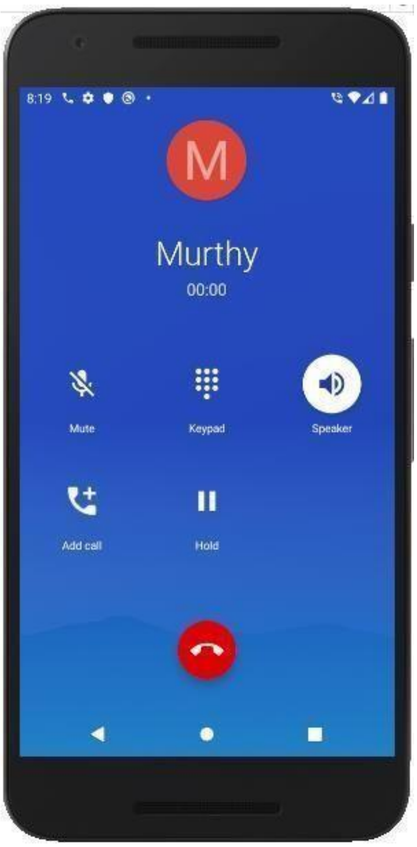
(991) 234-5678

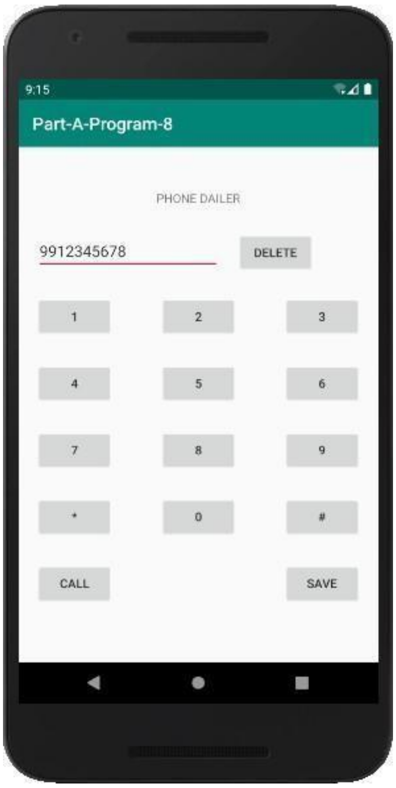
Mobile

Phone

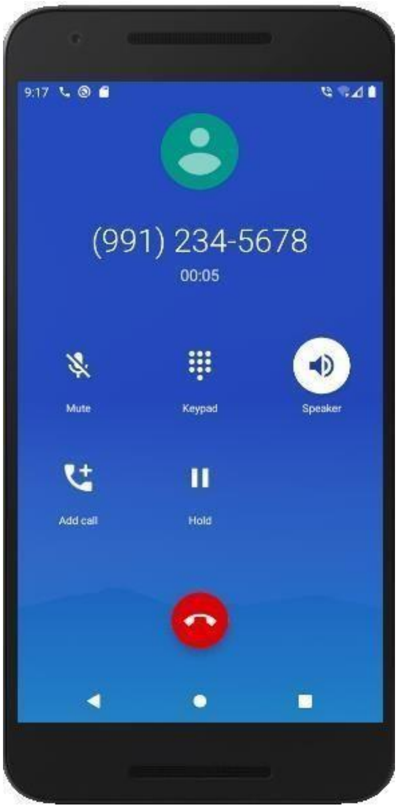
Home

Email




9:15
Part-A-Program-8
PHONE DAILER
DELETE
1
2
3
4
5
6
7
8
9
*
0
#
CALL
SAVE

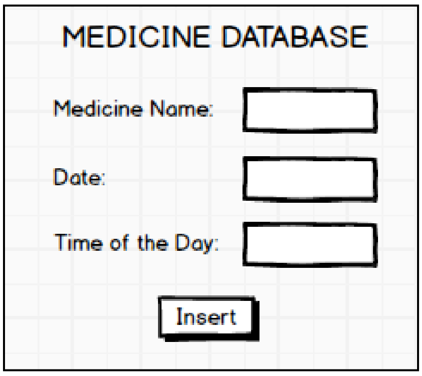
9912345678


9:17
(991) 234-5678
00:05
Mute
Keypad
Speaker
Add call
Hold

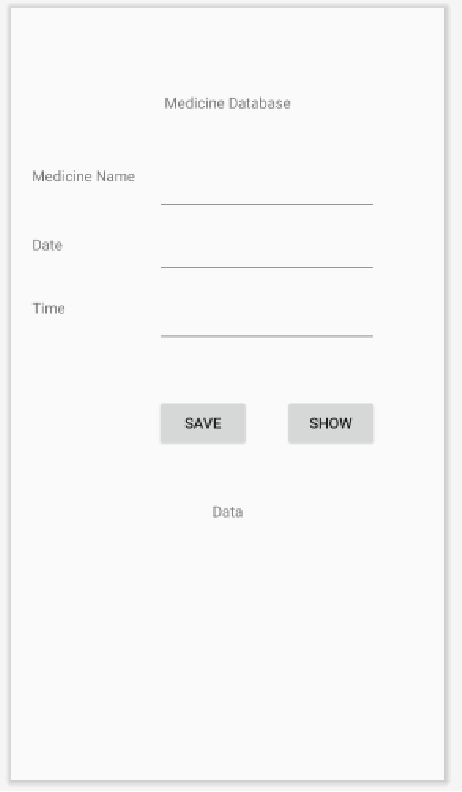
Additional Experiments

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



Design



activity_main.xml

```
<?xmlversion="1.0"encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayoutxmlns:android="http://schemas.androi
```

```
d.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tool
s" 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="MedicineDatabase"
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="MedicineName"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="@+id/txt_medici
ne_name"/>


<TextView
android:id="@+id/textView
4"
android:layout_width="wrap
_content"
android:layout_height="wra
p_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/tx
t_time"/>

<EditText
```

```
android:id="@+id/txt_medic
ine_name"
android:layout_width="wrap
_content"
android:layout_height="wra
p_content"
android:layout_marginStart=
"20dp"
android:layout_marginTop=
"50dp" android:ems="10"
android:inputType="textPers
onName"
app:layout_constraintStart_toEndOf="@+id/textView3"
app:layout_constraintTop_toBottomOf="@+id/textView2"/>

<EditText
android:id="@+id/txt_date
"
android:layout_width="wra
p_content"
android:layout_height="wra
p_content"
android:layout_marginTop=
"15dp" android:ems="10"
android:inputType="textPers
onName"
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="wra
p_content"
android:layout_height="wra
p_content"
android:layout_marginTop=
"20dp" android:ems="10"
android:inputType="textPers
onName"
app:layout_constraintStart_toStartOf="@+id/txt_date"
app:layout_constraintTop_toBottomOf="@+id/txt_date"/>

<Button
android:id="@+i
d/btn_save"
android:layout_width="wrap
_content"
android:layout_height="wra
p_content"
android:layout_marginTop=
"50dp" android:text="Save"
```

```
app:layout_constraintStart_toStartOf="@+id/txt_time"
app:layout_constraintTop_toBottomOf="@+id/txt_time"/>


<Button
ndroid:id="@+i
d/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="wra
p_content"
android:layout_height="wrap_content"
android:layout_marginTop="50dp"
android:text="Data"
app:layout_constraintEnd_toEndOf="par
ent"
app:layout_constraintStart_toStartOf="p
arent"
app:layout_constraintTop_toBottomOf="@+id/btn_save"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

MyDatabase.java

```
packagecom.example.partb_program1;

importandroid.content.Context;
importandroid.database.sqlite.SQLiteDatabase
;
importandroid.database.sqlite.SQLiteOpenHel
per;

importandroidx.annotation.Nullable;

publicclassMyDatabaseextendsSQLiteOpenHe

lper{

publicstaticStringDATABASE_NAME="medi

cine.db";

publicMyDatabase(@NullableContextcontext,@NullableStringname,@Nullable
SQLiteDatabase.CursorFactoryfactory,intversion){
super(context,name,factory,version);
        }

@Override
publicvoidonCreate(SQLiteD
atabasedb){

db.execSQL("CREATETABLEMEDICINE_NAMES(NAMETEXT,MDATETEXT,MTIME
TEXT)");

        }

@Override
publicvoidonUpgrade(SQLiteDatabasedb,intoldVersion,intnewVers
ion){


        }
}
```

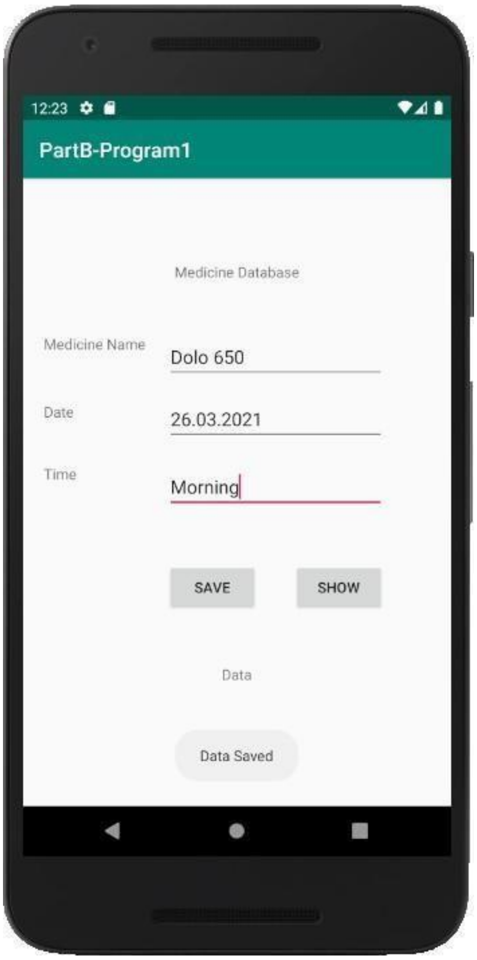
MainActivity.java

```
packagecom.example.partb_program1;

importandroidx.appcompat.app.AppCompatActivity;

importandroid.content.ContentValues;
importandroid.database.Cursor;
importandroid.database.sqlite.SQLiteDat
abase; importandroid.os.Bundle;
importandroid.view.View;
importandroid.widget.Button;
importandroid.widget.EditText;
importandroid.widget.TextView;
importandroid.widget.Toast;

importorg.w3c.dom.Text;

publicclassMainActivityextendsAppCompatActivityimplementsView.OnClickListener{

EditTexttxtMedicineName,txtDate,txtTime;
        ButtonbtnSave,btnShow;
TextViewlblData;

MyDatabas

emyDataba

se;

@Override
protectedvoidonCreate(BundlesavedInstanceState){
```

```
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
txtMedicineName=(EditText)findViewById(R.id.txt_medicine_na
me); 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_sho
w); btnShow.setOnClickListener(this);
lblData=(TextView)findViewById(R.id.lbl_da
ta);

myDatabase=newMyDatabase(getBaseContex
t(), MyDatabase.DATABASE_NAME,null,1);

}

publicvoidonClick(Viewv)
{
if(v.equals(btnSave))
{
StringmedicineName=txtMedicineName.getText().toString();
Stringdate=txtDate.getText().toString();
Stringtime=txtTime.getText().toString();

SQLiteDatabasedatabase=myDatabase.getWritable
Database(); ContentValuescv=newContentValues();
cv.put("NAME",medicineName);
cv.put("MDATE",date);
cv.put("MTIME",time);



database.insert("MEDICINE_NAMES",null,cv);
Toast.makeText(getBaseContext(),"DataSaved",Toast.LENGTH_LONG).show();
}
elseif(v.equals(btnShow))
{

SQLiteDatabasedatabase=myDatabase.getReadableDatabas
e(); Cursorcursor=database.query("MEDICINE_NAMES",
newString[]{"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



12:23

PartB-Program1

Medicine Database

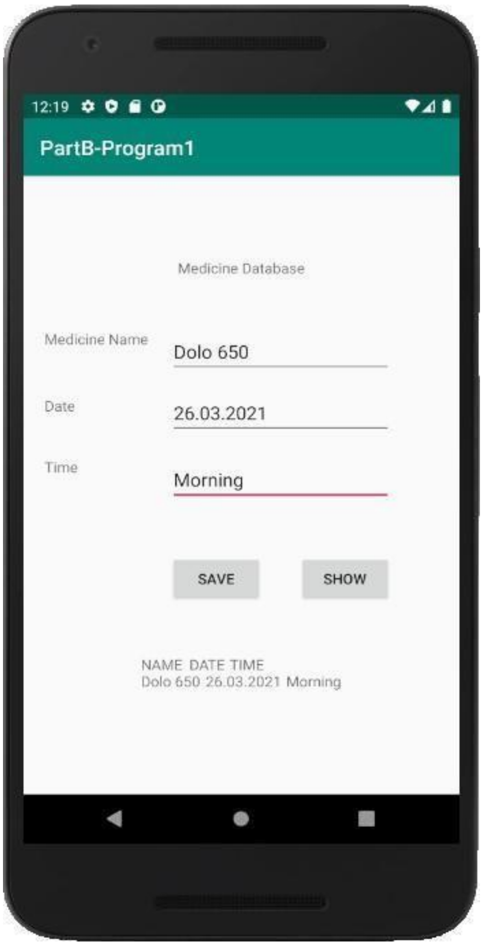
Medicine Name Dolo 650

Date 26.03.2021

Time Morning

Data

Data Saved



12:19

PartB-Program1

Medicine Database

Medicine Name Dolo 650

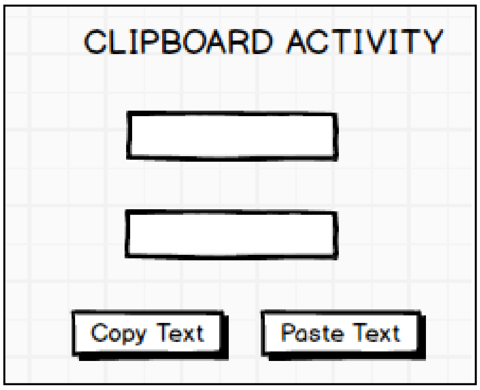
Date 26.03.2021

Time Morning

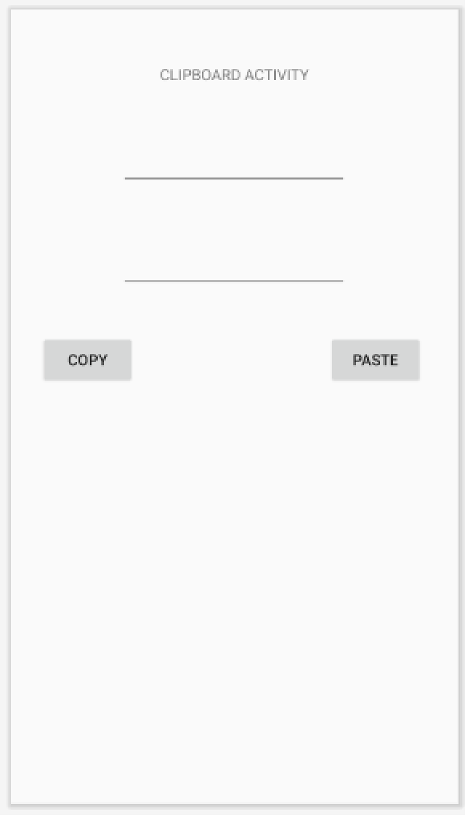
NAME DATE TIME
Dolo 650 26.03.2021 Morning

Program 2

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

```
<?xmlversion="1.0"encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayoutxmlns:android="http://schemas.androi
d.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="FileApplication" 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

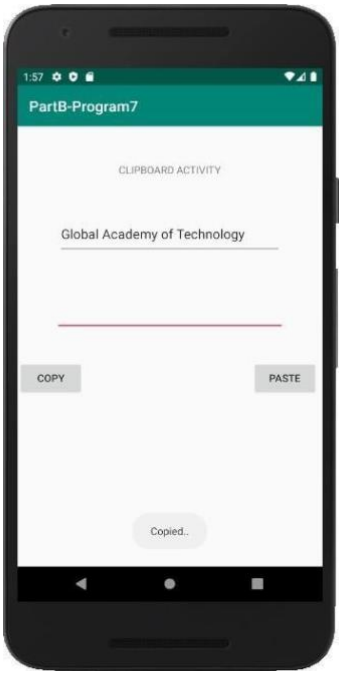
```
packagecom.example.partbprogram7;

importandroidx.appcompat.app.AppCompatActivity; importandroid.content.ClipData;
importandroid.content.ClipboardManager; importandroid.os.Bundle;
importandroid.view.View; importandroid.widget.Button; importandroid.widget.EditText;
importandroid.widget.Toast;

publicclassMainActivityextendsAppCompatActivityimplementsView.OnClickListener{
EditTexttxtCopy,txtPaste;
ButtonbtnCopy,btnPaste;
ClipboardManagermyClipboard; @Override
protectedvoidonCreate(BundlesavedInstanceState){ 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 publicvoidonClick(Viewv){ if(v.equals(btnCopy))
{
ClipDatamyClip; Stringdata=txtCopy.getText().toString();
myClip=ClipData.newPlainText("text",data); myClipboard.setPrimaryClip(myClip);
Toast.makeText(getBaseContext(),"Copied..",Toast.LENGTH_LONG).show();
}

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

AndroidManifest.xml

```
<?xmlversion="1.0"encoding="utf-8"?>
<manifestxmlns: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">
<activityandroid:name=".MainActivity">
<intent-filter>
<actionandroid:name="android.intent.action.MAIN"/>
<categoryandroid:name="android.intent.category.LAUNCHER"/>
</intent-filter>
</activity>
</application>
</manifest>
```

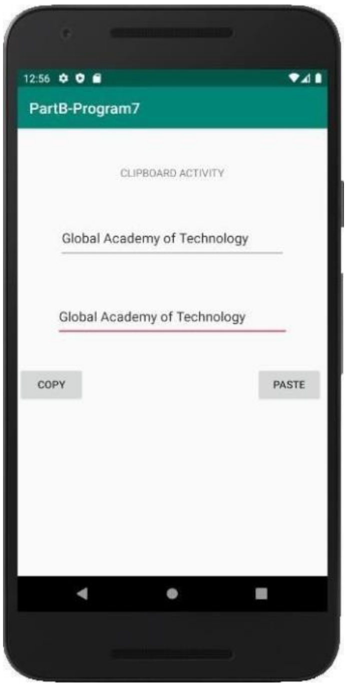
Sample Output



CLIPBOARD ACTIVITY

Global Academy of Technology

Copied..



CLIPBOARD ACTIVITY

Global Academy of Technology

Global Academy of Technology

Program 3

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 amountP = The Car loan

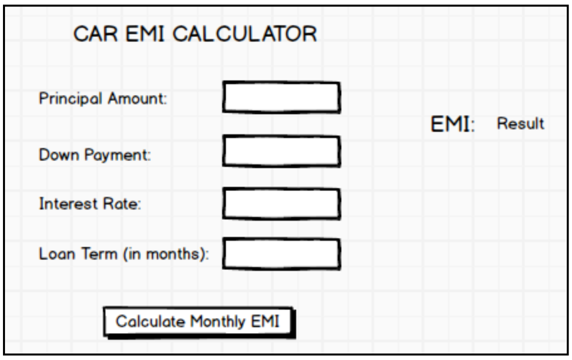
Principal Amount

r = The interest rate value computed on a

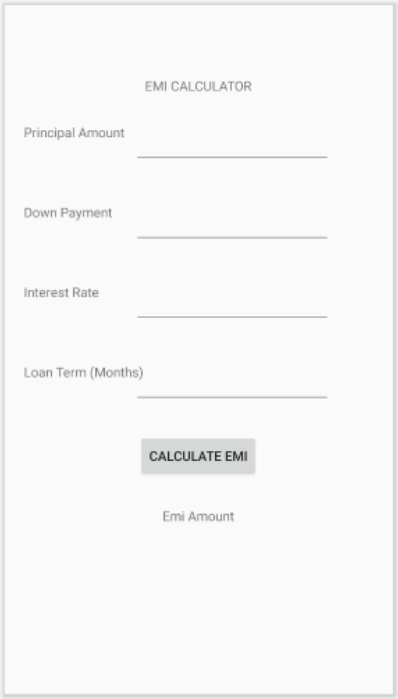
monthly basisn = The loan tenure in the

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, InterestRate,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

```
<?xmlversion="1.0"encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayoutxmlns:android="http://schemas.androi
d.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/r
es-auto"
xmlns:tools="http://schemas.android.com/tool
s" 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="EMICALCULATOR"
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="PrincipalAmount"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/t
extView"/>

<EditText
android:id="@+id/txt_princi
pal"
android:layout_width="wrap
_content"
android:layout_height="wra
p_content"
android:layout_marginStart=
"10dp"
android:layout_marginTop=
"30dp" android:ems="10"
android:inputType="textPers
onName"
app:layout_constraintStart_toEndOf="@+id/textView2"
app:layout_constraintTop_toBottomOf="@+id/textView"/>

<TextView
android:id="@+id/downpay
ment"
android:layout_width="wrap
_content"
android:layout_height="wra
p_content"
```

```
android:text="DownPaymen
t"
app:layout_constraintStart_toStartOf="@+id/textView2"
app:layout_constraintTop_toTopOf="@+id/txt_downnpayment"/>

<EditText
android:id="@+id/txt_down
npayment"
android:layout_width="wrap
_content"
android:layout_height="wra
p_content"
android:layout_marginTop=
"40dp" android:ems="10"
android:inputType="textPers
onName"
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="InterestRate"
app:layout_constraintStart_toStartOf="@+id/downpayme
nt"
app:layout_constraintTop_toTopOf="@+id/txt_interestrat
e"/>


<EditText
android:id="@+id/txt_interestra
te"
android:layout_width="wrap_c
ontent"
android:layout_height="wrap_c
ontent"
android:layout_marginTop="40
dp" android:ems="10"
android:inputType="textPerson
Name"
app:layout_constraintStart_toStartOf="@+id/txt_downnpayment"
app:layout_constraintTop_toBottomOf="@+id/txt_downnpayment"/>

<TextView
android:id="@+id/textVie
w5"
android:layout_width="13
0dp"
android:layout_height="3
3dp"
android:layout_marginTo
p="8dp"
android:text="LoanTerm(
Months)"
app:layout_constraintStart_toStartOf="@+id/textView4"
app:layout_constraintTop_toTopOf="@+id/txt_termmonths"/>

<EditText
android:id="@+id/txt_termmon
ths"
android:layout_width="wrap_c
ontent"
android:layout_height="wrap_c
ontent"
android:layout_marginStart="2
0dp"
android:layout_marginTop="32
dp" android:ems="10"
```

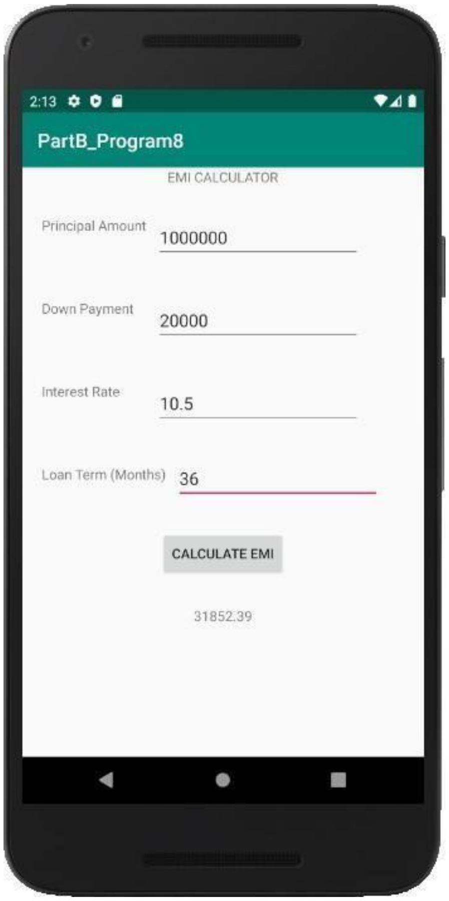
```
android:inputType="textPerson
Name"
app:layout_constraintStart_toStartOf="@+id/txt_interestrate"
app:layout_constraintTop_toBottomOf="@+id/txt_interestrate"/>

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

<TextView
android:id="@+id/lbl_emiamou
nt"
android:layout_width="wrap_c
ontent"
android:layout_height="wrap_c
ontent"
android:layout_marginTop="30
dp" android:text="EmiAmount"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"


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

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

MainActivity.java

```
packagecom.example.partb_program8;

importandroidx.appcompat.app.AppCompat
Activity; importandroid.os.Bundle;
importandroid.view.View;
importandroid.widget.Button;
importandroid.widget.EditText;
importandroid.widget.TextView;
importandroid.widget.Toast;
```

```
importjava.text.DecimalFormat;
importjava.util.logging.SimpleForm
atter;

publicclassMainActivityextendsAppCompatActivityimplementsView.OnClickListe

ner{ EditTexttxtPrinicple,txtDownPayment,txtInterestRate,txtLoanTerm;

ButtonbtnCalculate;

TextViewlblResult;

@Override
protectedvoidonCreate(BundlesavedInstanceState){
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
txtPrinicple=(EditText)findViewById(R.id.txt_princip
al);
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);

}

publicvoidonClick(Viewv)
{
try
{

DecimalFormatform
atter=new
DecimalFormat("#0.
00");
doubleprinicipleAmount=
Double.parseDouble(txtPri
nicple.
getText().toString());
doubledownPayment=Double.parseDouble(txtDownPayment.getText().toString());

prinicipleAmount=prinicipleAmount-downPayment;
doubleinterestRate=Double.parseDouble(txtInterestRate.getText().toString());
interestRate=interestRate/(12*100);
doubleloanTerm=Double.parseDouble(txtLoanTerm.getText().toString());
doubleemi=prinicipleAmount*
                                (interestRate*Math.pow((1+interestRate),loanTerm))
                                /(Math.pow((1+interestRate),loanTerm)-1);
lblResult.setText(String.valueOf(formatter.format(emi)));
}

catch(Exceptione)
{
Toast.makeText(getBaseContext(),"InvalidInput",Toast.LENGTH_LONG).show();
}
}
}
```

Sample Output



Principal Amount 1000000

Down Payment 20000

Interest Rate 10.5

Loan Term (Months) 36

31852.39

VIVA QUESTIONS

1. What is Android?
It is an open-sourced operating system that is used primarily on mobile devices, such as cell phones and tablets. It is a Linux kernel-based system that's been equipped with rich components that allowsdevelopers to create and run apps that can perform both basic and advanced functions.</p>

2. What Is the Google Android SDK?
The Google Android SDK is a toolset that developers need in order to write apps on Android enableddevices. It contains a graphical interface that emulates an Android driven handheld environment, allowing them to test and debug their codes.

3. What is the Android Architecture?
Android Architecture is made up of 4 key components:

4. Describe the Android Framework.
The Android Framework is an important aspect of the Android Architecture. Here you can find all the classes and methods that developers would need in order to write applications on the Android environment.

5. What is AAPT?
AAPT is short for Android Asset Packaging Tool. This tool provides developers with the ability to deal with zip-compatible archives, which includes creating, extracting as well as viewing its contents.

6. What is the importance of having an emulator within the Android environment?
The emulator lets developers "play" around an interface that acts as if it were an actual mobile device.They can write and test codes, and even debug. Emulators are a safe place for testing codes especiallyif it is in the early design phase.

7. What is the use of an activityCreator?
An activity Creator is the first step towards the creation of a new Android project. It is made up of a shell script that will be used to create new file system structure necessary for writing codes within theAndroid IDE.

8 . Describe Activities.
Activities are what you refer to as the window to a user interface. Just as you create windows in orderto display output or to ask for an input in the form of dialog boxes, activities play the same role, though it may not always be in the form of a user interface.

9. What are Intents?
Intents displays notification messages to the user from within the Android enabled device. It can be used to alert the user of a particular state that occurred. Users can be made to respond to intents.

10. Differentiate Activities from Services.

Activities can be closed, or terminated anytime the user wishes. On the other hand, services are designed to run behind the scenes, and can act independently. Most services run continuously, regardless of whether there are certain or no activities being executed.

11. What items are important in every Android project?

These are the essential items that are present each time an Android project is created:

Android Manifest.xml
build.xml
bin/
src/
res/
assets/

12. What is the importance of XML-based layouts?

The use of XML-based layouts provides a consistent and somewhat standard means of setting GUI definition format. In common practice, layout details are placed in XML files while other items are placed in source files.

13. What are containers?

Containers, as the name itself implies, holds objects and widgets together, depending on which specific items are needed and in what particular arrangement that is wanted. Containers may hold labels, fields, buttons, or even child containers, as examples.

14. What is Orientation?

Orientation, which can be set using set Orientation(), dictates if the Linear Layout is represented asa row or as a column. Values are set as either HORIZONTAL or VERTICAL.

15. What is the importance of Android in the mobile market?

Developers can write and register apps that will specifically run under the Android environment. Thismeans that every mobile device that is Android enabled will be able to support and run these apps. With the growing popularity of Android mobile devices, developers can take advantage of thistrend by creating and uploading their apps on the Android Market for distribution to anyone who wants todownload it.

16. What do you think are some disadvantages of Android?

Given that Android is an open-source platform, and the fact that different Android operatingsystemshave been released on different mobile devices, there's no clear cut policy to how applications can adapt with various OS versions and upgrades.

–One app that runs on this particular version of Android OS may or may not run on another version.

–Another disadvantage is that since mobile devices such as phones and tabs come in different sizes and forms, it poses a challenge for developers to create apps that can adjust correctly to the right screen size and other varying features and specs.

17. What is adb?

Adb is short for "Android Debug Bridge". It allows developers the power to execute remote shell commands. Its basic function is to allow and control communication towards and from the emulator port.

18. What are the four essential states of an activity?
Active – if the activity is at the foreground
Paused – if the activity is at the background and still visible
Stopped – if the activity is not visible and therefore is hidden or obscured by another activityDestroyed – when the activity process is killed or completed terminated

19. What is ANR?
ANR is short for Application Not Responding. This is actually a dialog that appears to the userwhenever an application have been unresponsive for a long period of time.

Which elements can occur only once and must be present?
Among the different elements, the and elements must be present and can occur only once. The rest are optional, and can occur as many times as needed.

20. How are escape characters used as attribute?
Escape characters are preceded by double backslashes. For example, a newline character is created using

21. What is the importance of settings permissions in app development?
Permissions allow certain restrictions to be imposed primarily to protect data and code. Without these,codes could be compromised, resulting to defects in functionality.

22. What is the function of an intent filter?
Because every component needs to indicate which intents they can respond to, intent filters are usedto filter out intents that these components are willing to receive. One or more intent filters are possible,depending on the services and activities that is going to make use of it

23. Enumerate the three key loops when monitoring an activity? Entire lifetime – activity happens between on Create and on Destroy Visible lifetime – activity happens between on Start and on Stop Foreground lifetime – activity happens between on Resume and on Pause

24. When is the on Stop(. method invoked?
A call to on Stop method happens when an activity is no longer visible to the user, either because another activity has taken over or if in front of that activity.

25. Is there a case wherein other qualifiers in multiple resources take precedence over locale? Yes, there are actually instances wherein some qualifiers can take precedence over locale. There aretwo known exceptions, which are the MCC (mobile country code. and MNC (mobile network code.qualifiers.

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 IndiaPvt Ltd,2014. ISBN-13:978-8126547197
3. Dawn Griffiths and David Griffiths, "Head First Android Development", 1st Edition, O‟ReillySPD Publishers, 2015. ISBN-13:978-9352131341
4. BillPhillips,ChrisStewartandKristinMarsicano, "AndroidProgramming:TheBigNerd RanchGuide", 3rd Edition, Big Nerd Ranch Guides, 2017. ISBN-13:978-0134706054