Practical No.1

1. List different android Os version.

Android versions and their names

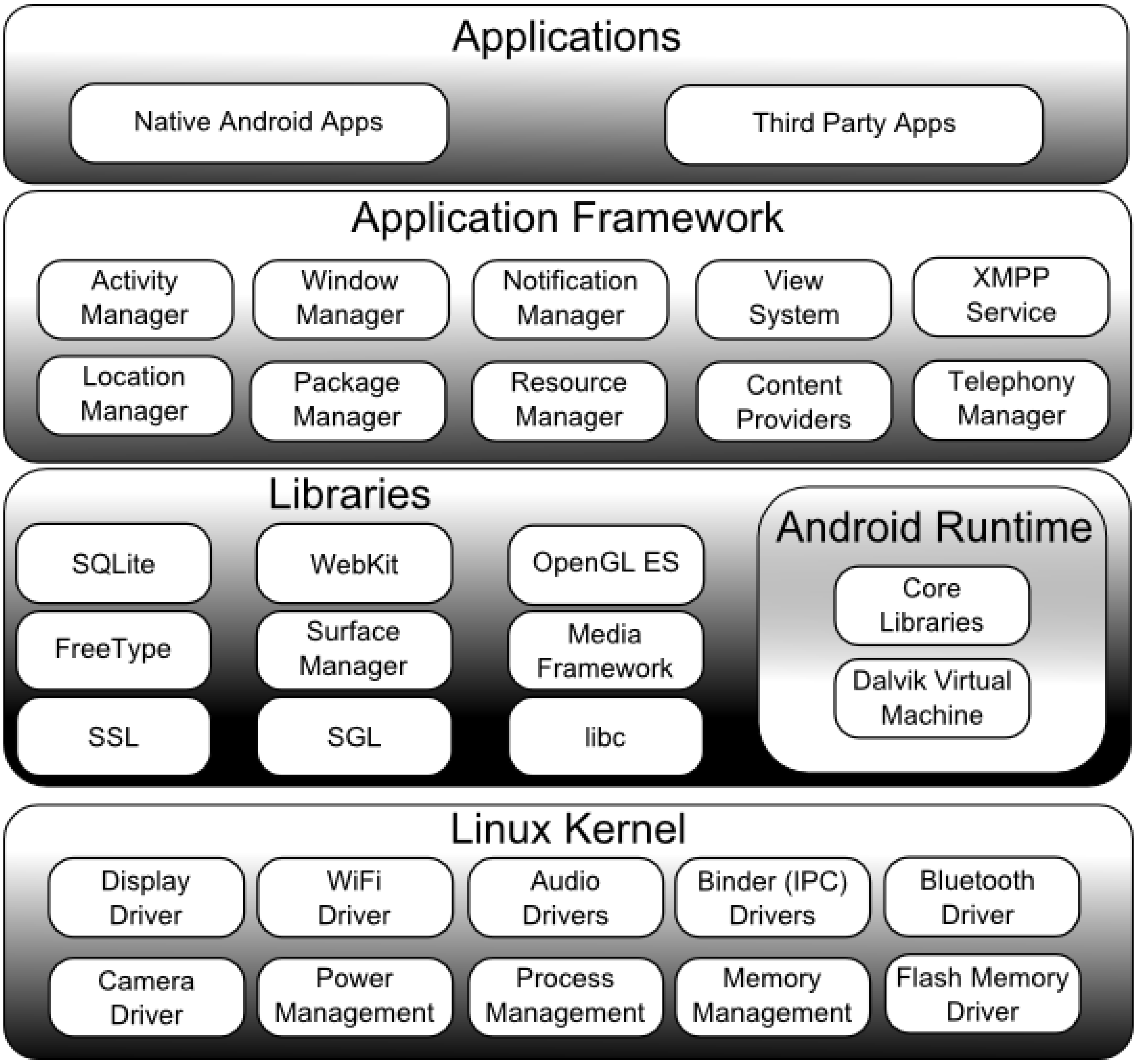
1. Android 1.5: Android Cupcake
2. Android 1.6: Android Donut
3. Android 2.0: Android Eclair
4. Android 2.2: Android Froyo
5. Android 2.3: Android Gingerbread
6. Android 3.0: Android Honeycomb
7. Android 4.0: Android Ice Cream Sandwich
8. Android 4.1 to 4.3.1: Android Jelly Bean
9. Android 4.4 to 4.4.4: Android KitKat
10. Android 5.0 to 5.1.1: Android Lollipop
11. Android 6.0 to 6.0.1: Android Marshmallow
12. Android 7.0 to 7.1: Android Nougat
13. Android 8.0 to Android 8.1: Android Oreo
14. Android 9.0: Android Pie

2. State characteristics of android OS.

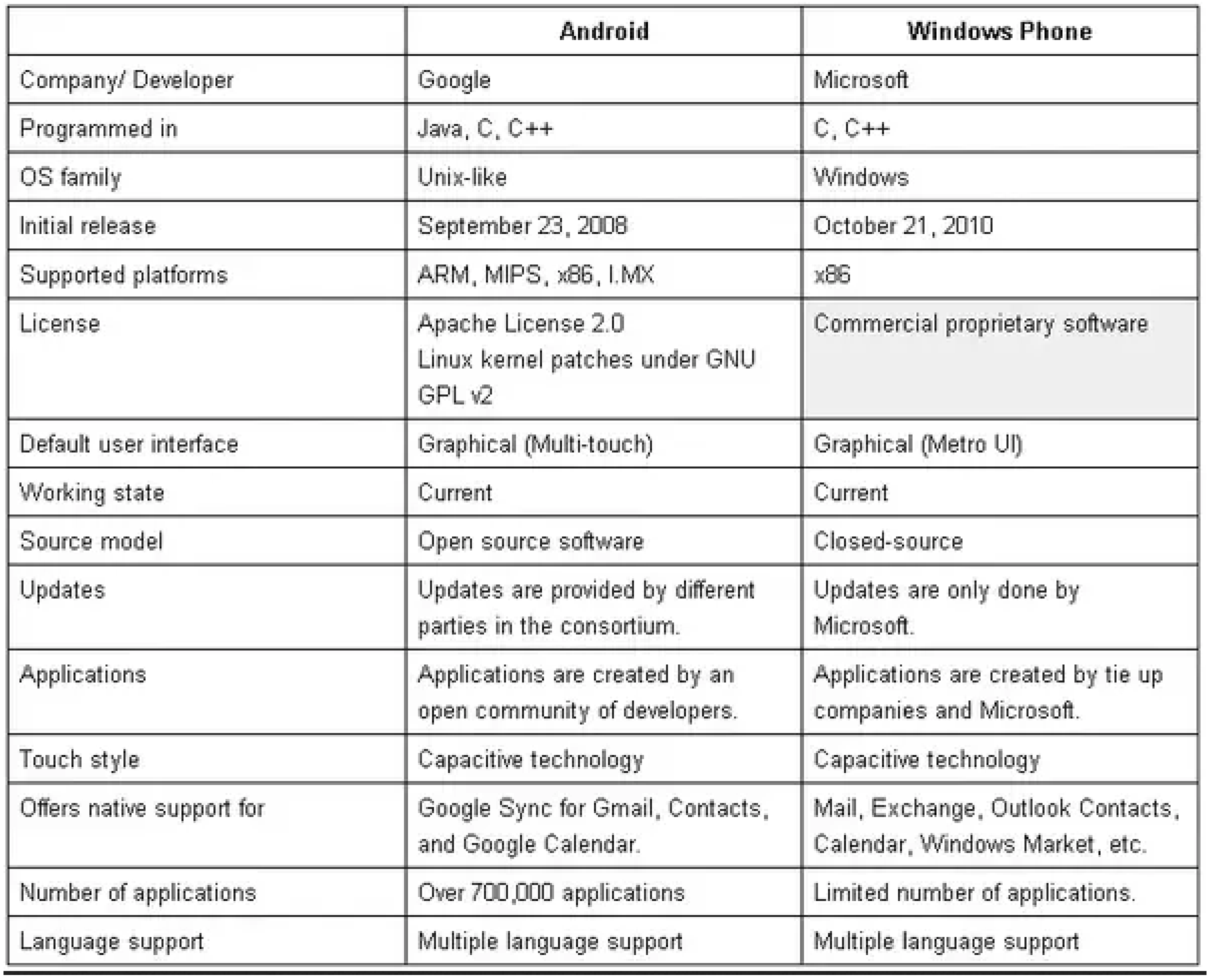
characteristics Of Android Are:

- Head set layout.
- Storage.
- Connectivity: GSM/EDGE, IDEN, CDMA, Bluetooth, WI-FI, EDGE,3G,NFC, LTE,GPS.
- Messaging: SMS, MMS, C2DM (could to device messaging), GCM (Google could messaging)
- Multilanguage support.
- Multi-touch.
- Video calling.
- Screen capture.

3. Draw the architectural diagram of android OS.



4. Difference between windows OS and android OS.



	Android	Windows Phone
Company/ Developer	Google	Microsoft
Programmed in	Java, C, C++	C, C++
OS family	Unix-like	Windows
Initial release	September 23, 2008	October 21, 2010
Supported platforms	ARM, MIPS, x86, I.MX	x86
License	Apache License 2.0 Linux kernel patches under GNU GPL v2	Commercial proprietary software
Default user interface	Graphical (Multi-touch)	Graphical (Metro UI)
Working state	Current	Current
Source model	Open source software	Closed-source
Updates	Updates are provided by different parties in the consortium.	Updates are only done by Microsoft.
Applications	Applications are created by an open community of developers.	Applications are created by tie up companies and Microsoft.
Touch style	Capacitive technology	Capacitive technology
Offers native support for	Google Sync for Gmail, Contacts, and Google Calendar.	Mail, Exchange, Outlook Contacts, Calendar, Windows Market, etc.
Number of applications	Over 700,000 applications	Limited number of applications.
Language support	Multiple language support	Multiple language support

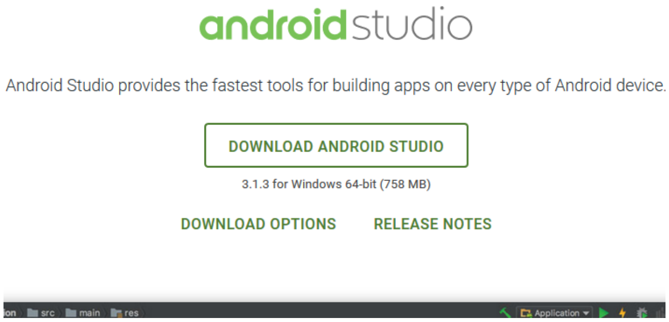
Practical no.2

1.List all the steps to install android OS.

Installation guide:

Step – 1 :
Head over to this link to get the Android Studio executable or zip file .

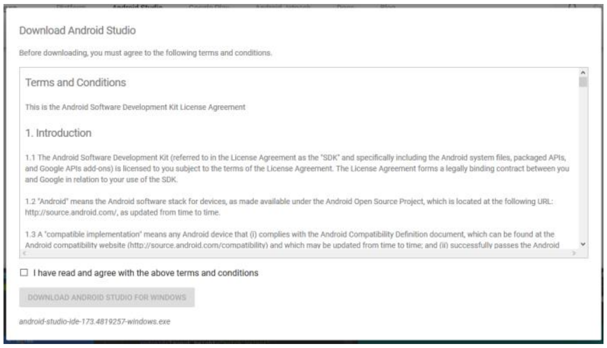
Step – 2 :
Click on the download android studio button .



Android Studio provides the fastest tools for building apps on every type of Android device.

3.1.3 for Windows 64-bit (758 MB)

Click on the “I have read and agree with the above terms and conditions” checkbox followed by the download button.



Download Android Studio

Before downloading, you must agree to the following terms and conditions.

Terms and Conditions

This is the Android Software Development Kit License Agreement

1. Introduction

1.1 The Android Software Development Kit (referred to in the License Agreement as the "SDK" and specifically including the Android system files, packaged APIs, and Google APIs add-ons) is licensed to you subject to the terms of the License Agreement. The License Agreement forms a legally binding contract between you and Google in relation to your use of the SDK.

1.2 "Android" means the Android software stack for devices, as made available under the Android Open Source Project, which is located at the following URL: http://source.android.com/, as updated from time to time.

1.3 A "compatible implementation" means any Android device that (i) complies with the Android Compatibility Definition document, which can be found at the Android compatibility website (http://source.android.com/compatibility) and which may be updated from time to time; and (ii) successfully passes the Android

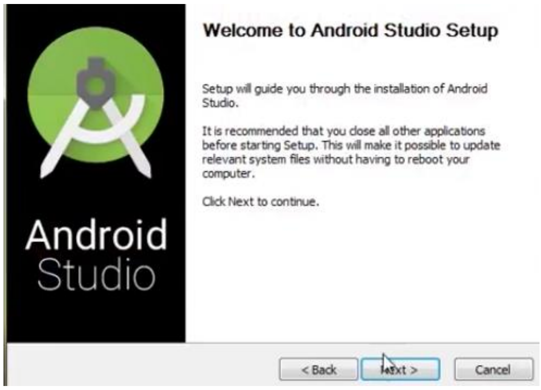
☐ I have read and agree with the above terms and conditions

android-studio-ide-173.4819257-windows.exe

Click on Save file button in the appeared prompt box and the file will start downloading .

Step – 3:

After the downloading has finished, open the file from downloads and run it .
It will prompt the following dialogue box .



Welcome to Android Studio Setup

Setup will guide you through the installation of Android Studio.

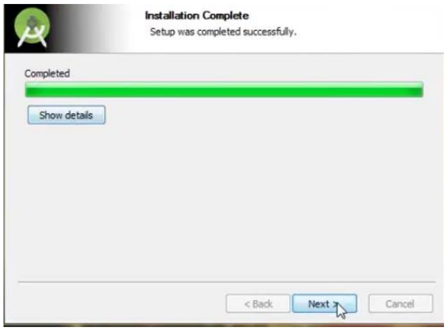
It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Click on next .

In the next prompt it'll ask for a path for installation. Choose a path and hit next.Note :The installation path should have the required minimum space.

Step – 4 : It will start the installation, and once it is completed, it will be like the image shown below .

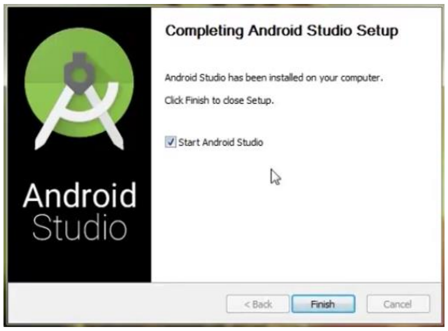


Installation Complete

Setup was completed successfully.

Completed

Click on next .



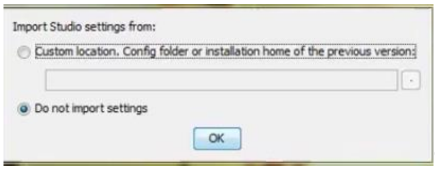
Completing Android Studio Setup

Android Studio has been installed on your computer.

Click Finish to close Setup.

☑ Start Android Studio

Step – 5 :Once "Finish" is clicked, it will ask whether the previous settings needs to be impoerted [if android studio had been installed earlier], or not.
It is better to choose the 'Don't import Settings option' .



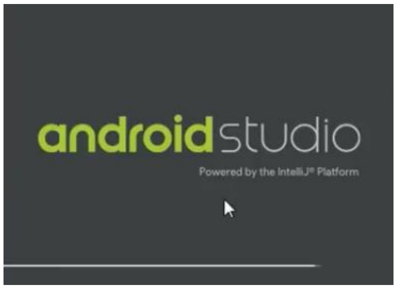
Import Studio settings from:

○ Custom location. Config folder or installation home of the previous version:

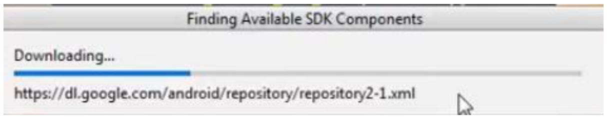
● Do not import settings

Click the OK button.

Step – 6 : This will start the Android Studio.



Meanwhile it will be finding the available SDK components .

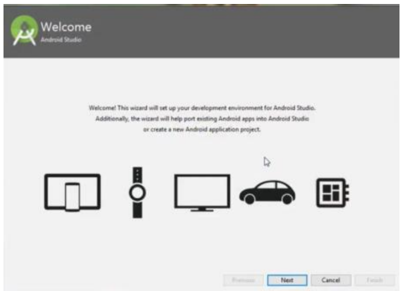


Finding Available SDK Components

Downloading...

https://dl.google.com/android/repository/repository2-1.xml

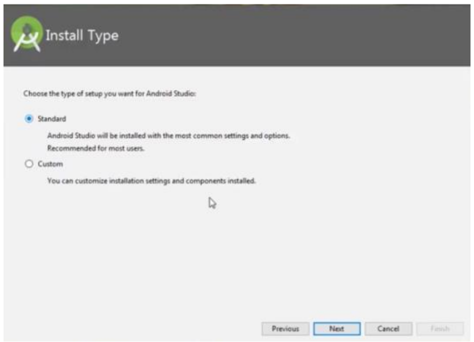
Step – 7: After it has found the SDK components, it will redirect to the Welcome dialog box .



Welcome
Android Studio

Welcome! This wizard will set up your development environment for Android Studio.
Additionally, the wizard will help port existing Android apps into Android Studio
or create a new Android application project.

Click on next .



Install Type

Choose the type of setup you want for Android Studio:

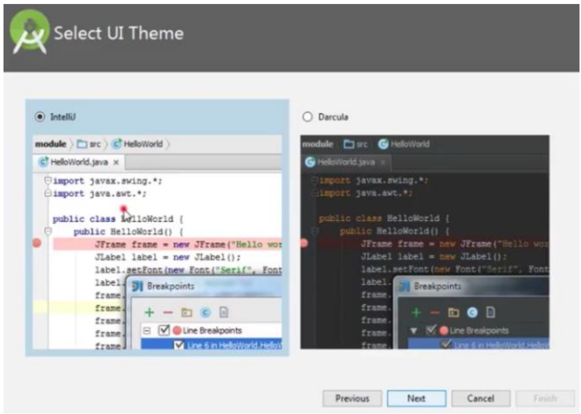
● Standard

Android Studio will be installed with the most common settings and options.
Recommended for most users.

○ Custom

You can customize installation settings and components installed.

Choose Standard and click on Next.
Now choose the theme, whether Light theme or the Dark one .
The light one is called the IntelliJ theme whereas the dark theme is called Darcula .
Choose as required.

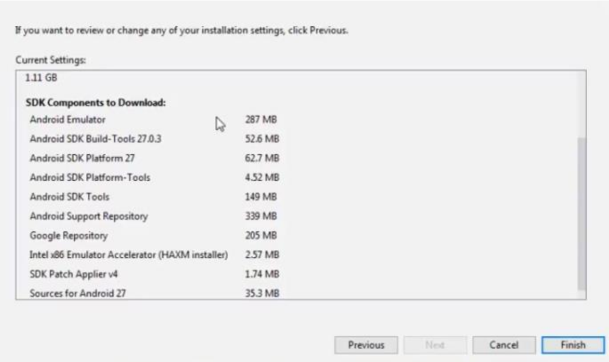


● IntelliJ ○ Darcula

```
import javax.swing.*;
import java.awt.*;

public class HelloWorld {
    public HelloWorld() {
        JFrame frame = new JFrame("Hello wor
        JLabel label = new JLabel();
        label.setFont(new Font("Serif", Font
        label.
        frame.
        frame.
        frame.
        frame.
        frame.
```

Click on the Next button

Step – 8 : Now it is time to download the SDK components .



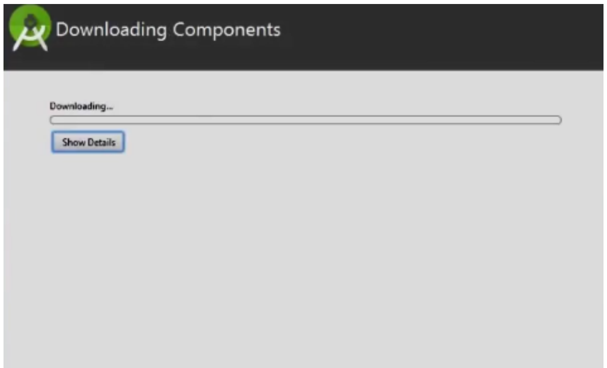
If you want to review or change any of your installation settings, click Previous.

Current Settings:
1.11 GB

SDK Components to Download:

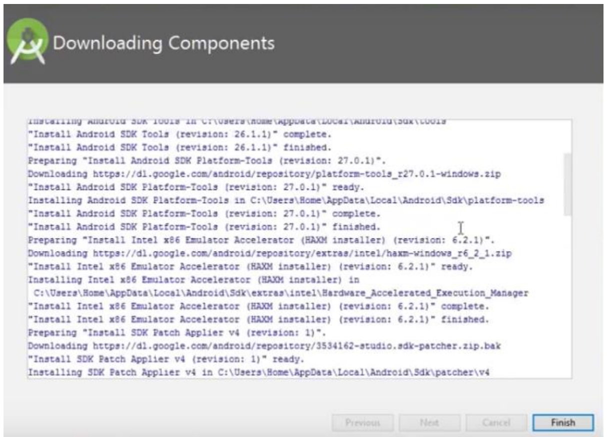
Android Emulator	287 MB
Android SDK Build-Tools 27.0.3	52.6 MB
Android SDK Platform 27	62.7 MB
Android SDK Platform-Tools	4.52 MB
Android SDK Tools	149 MB
Android Support Repository	339 MB
Google Repository	205 MB
Intel x86 Emulator Accelerator (HAXM installer)	2.57 MB
SDK Patch Applier v4	1.74 MB
Sources for Android 27	35.3 MB

Click on Finish .



Downloading...

It has started downloading the components



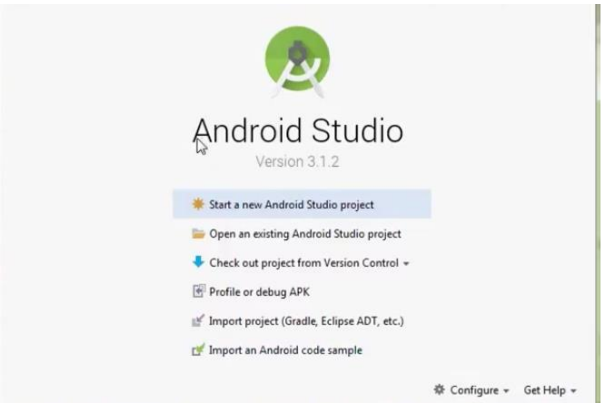
```
Installing Android SDK Tools in C:\Users\Home\AppData\Local\Android\Sdk\tools
"Install Android SDK Tools (revision: 26.1.1)" complete.
"Install Android SDK Tools (revision: 26.1.1)" finished.
Preparing "Install Android SDK Platform-Tools (revision: 27.0.1)".
Downloading https://dl.google.com/android/repository/platform-tools_r27.0.1-windows.zip
"Install Android SDK Platform-Tools (revision: 27.0.1)" ready.
Installing Android SDK Platform-Tools in C:\Users\Home\AppData\Local\Android\Sdk\platform-tools
"Install Android SDK Platform-Tools (revision: 27.0.1)" complete.
"Install Android SDK Platform-Tools (revision: 27.0.1)" finished.
Preparing "Install Intel x86 Emulator Accelerator (HAXM installer) (revision: 6.2.1)".
Downloading https://dl.google.com/android/repository/extras/intel/haxm-windows_r6_2_1.zip
"Install Intel x86 Emulator Accelerator (HAXM installer) (revision: 6.2.1)" ready.
Installing Intel x86 Emulator Accelerator (HAXM installer) in
 C:\Users\Home\AppData\Local\Android\Sdk\extras\intel\Hardware_Accelerated_Execution_Manager
"Install Intel x86 Emulator Accelerator (HAXM installer) (revision: 6.2.1)" complete.
"Install Intel x86 Emulator Accelerator (HAXM installer) (revision: 6.2.1)" finished.
Preparing "Install SDK Patch Applier v4 (revision: 1)".
Downloading https://dl.google.com/android/repository/3534162-studio.sdk-patcher.zip.bak
"Install SDK Patch Applier v4 (revision: 1)" ready.
Installing SDK Patch Applier v4 in C:\Users\Home\AppData\Local\Android\Sdk\patcher\v4
```

The Android Studio has been successfully configured.
Now it’s time to launch and build apps.

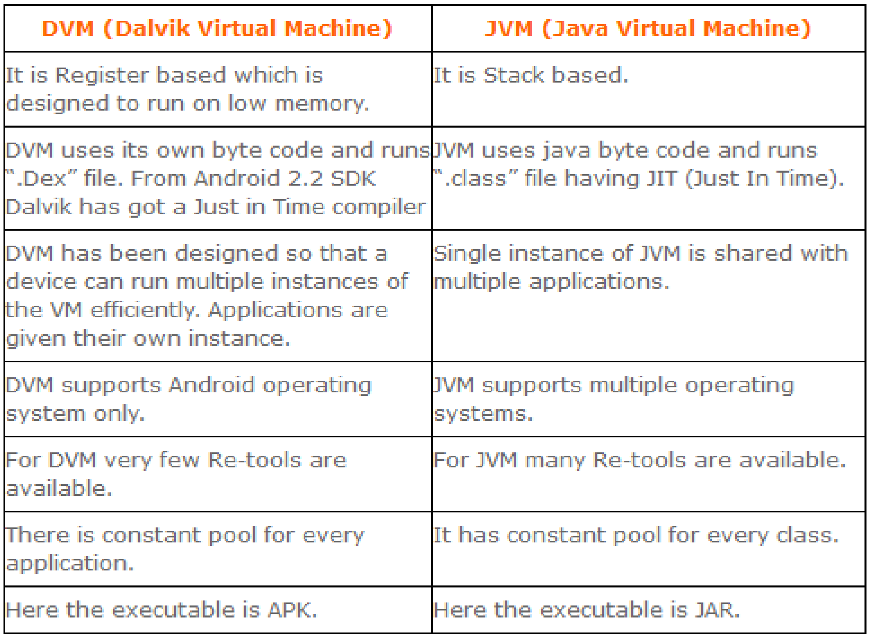
Click on the Finish button to launch it.

- Step – 9 :

  Click on ‘Start new android project’ to build a new app.


Android Studio
Version 3.1.2
Start a new Android Studio project
Open an existing Android Studio project
Check out project from Version Control
Profile or debug APK
Import project (Gradle, Eclipse ADT, etc.)
Import an Android code sample
Configure
Get Help

2.List various IDEs that can be used to execute android OS.
Ans:1)Android studio.2)Eclipse.
3.Difference between DVM and JVM.



DVM (Dalvik Virtual Machine)	JVM (Java Virtual Machine)
It is Register based which is designed to run on low memory.	It is Stack based.
DVM uses its own byte code and runs ".Dex" file. From Android 2.2 SDK Dalvik has got a Just in Time compiler	JVM uses java byte code and runs ".class" file having JIT (Just In Time).
DVM has been designed so that a device can run multiple instances of the VM efficiently. Applications are given their own instance.	Single instance of JVM is shared with multiple applications.
DVM supports Android operating system only.	JVM supports multiple operating systems.
For DVM very few Re-tools are available.	For JVM many Re-tools are available.
There is constant pool for every application.	It has constant pool for every class.
Here the executable is APK.	Here the executable is JAR.

4. What is IDE? Why java development tool kit is essential to install an android OS.

Ans:-An IDE, or Integrated Development Environment, enables programmers to consolidate the different aspects of writing a computer program. IDEs increase programmer productivity by combining common activities of writing software into a single application: editing source code, building executable, and debugging.

If your system has an up-to-date JDK installed, you won't need to install it again. The JDK provides tools, such as the Java compiler, used by IDEs and SDKs for developing Java programs. The JDK also contains a Java Runtime Environment (JRE), which enables Java programs, such as Eclipse, to run on your system.

Practical No.3

1. List basic requirements for configuring android OS.

The absolute minimum requirements for Android were originally a 200 MHz processor, 32 MB of RAM, and 32 MB of storage. Out of the box, Android is incompatible with ARMv4 or lower; ARMv5 or higher is needed to run native code without modifications. Android 4.4+ requires an ARMv7 processor.

2. Why bytecode cannot run on android.

We cannot run Java Bytecode on Android because: Android uses Dalvik VM(virtual machine) instead of Java VM. ... Android has been modified to run on smaller devices with the exhaustion of less computing power. In Android, we have to novitiate Java class file into Dalvik executable files using an android tool called dx.

3. What is Build Type in gradle.

A build variant is a cross product of a build type and product flavor, and is the configuration Gradle uses to build your app. Using build variants, you can build the debug version of your product flavors during development, or signed release versions of your product flavors for distribution.

4. Explain the build process in android.

The build process

1. The compilers convert your source code into DEX (Dalvik Executable) files, which include the bytecode that runs on Android devices, and everything else into compiled resources.
2. The APK Packager combines the DEX files and compiled resources into a single APK.

Practical No-4

1. List the file used to helloworld program.

Ans:-Java-This contains the .java source files for your project. By default, it includes an MainActivity.java source file having an activity class that runs when your app is launched using the app icon.

res/drawable-hdpi-This is a directory for drawable objects that are designed for high-density screens.

res/layout-This is a directory for files that define your app's user interface.

res/values-This is a directory for other various XML files that contain a collection of resources, such as strings and colours definitions.

AndroidManifest.xml-This is the manifest file which describes the fundamental characteristics of the app and defines each of its components.

Build.gradle-This is an auto generated file which contains compileSdkVersion, buildToolsVersion, applicationId, minSdkVersion, targetSdkVersion, versionCode and versionName

2. What is an activity in an android programming?

An activity represents a single screen with a user interface just like window or frame of Java. Android activity is the subclass of ContextThemeWrapper class.

3. WAP to display HelloWorld.

MainActivity.java:

```
package com.example.helloworld;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_main);
   }
}
```

The Manifest File

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.tutorialspoint7.myapplication">

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

activity_main.xml

```
<RelativeLayout
 xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent" >

  <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="hello_world"
      tools:context=".MainActivity" />

</RelativeLayout>
```

4.WAP to display student name and marks.

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

    <TableLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
<TableRow
            android:layout_width="fill_parent"
            android:layout_height=" fill_parent ">
<TextView
               android:layout_width="100dp"
                android:layout_height="wrap_content"
                android:text="Name"/>
  <TextView
                android:layout_width="100dp"
                android:layout_height="wrap_content"
                android:text="Marks"/>
 </TableRow>
<TableRow
             android:layout_width="fill_parent"
            android:layout_height=" fill_parent "
            android:text="Abc"/>
 <TextView
               android:layout_width="100dp"
                android:layout_height="wrap_content"
                android:text="90"/>
        </TableRow>


   </TableLayout>
```

Practical No-5

1. Name any three layout manager.
    - Linear Layout
    - Relative Layout
    - Absolute Layout
    - Frame Layout
    - Table Layout

2. What is card view?

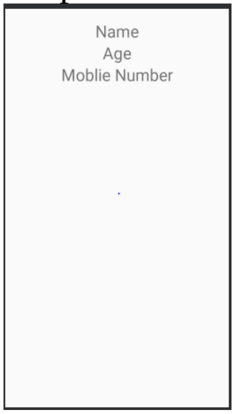
    As with material design a new view was introduced through the support v7 library, called Card View. It can be used in many ways to display cards in android. For example, it can be used like an independent view to show floating search bar on top of the screen also it can be used to display cards in a list.

1. Write a program to place Name, Age and mobile number linearly on the display screen using linear layout.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".MainActivity"
   android:orientation="vertical">
 <TextView
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:text="Name"
      android:textSize="30dp"
      android:textAlignment="center"
      android:layout_marginTop="20dp"/>
   <TextView
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:text="Age"
      android:textSize="30dp"
      android:textAlignment="center"
      android:layout_marginTop="20dp" />
```

```
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Moblie Number"
        android:textSize="30dp"
        android:textAlignment="center"
        android:layout_marginTop="20dp"/>
</LinearLayout>
```

Output:-

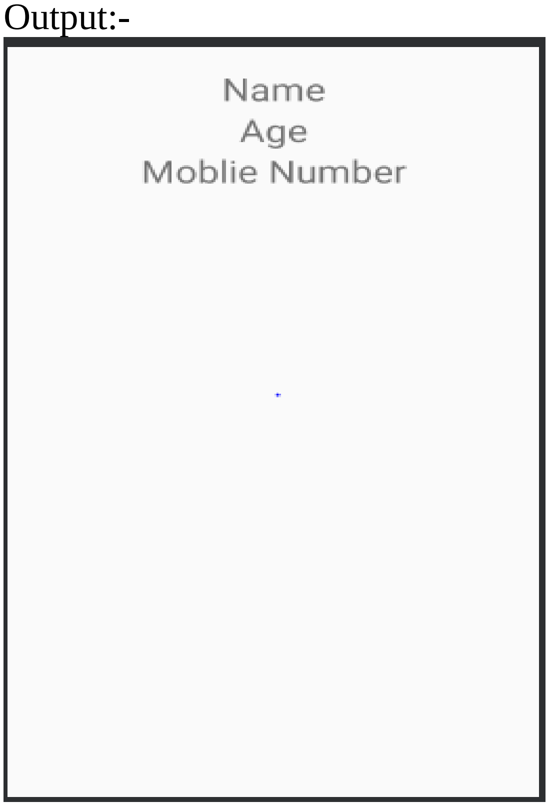


Name
Age
Moblie Number

2. Write a program to place Name, Age and mobile number linearly on the display screen using linear layout.

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity" >
<TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Name"
        android:textSize="30dp"
        android:textAlignment="center"
       android:paddingTop="20dp"/>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Age"
        android:textSize="30dp"
        android:textAlignment="center"
        android:paddingTop="60dp"/>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Moblie Number"
        android:textSize="30dp"
        android:textAlignment="center"
        android:paddingTop="100dp" />
</AbsoluteLayout>
```

Output:-



Name
Age
Moblie Number

Practical No- 06

Theory question

1. List different attribute which can be used with any layout managers.

- android:id
  - This is the ID which uniquely identifies the view.
- android:layout_width
  - This is the width of the layout.
- android:layout_height
  - This is the height of the layout
- android:layout_marginTop
  - This is the extra space on the top side of the layout.
- android:layout_marginBottom
  - This is the extra space on the bottom side of the layout.
- android:layout_marginLeft
  - This is the extra space on the left side of the layout.
- android:layout_marginRight
  - This is the extra space on the right side of the layout.
- android:layout_gravity
  - This specifies how child Views are positioned.
- android:layout_x
  - This specifies the x-coordinate of the layout.
- android:layout_y
  - This specifies the y-coordinate of the layout.
- android:layout_width
  - This is the width of the layout.

2. What is Grid Layout?

Ans-

In android GridView is a view group that display items in two dimensional scrolling grid (rows and columns), the grid items are not necessarily predetermined but they are automatically inserted to the layout using a ListAdapter. Users can then select any grid item by clicking on it. Android.widget.GridLayout. A layout that places its children in a rectangular grid. The grid is composed of a set of infinitely thin lines that separate the viewing area into cells. Throughout the API, grid lines are referenced by grid indices.

Programs

1. WAP to display 10 student basic information in table form using table layout.

XML file

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout 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">
<TableRow
      android:layout_width="fill_parent"
      android:layout_height="fill_parent">
<TextView
      android:layout_width="100dp"
      android:layout_height="wrap_content"
      android:text="Roll no"/>
<TextView
         android:layout_width="100dp"
         android:layout_height="wrap_content"
         android:text="Name"/>
<TextView
         android:layout_width="100dp"
         android:layout_height="wrap_content"
         android:text=" Mobile"/>
 <TextView
         android:layout_width="100dp"
         android:layout_height="wrap_content"
         android:text=" Address"/>
   </TableRow>
<TableRow android:layout_height="fill_parent"
   android:layout_width="fill_parent">
      <TextView
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="1"/>
<TextView
         android:layout_width="100dp"
         android:layout_height="wrap_content"
         android:text="aaa"/>
<TextView
         android:layout_width="100dp"
```

```
            android:layout_height="wrap_content"
            android:text=" 3579034679"/>
 <TextView
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:text=" xyz"/>
    </TableRow>
    <TableRow android:layout_height="fill_parent"
        android:layout_width="fill_parent">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="2"/>
<TextView
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:text="bbb"/>
<TextView
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:text=" 3579037870"/>
<TextView
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:text=" aaa"/>
       </TableRow>
<TableRow android:layout_height="fill_parent"
        android:layout_width="fill_parent">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="3" />
    <TextView
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:text="ccc"/>
<TextView
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:text=" 3574534679"/>
 <TextView
         android:layout_width="100dp"
         android:layout_height="wrap_content"
         android:text=" bbb"/>
   </TableRow>
   <TableRow android:layout_height="fill_parent"
      android:layout_width="fill_parent">
      <TextView
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="4"/>
<TextView
         android:layout_width="100dp"
         android:layout_height="wrap_content"
         android:text="ddd"/>
<TextView
         android:layout_width="100dp"
         android:layout_height="wrap_content"
         android:text=" 3898834679"/>
<TextView
         android:layout_width="100dp"
         android:layout_height="wrap_content"
         android:text=" ccc"/>
   </TableRow>
   <TableRow android:layout_height="fill_parent"
      android:layout_width="fill_parent">
      <TextView
         android:layout_width="wrap_content"
android:layout_height="wrap_content"
         android:text="5"/>
<TextView
         android:layout_width="100dp"
         android:layout_height="wrap_content"
         android:text="eee"/>
 <TextView
         android:layout_width="100dp"
         android:layout_height="wrap_content"
         android:text=" 5555034679"/>

      <TextView
```

```
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:text=" ddd"/>
    </TableRow>
    <TableRow android:layout_height="fill_parent"
        android:layout_width="fill_parent">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="6"/>
<TextView
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:text="fff"/>
<TextView
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:text=" 3572222679"/>
 <TextView
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:text=" eee"/>
    </TableRow>
    <TableRow android:layout_height="fill_parent"
        android:layout_width="fill_parent">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="7"/>
 <TextView
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:text="ggg"/>
  <TextView
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:text=" 3579030079"/>
<TextView
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:text=" fff"/>
    </TableRow>
    <TableRow android:layout_height="fill_parent"
        android:layout_width="fill_parent">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="8"/>
  <TextView
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:text="hhhh"/>
 <TextView
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:text=" 3573456679"/>
 <TextView
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:text="ggg"/>
    </TableRow>
    <TableRow android:layout_height="fill_parent"
        android:layout_width="fill_parent">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="9"/>
<TextView
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:text="iii"/>
<TextView
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:text=" 3579345679"/>
 <TextView
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:text=" jjj"/>
    </TableRow>
```

```
<TableRow
            android:layout_height="fill_parent"
            android:layout_width="fill_parent">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="10"/>
<TextView
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:text="kkk"/>
<TextView
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:text=" 6789034679"/>
 <TextView
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:text=" sss"/>
</TableRow>
    </TableLayout>
```

2.WAP to display all the data type in object oriented programming using FrameLayout.

```
      <?xml version="1.0" encoding="utf-8"?>
      <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
          xmlns:app="http://schemas.android.com/apk/res-auto"
          xmlns:tools="http://schemas.android.com/tools"
          android:layout_width="match_parent"
          android:layout_height="match_parent"
          tools:context=".MainActivity">
      <TextView
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
             android:textSize="30dp"
              android:text="integer"
              android:layout_marginTop="20dp"/>
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="float"
    android:textSize="30dp"
    android:layout_marginTop="20dp"
    android:layout_marginLeft="150dp"/>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="string"
    android:textSize="30dp"
    android:layout_marginTop="20dp"
    android:layout_marginLeft="300dp"/>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="150dp"
    android:text="double"
    android:textSize="30dp”/>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="long"
    android:textSize="30dp"
    android:layout_marginTop="150dp"
    android:layout_marginLeft="150dp"/>
  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="char"
    android:textSize="30dp"
    android:layout_marginTop="150dp"
    android:layout_marginLeft="300dp"/>
</FrameLayout>
```

Practical No-07

1. Which of these is not defined as process state.

    1) Non-visible
    2) Visible
    3) Forground
    4) Background

2. What is the name of the folder that contains the R.java file?

    1) Src
    2) Res
    3) Bin
    4) Gen

Programs:

2. WAP to accept and display personal information of the student.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">
<TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Sign Up"
        android:layout_marginLeft="150dp"
        android:layout_marginTop="15dp"
        android:textSize="35dp"/>
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">
   <TextView
            android:id="@+id/fname"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="First Name"
            android:textSize="20dp"
```

```
       android:layout_marginTop="50dp"
       android:layout_marginLeft="40dp" />
    <TextView
       android:id="@+id/lname"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="Last Name"
       android:textSize="20dp"
       android:layout_marginTop="110dp"
       android:layout_marginLeft="40dp”/>
    <TextView
       android:id="@+id/mono"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="Mobile No"
       android:textSize="20dp"
       android:layout_marginTop="170dp"
       android:layout_marginLeft="40dp”/>
<TextView
       android:id="@+id/email"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="Email ID"
       android:textSize="20dp"
       android:layout_marginTop="230dp"
       android:layout_marginLeft="40dp”/>
    <TextView
       android:id="@+id/bdate"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="Birth Date"
       android:textSize="20dp"
       android:layout_marginTop="290dp"
       android:layout_marginLeft="40dp"/>
    <EditText
       android:id="@+id/fn"
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:layout_marginTop="30dp"
       android:layout_marginLeft="170dp"/>
```

```
        <EditText
            android:id="@+id/ln"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="90dp"
            android:layout_marginLeft="170dp"/>
<EditText
            android:id="@+id/mo"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="150dp"
            android:layout_marginLeft="170dp” />
        <EditText
            android:id="@+id/em"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="210dp"
            android:layout_marginLeft="170dp"/>
        <EditText
            android:id="@+id/bd"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="270dp"
            android:layout_marginLeft="170dp"/>
        <Button
            android:id="@+id/btn"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Submit"
            android:textSize="25dp"
            android:layout_marginTop="380dp"
            android:layout_marginLeft="140dp"/>
    </RelativeLayout>
</LinearLayout>
```

Java code

```
package com.example.student;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity
{
    EditText fanem,lname,email,mono,bdate;
    Button b;
@Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        fanem=findViewById(R.id.fn);
        lname=findViewById(R.id.ln);
        email=findViewById(R.id.em);
        mono=findViewById(R.id.mo);
        bdate=findViewById(R.id.bd);
        b=findViewById(R.id.btn);

        b.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                Toast.makeText(MainActivity.this,"successfully
inserted"+fanem,Toast.LENGTH_SHORT).show();
            }
        });

    }
}
```

Practical No-8

Theory:

1. What does android: completionHint attribute in Auto Complete Text View does?

   Ans:

   AutocompleteTextView is an editable text view that shows completion suggestions automatically while the user is typing. The list of suggestions is displayed in a drop down menu from which the user can choose an item to replace the content of the edit box with.

   android: completionHint

   This defines the hint displayed in the drop down menu.

2. How to create AutoCompleteTextView field in XML?

Ans:

```
<AutoCompleteTextView
    android:id="@+id/autoCompleteTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="65dp"
    android:ems="10" >
```

Program:

Write a program to create a first display screen of any search engine using Auto complete text view.

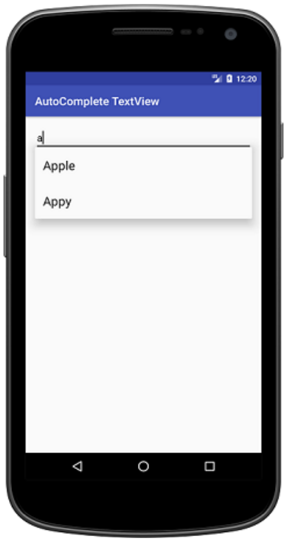
Java File

```
package com.example.autocompletetextview;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.AutoCompleteTextView;

public class MainActivity extends AppCompatActivity {
 String[] course ={"co","if", "Me","ej","MMM"};
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ArrayAdapter<String> ad= new ArrayAdapter<>(this, android.R.layout.select_dialog_item, course);

        AutoCompleteTextView ac=(AutoCompleteTextView)findViewById(R.id.auto);
        ac.setThreshold(1);
        ac.setAdapter(ad);
    }
}
```

Xml File

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

    <AutoCompleteTextView
        android:id="@+id/auto"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/auto1"
        android:textSize="20sp"
        tools:ignore="LabelFor"
        android:ems="10"
        tools:visibility="visible">
    </AutoCompleteTextView>
</RelativeLayout>
```

- Output:



1. Write a program to display all the subjects of sixth semester using Auto complete text view.

Ans:

MainActivity.java

```
package com.example.demoproject;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.AppCompatAutoCompleteTextView;

import android.widget.ArrayAdapter;public class MainActivity extends
AppCompatActivity
{
private String[] subject = {“MAD”, “MAN”, “PWP”, “NIS”, “CPE”, “EDE”,
“ETI”};
 private AppCompatAutoCompleteTextView autoTextView;
@Override
 protected void onCreate(Bundle savedInstanceState)
 {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_main);

 autoTextView = findViewById(R.id.autoTextView);
ArrayAdapter<String> adapter = new
ArrayAdapter<String>(this,android.R.layout.select_dialog_item, subject);

 autoTextView.setThreshold(1);
 autoTextView.setAdapter(adapter);
 }
}
```

activity_main.xml

```
<?xml version=”1.0" encoding=”utf-8"?>
<LinearLayout xmlns:android=”http://schemas.android.com/apk/res/android"
 android:layout_width=”match_parent”
 android:layout_height=”match_parent”
 android:layout_margin=”16dp”
android:orientation=”vertical”>

<android.support.v7.widget.AppCompatAutoCompleteTextView
 android:id=”@+id/autoTextView”
 android:layout_width=”match_parent”
 android:layout_height=”wrap_content”
 android:hint=”Enter fruit name”
 android:textColor=”#000000"
 android:textColorHint=”#000000" />
</LinearLayout>
```

Practical No.9

Theory :-

1. Write a piece of code to set id of the button

Ans :-

```
Button button = new Button (getApplicationContent());
button.setText("Change");
button.setId(1);
```

2.How to add image to resource file ?

Ans :-

i. Open your project in Android Studio.

ii. Click on res.

iii. Right click on drawable.

iv. Click on Show in Explorer.

v. Double click on drawable folder.

vi. Copy your image file in it and rename as your wish.

vii. Now write your image file name after @drawable/ .

3.List four Android Toggle Button control attribute.

Ans :-

i. android: textOn

ii. android: textOff

iv. android: textColor

v. android:checked="true"

vi. android:text

vii. android:id

viii. android:textSize

Program code

1) Write a program to create a toggle button to display ON / OFF Bluetooth on the display screen.

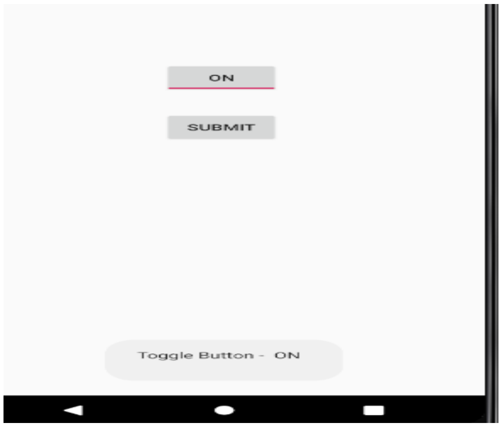
XML code

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

    <ToggleButton
        android:id="@+id/toggle1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="150dp"
        android:layout_marginTop="120dp"
        android:checked="true"
        android:textOff="OFF"
        android:textOn="ON"/>

    <Button
        android:id="@+id/getBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="150dp"
        android:layout_marginTop="200dp"
        android:text="Submit" />
</RelativeLayout>
```

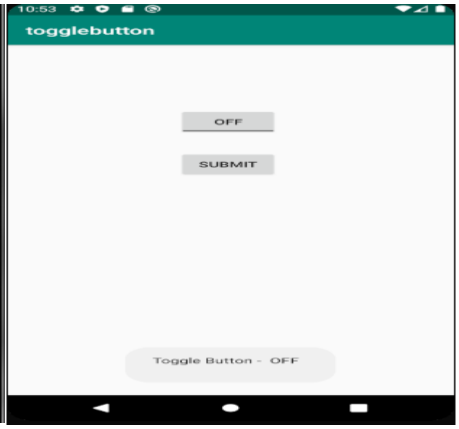
MainActivity (Java file) :-

```
package com.example.togglebutton;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import android.widget.ToggleButton;
public class MainActivity extends AppCompatActivity {

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

        final ToggleButton tb1 = (ToggleButton)findViewById(R.id.toggle1);
       // final ToggleButton tb2 = (ToggleButton)findViewById(R.id.toggle2);m
        Button btnGet = (Button)findViewById(R.id.getBtn);
        btnGet.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Toast.makeText(getApplicationContext(), "Toggle Button -  " +
tb1.getText().toString() + " \n" ,Toast.LENGTH_SHORT).show();
            }
        });
    }
    }
```

Output : -



Toggle Button - ON



Toggle Button - OFF

2.Write a program to create a simple calculator.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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="com.example.a84.calculator.MainActivity">

<RelativeLayout
android:layout_width="368dp"
android:layout_height="495dp"
android:layout_marginBottom="8dp"
android:layout_marginEnd="8dp"
android:layout_marginTop="8dp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toTopOf="parent">

<Button
android:id="@+id/btn_1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
android:layout_below="@+id/edText1"
```

```
android:layout_marginTop="60dp"
android:onClick="PressOne"
android:text="1"
android:textSize="18sp"/>

<Button
android:id="@+id/btn_0"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@+id/btn_8"
android:layout_toEndOf="@+id/btn_7"
android:layout_toRightOf="@+id/btn_7"
android:text="0"
android:textSize="18sp" />

<Button
android:id="@+id/btn_9"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@+id/btn_6"
android:layout_toEndOf="@+id/btn_5"
android:layout_toRightOf="@+id/btn_5"
android:text="9"
android:textSize="18sp" />

<Button
android:id="@+id/btn_8"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
```

```
android:layout_below="@+id/btn_5"
android:layout_toEndOf="@+id/btn_7"
android:layout_toRightOf="@+id/btn_7"
android:text="8"
android:textSize="18sp" />

<Button
android:id="@+id/btn_7"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/btn_4"
android:layout_alignStart="@+id/btn_4"
android:layout_below="@+id/btn_4"
android:text="7"
android:textSize="18sp" />

<Button
android:id="@+id/btn_6"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/btn_5"
android:layout_alignBottom="@+id/btn_5"
android:layout_toEndOf="@+id/btn_5"
android:layout_toRightOf="@+id/btn_5"
android:text="6"
android:textSize="18sp" />

<Button
android:id="@+id/btn_5"
```

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@+id/btn_2"
android:layout_toEndOf="@+id/btn_4"
android:layout_toRightOf="@+id/btn_4"
android:text="5"
android:textSize="18sp" />

<Button
android:id="@+id/btn_4"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/btn_1"
android:layout_alignStart="@+id/btn_1"
android:layout_below="@+id/btn_1"
android:text="4"
android:textSize="18sp" />

<Button
android:id="@+id/btn_3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/btn_2"
android:layout_alignBottom="@+id/btn_2"
android:layout_toEndOf="@+id/btn_2"
android:layout_toRightOf="@+id/btn_2"
android:text="3"
android:textSize="18sp" />
```

```
<Button
android:id="@+id/btn_2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/btn_1"
android:layout_alignBottom="@+id/btn_1"
android:layout_toEndOf="@+id/btn_1"
android:layout_toRightOf="@+id/btn_1"
android:text="2"
android:textSize="18sp" />

<Button
android:id="@+id/btn_Add"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_above="@+id/btn_6"
android:layout_alignParentEnd="true"
android:layout_alignParentRight="true"
android:backgroundTint="@android:color/darker_gray"
android:text="+"
android:textColor="@android:color/background_light"
android:textSize="18sp" />

<Button
android:id="@+id/btn_Sub"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/btn_Add"
```

```
android:layout_alignStart="@+id/btn_Add"
android:layout_below="@+id/btn_Add"
android:backgroundTint="@android:color/darker_gray"
android:text="-"
android:textColor="@android:color/background_light"
android:textSize="18sp" />

<Button
android:id="@+id/btn_Mul"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/btn_Sub"
android:layout_alignStart="@+id/btn_Sub"
android:layout_below="@+id/btn_6"
android:backgroundTint="@android:color/darker_gray"
android:text="*"
android:textColor="@android:color/background_light"
android:textSize="18sp" />

<Button
android:id="@+id/btn_Div"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/btn_Mul"
android:layout_alignStart="@+id/btn_Mul"
android:layout_below="@+id/btn_9"
android:backgroundTint="@android:color/darker_gray"
android:text="/"
```

```
android:textColor="@android:color/background_light"
android:textSize="18sp" />

<EditText
android:id="@+id/edText1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentEnd="true"
android:layout_alignParentLeft="true"
android:layout_alignParentRight="true"
android:layout_alignParentStart="true"
android:layout_alignParentTop="true"
android:layout_marginTop="22dp"
android:ems="10"
android:inputType="textPersonName"
android:textAlignment="textEnd"
android:textSize="24sp" />

<Button
android:id="@+id/btn_calc"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@+id/btn_0"
android:layout_toEndOf="@+id/btn_0"
android:layout_toRightOf="@+id/btn_0"
android:backgroundTint="@android:color/holo_green_light"
android:text="="
android:textColor="@android:color/background_light"
```

```
android:textSize="18sp" />

<Button
android:id="@+id/btn_dec"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@+id/btn_7"
android:layout_toLeftOf="@+id/btn_8"
android:layout_toStartOf="@+id/btn_8"
android:text="."
android:textSize="18sp" />

<Button
android:id="@+id/btn_clear"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentEnd="true"
android:layout_alignParentRight="true"
android:layout_below="@+id/btn_Div"
android:backgroundTint="@android:color/holo_blue_dark"
android:text="clear"
android:textColor="@android:color/background_light"
android:textSize="18sp" />

</RelativeLayout>
</android.support.constraint.ConstraintLayout>
```

Java file

```
package com.example.calculator;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {

Button btn_1,btn_2,btn_3,btn_4,btn_5,btn_6,btn_7,btn_8,btn_9,btn_0,btn_Add,btn_Sub,btn_Mul,btn_Div,btn_calc,btn_dec,btn_clear;
EditText ed1;

float Value1, Value2;
boolean mAddition, mSubtract, mMultiplication, mDivision ;

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

btn_0 = (Button) findViewById(R.id.btn_0);
btn_1 = (Button) findViewById(R.id.btn_1);
btn_2 = (Button) findViewById(R.id.btn_2);
btn_3 = (Button) findViewById(R.id.btn_3);
btn_4 = (Button) findViewById(R.id.btn_4);
btn_5 = (Button) findViewById(R.id.btn_5);
btn_6 = (Button) findViewById(R.id.btn_6);
```

```
btn_7 = (Button) findViewById(R.id.btn_7);
btn_8 = (Button) findViewById(R.id.btn_8);
btn_9 = (Button) findViewById(R.id.btn_9);
btn_Add = (Button) findViewById(R.id.btn_Add);
btn_Div = (Button) findViewById(R.id.btn_Div);
btn_Sub = (Button) findViewById(R.id.btn_Sub);
btn_Mul = (Button) findViewById(R.id.btn_Mul);
btn_calc = (Button) findViewById(R.id.btn_calc);
btn_dec = (Button) findViewById(R.id.btn_dec);
btn_clear = (Button) findViewById(R.id.btn_clear);
ed1 = (EditText) findViewById(R.id.edText1);

btn_0.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
ed1.setText(ed1.getText()+"0");
}
});

btn_1.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
ed1.setText(ed1.getText()+"1");
}
});

btn_2.setOnClickListener(new View.OnClickListener() {
```

```
@Override
public void onClick(View v) {
ed1.setText(ed1.getText()+"2");
}
});

btn_3.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
ed1.setText(ed1.getText()+"3");
}
});

btn_4.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
ed1.setText(ed1.getText()+"4");
}
});

btn_5.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
ed1.setText(ed1.getText()+"5");
}
});
```

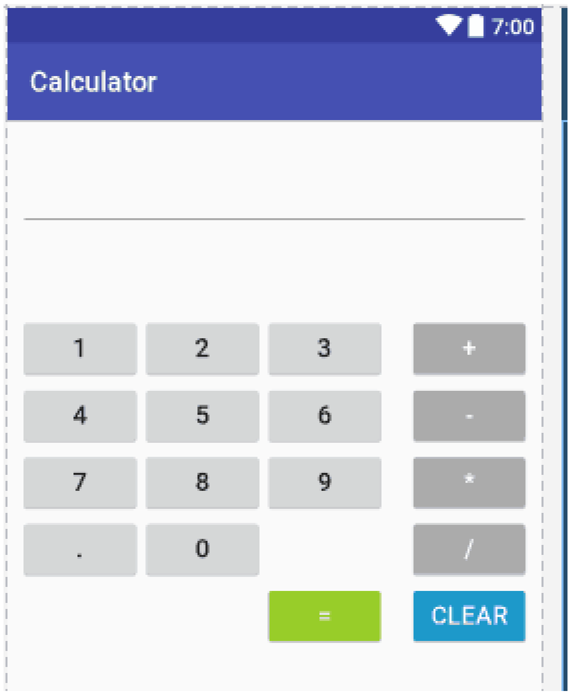
```
btn_6.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
ed1.setText(ed1.getText()+"6");
}
});

btn_7.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
ed1.setText(ed1.getText()+"7");
}
});

btn_8.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
ed1.setText(ed1.getText()+"8");
}
});

btn_9.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
ed1.setText(ed1.getText()+"9");
}
});
```

```
btn_dec.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
ed1.setText(ed1.getText()+".");
}
});

btn_Add.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
if (ed1 == null){
ed1.setText("");
}else {
Value1 = Float.parseFloat(ed1.getText() + "");
mAddition = true;
ed1.setText(null);
}
}
});

btn_Sub.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
Value1 = Float.parseFloat(ed1.getText() + "");
mSubtract = true ;
```

```
ed1.setText(null);
}
});

btn_Mul.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
Value1 = Float.parseFloat(ed1.getText() + "");
mMultiplication = true ;
ed1.setText(null);
}
});

btn_Div.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
Value1 = Float.parseFloat(ed1.getText()+"");
mDivision = true ;
ed1.setText(null);
}
});

btn_calc.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
Value2 = Float.parseFloat(ed1.getText() + "");
```

```
if (mAddition == true){

ed1.setText(Value1 + Value2 +"");
mAddition=false;
}

if (mSubtract == true){
ed1.setText(Value1 - Value2 +"");
mSubtract=false;
}
if (mMultiplication == true){
ed1.setText(Value1 * Value2 + "");
mMultiplication=false;
}
if (mDivision == true){
ed1.setText(Value1 / Value2+"");
mDivision=false;
}
}
});


btn_clear.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
ed1.setText("");
}
});
```

```
}
}
```


7:00
Calculator
1
2
3
+
4
5
6
-
7
8
9
*
.
0
/
=
CLEAR

Practical No -10

Theory

1. Name the file in which respective XML component can be added.

   Ans:

   The AndroidManifest.xml file contains information of your package, including components of the application such as activities, services, broadcast receivers, content providers etc.

   Layout xml file component can be added in res, layout directory.

2. List all the UI components which can be used to develop login window.

   Ans:

   UI components be used to develop login window:

   1. EditText
   2. TextView
   3. Button

- Program:

  1. Write a program to create a login form for a social networking site.

  Ans:

  Activity_main.xml:

```
<?xmlversion="1.0"encoding="utf-8"?>
<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
```

```
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
tools:context=".MainActivity">
<TextView
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:layout_marginTop="40sp"
   android:text="FacebookLoginForm"
   android:textAlignment="center"
   android:textColor="@color/colorAccent"
   android:textSize="36sp"
   android:gravity="center_horizontal"/>
<LinearLayout
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:orientation="horizontal">
      <TextView
         android:layout_width="match_parent"
         android:layout_height="match_parent"
         android:layout_marginLeft="30sp"
         android:layout_marginTop="70dp"
         android:text="Username:"
         android:textColor="@color/colorAccent"
         android:textSize="30sp"/>
      <EditText
```

```
            android:id="@+id/edit1"
            android:layout_width="200sp"
            android:layout_height="match_parent"
            android:layout_marginTop="70dp"
            android:hint="Enterusername"
            android:textAlignment="center"
            android:textColor="@color/colorPrimary"
            android:textSize="20sp"
            android:gravity="center_horizontal"/>
    </LinearLayout>
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_marginLeft="30sp"
            android:layout_marginTop="50dp"
            android:text="Password:"
            android:textColor="@color/colorAccent"
            android:textSize="30sp"/>
        <EditText
            android:id="@+id/edit2"
            android:layout_width="200sp"
            android:layout_height="match_parent"
            android:layout_marginTop="50dp"
```

```
                    android:hint="Enterpassword"
                    android:inputType="textPassword"
                    android:textAlignment="center"
                    android:textColor="@color/colorPrimary"
                    android:textSize="20sp"
                    android:gravity="center_horizontal"/>
                </LinearLayout>
                <Button
                  android:id="@+id/button1"
                  android:layout_width="158dp"
                  android:layout_height="45dp"
                  android:layout_marginLeft="130sp"
                  android:layout_marginTop="50sp"
                  android:background="@color/colorPrimary"
                  android:text="Submit"
                  android:textSize="30sp"/>
            </LinearLayout>
```

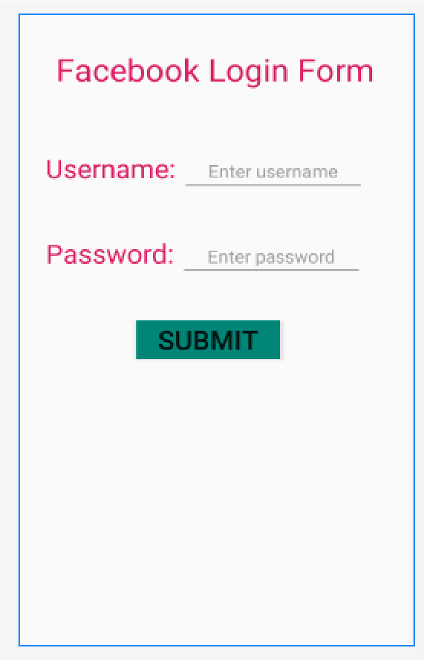
MainActivity.java

```
package com.example.loginform;
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;
```

```
public class MainActivity extends AppCompatActivity
{
   EditText e1,e2;
   Button button;
   @Override
   protected void onCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      e1=findViewById(R.id.edit1);
      e2=findViewById(R.id.edit2);
      button=findViewById(R.id.button1);
      button.setOnClickListener(new View.OnClickListener()
      {
         @Override
         public void onClick(View v)
         {
            String s1=e1.getText().toString();
            String s2=e2.getText().toString();
            if(s1.equals("admin")&&s2.equals("admin")) {
               Toast.makeText(MainActivity.this, "SuccessfullyLogin",
Toast.LENGTH_SHORT).show();
            }
            else {
               Toast.makeText(MainActivity.this, "Failed Login",
Toast.LENGTH_SHORT).show();
```

```
                }
            }
        });
    }
}
```

Output:



Facebook Login Form

Username: Enter username

Password: Enter password

2. Write a program to create a login form for a student registration system.

   Ans:

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"
tools:context=".MainActivity">
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="40sp"
    android:text="Login Form"
    android:textAlignment="center"
    android:textColor="@color/colorAccent"
    android:textSize="40sp"
    android:gravity="center_horizontal" />
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginLeft="20sp"
        android:layout_marginTop="70dp"
        android:text="Enrollment no.: "
        android:textColor="@color/colorAccent"
        android:textSize="20sp"
        android:textStyle="bold"/>

    <EditText
```

```
        android:id="@+id/edit1"
        android:layout_width="250sp"
        android:layout_height="match_parent"
        android:layout_marginTop="70dp"
        android:hint="Enter username"
        android:textColor="@color/colorPrimary"
        android:textSize="17sp" />
</LinearLayout>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginLeft="30sp"
        android:layout_marginTop="50dp"
        android:text="Password: "
        android:textColor="@color/colorAccent"
        android:textSize="20sp"
        android:textStyle="bold"/>
    <EditText
        android:id="@+id/edit2"
        android:layout_width="250sp"
        android:layout_height="match_parent"
        android:layout_marginTop="50dp"
        android:hint="Enter password"
```

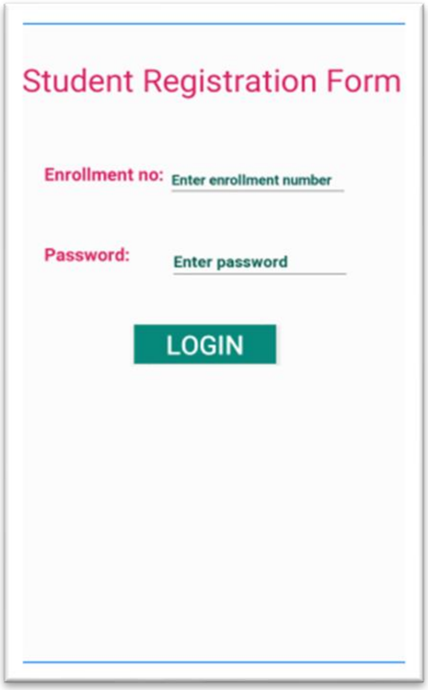
```
            android:inputType="textPassword"
            android:textColor="@color/colorPrimary"
            android:textSize="17sp"
            android:layout_marginLeft="30dp"   />
    </LinearLayout>
    <Button
        android:id="@+id/button1"
        android:layout_width="158dp"
        android:layout_height="45dp"
        android:layout_marginLeft="130sp"
        android:layout_marginTop="50sp"
        android:background="@color/colorPrimary"
        android:text="Submit"
        android:textSize="30sp" />
</LinearLayout>
```

MainActivity.java

```
package com.example.loginform;
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;
public class MainActivity extends AppCompatActivity
{
    EditText e1,e2;
    Button button;
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        e1=findViewById(R.id.edit1);
        e2=findViewById(R.id.edit2);
        button=findViewById(R.id.button1);

        button.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                String s1=e1.getText().toString();
                String s2=e2.getText().toString();
                if(s1.equals("1710740003")&&s2.equals("Student123")) {
                    Toast.makeText(MainActivity.this, "Successfully Login",
Toast.LENGTH_SHORT).show();

                }
                else {
                    Toast.makeText(MainActivity.this, "Failed Login",
Toast.LENGTH_SHORT).show();
```

```
                }
            }
        });
    }
}
```

Output:



Student Registration Form

Enrollment no: Enter enrollment number

Password: Enter password

Practical no:11

Develope a program to implement the check box.

1)Name the Different methods of Checkbox.

Method	Description
public boolean isChecked()	Returns true if it is checked otherwise false.
public void setChecked(boolean status)	Changes the state of the CheckBox.

2)List the attributes of Checkbox.

android:id

android:checked

android:gravity

android:text

android:textColor

android:textSize

android:textStyle

android:background

android:padding

android:onClick

3)Write the xml Tag to createt a ckeckbox named"Android".

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <CheckBox android:id="@+id/Android"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/Android"
        android:onClick="onCheckboxClicked"/>
   </LinearLayout>
```

1.WAP to show five checkboxes and toast selected checkbox

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

```
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity">
<TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Select Your Programming language: "
        android:textColor="#f00"
        android:textSize="20sp"
        android:textStyle="bold" />
<LinearLayout
        android:id="@+id/linearLayout"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="30dp"
        android:background="#e0e0e0"
        android:orientation="vertical">
<CheckBox
            android:id="@+id/androidCheckBox"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerHorizontal="true"
            android:checked="false"
            android:padding="20dp"
            android:text="@string/android"
            android:textColor="#44f"
            android:textSize="20sp"
            android:textStyle="bold|italic" />
<CheckBox
            android:id="@+id/javaCheckBox"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerHorizontal="true"
            android:checked="false"
            android:padding="20dp"
            android:text="@string/java"
            android:textColor="#f44"
            android:textSize="20sp"
            android:textStyle="bold|italic" />
<CheckBox
            android:id="@+id/phpCheckBox"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerHorizontal="true"
            android:checked="false"
            android:padding="20dp"
            android:text="@string/php"
            android:textColor="#444"
            android:textSize="20sp"
            android:textStyle="bold|italic" />
 <CheckBox
            android:id="@+id/pythonCheckBox"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerHorizontal="true"
            android:checked="false"
            android:padding="20dp"
            android:text="@string/python"
            android:textColor="#888"
            android:textSize="20sp"
            android:textStyle="bold|italic" />
<CheckBox
            android:id="@+id/unityCheckBox"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerHorizontal="true"
            android:checked="false"
            android:padding="20dp"
            android:text="@string/unity"
            android:textColor="#101010"
            android:textSize="20sp"
            android:textStyle="bold|italic" />
  </LinearLayout></RelativeLayout>
```

Java file:

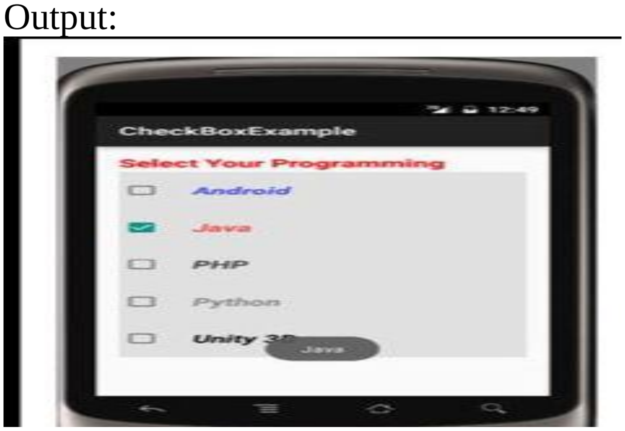
```
package example.abhiandriod.checkboxexample;
import android.graphics.Color;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity implements
View.OnClickListener {
CheckBox android, java, python, php, unity3D;
@Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // initiate views
        android = (CheckBox) findViewById(R.id.androidCheckBox);
        android.setOnClickListener(this);
        java = (CheckBox) findViewById(R.id.javaCheckBox);
        java.setOnClickListener(this);
        python = (CheckBox) findViewById(R.id.pythonCheckBox);
        python.setOnClickListener(this);
        php = (CheckBox) findViewById(R.id.phpCheckBox);
        php.setOnClickListener(this);
        unity3D = (CheckBox) findViewById(R.id.unityCheckBox);
        unity3D.setOnClickListener(this); }
 @Override
    public void onClick(View view) {

        switch (view.getId()) {
            case R.id.androidCheckBox:
                if (android.isChecked())
                    Toast.makeText(getApplicationContext(), "Android",
Toast.LENGTH_LONG).show();
                break;
            case R.id.javaCheckBox:
                if (java.isChecked())
```

```
          Toast.makeText(getApplicationContext(), "Java",
Toast.LENGTH_LONG).show();
        break;
      case R.id.phpCheckBox:
        if (php.isChecked())
          Toast.makeText(getApplicationContext(), "PHP",
Toast.LENGTH_LONG).show();
        break;
      case R.id.pythonCheckBox:
        if (python.isChecked())
          Toast.makeText(getApplicationContext(), "Python",
Toast.LENGTH_LONG).show();
        break;
      case R.id.unityCheckBox:
        if (unity3D.isChecked())
          Toast.makeText(getApplicationContext(), "Unity 3D",
Toast.LENGTH_LONG).show();
        break;
    }}}
```

Resources

```
<resources>
  <string name="app_name">CheckBoxExample</string>
  <string name="hello_world">Hello world!</string>
  <string name="action_settings">Settings</string>
  <string name="android">Android</string>
  <string name="java">Java</string>
  <string name="php">PHP</string>
  <string name="python" >Python</string>
  <string name="unity">Unity 3D</string>
</resources>
```

Output:



CheckBoxExample

Select Your Programming

☐ Android

☑ Java

☐ PHP

☐ Python

☐ Unity 3

Java

Practical No:12

1)Develope a Program to implement radio Button and Radio Group.

Write the xml Tag to create a radio button.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <RadioGroup
        android:id="@+id/radioSex"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" >

        <RadioButton
            android:id="@+id/radioMale"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/radio_male"
            android:checked="true" />

        <RadioButton
            android:id="@+id/radioFemale"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/radio_female" />

    </RadioGroup>

    </LinearLayout>
```

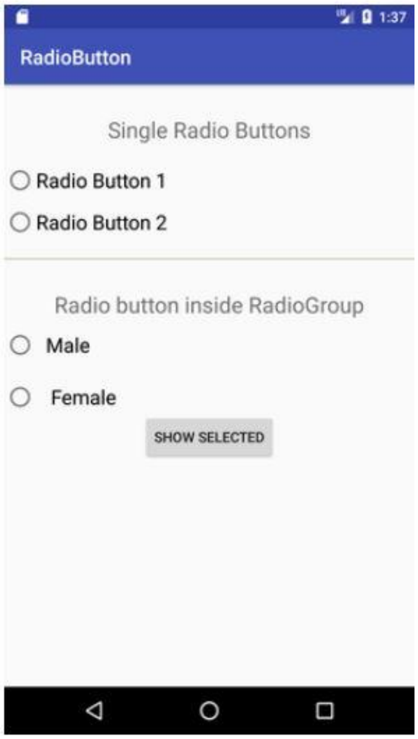
2)Write a purpose of radio button.

A radio button is an element usually found in forms and its main purpose is to allow the user to select a single option from a group of options.

3)List different methods of radiobutton.

1. getcheckedRadioButtonId()
2. selectedId()
3. clearCheck()

1)Write a program to show the following output.First two radio buttons are without using radio group and next radio buttons are using radio group. Note the changes between these two, Also toast which radio button has been selected.



XML File

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

<LinearLayout

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

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

    android:layout_width="match_parent"

    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">
  <TextView
        android:id="@+id/textView1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="30dp"
        android:gravity="center_horizontal"
        android:textSize="22dp"
        android:text="Single Radio Buttons" />
    <RadioButton
        android:id="@+id/radioButton1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:text="Radio Button 1"
        android:layout_marginTop="20dp"
         android:textSize="20dp" />
    <RadioButton
      android:id="@+id/radioButton2"
        android:layout_width="fill_parent"
```

```
         android:layout_height="wrap_content"

         android:text="Radio Button 2"

         android:layout_marginTop="10dp"

        android:textSize="20dp" />

    <View

         android:layout_width="fill_parent"

         android:layout_height="1dp"

         android:layout_marginTop="20dp"

         android:background="#B8B894" />

        <TextView

         android:id="@+id/textView2"

         android:layout_width="fill_parent"

         android:layout_height="wrap_content"

         android:layout_marginTop="30dp"

         android:gravity="center_horizontal"

         android:textSize="22dp"

         android:text="Radio button inside RadioGroup" />

         <RadioGroup

         android:layout_width="wrap_content"

         android:layout_height="wrap_content"

         android:id="@+id/radioGroup">
    <RadioButton
        android:id="@+id/radioMale"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
       android:text="  Male"
       android:layout_marginTop="10dp"
       android:checked="false"
      android:textSize="20dp" />
      <RadioButton
        android:id="@+id/radioFemale"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="   Female"
        android:layout_marginTop="20dp"
        android:checked="false"
        android:textSize="20dp" />
</RadioGroup
     <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Show Selected"
```

```
            android:id="@+id/button"

            android:onClick="onclickbuttonMethod"

             android:layout_gravity="center_horizontal" />

    </LinearLayout>
```

Java file:

```
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Toast;

import com.example.myapplication.AppCompatActivity;
import com.example.radiop.R

public class MainActivity extends AppCompatActivity {
    Button button;
    RadioButton genderradioButton;
    RadioGroup radioGroup;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
radioGroup=(RadioGroup)findViewByid(R.id.radioGroup);
    }
    public void onclickbuttonMethod(View v){
        int selectedId = radioGroup.getCheckedRadioButtonId();
        genderradioButton = (RadioButton) findViewById(selectedId);
        if(selectedId==-1){
            Toast.makeText(MainActivity.this,"Nothing selected",
Toast.LENGTH_SHORT).show();
        }
        else{
            Toast.makeText(MainActivity.this,genderradioButton.getText(),
Toast.LENGTH_SHORT).show();}}}
```

Practical 13

Theory Questions:

State the different methods to update the percentage of progress bar displayed.

void	drawableHotspotChanged(float x, float y) This function is called whenever the view hotspot changes and needs to be propagated to drawables or child views managed by the view.
CharSequence	getAccessibilityClassName() Return the class name of this object to be used for accessibility purposes.
Drawable	getCurrentDrawable() Returns the drawable currently used to draw the progress bar.
Drawable	getIndeterminateDrawable() Get the drawable used to draw the progress bar in indeterminate mode.
BlendMode	getIndeterminateTintBlendMode() Returns the blending mode used to apply the tint to the indeterminate drawable, if specified.
ColorStateList	getIndeterminateTintList()
PorterDuff.Mode	getIndeterminateTintMode() Returns the blending mode used to apply the tint to the indeterminate drawable, if specified.
Interpolator	getInterpolator() Gets the acceleration curve type for the indeterminate animation.

2.Write an xml tag for the determine the progress bar.

```
<ProgressBar
   android:id="@+id/pBar3"
   style="?android:attr/progressBarStyleHorizontal"
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:minHeight="50dp"
    android:minWidth="250dp"
    android:max="100"
    android:indeterminate="true"
    android:progress="1" />
```

3.List differnet progress bar styles provided by the system.

There are two types of progress bars and for each type android provides material styles. If total duration it takes to complete an operation is known, the progress bar which is used to show progress of such operation is called determinate progress bar or horizontal progress bar.

Program:

1.Write a program to display circular progress bar.

Xml file

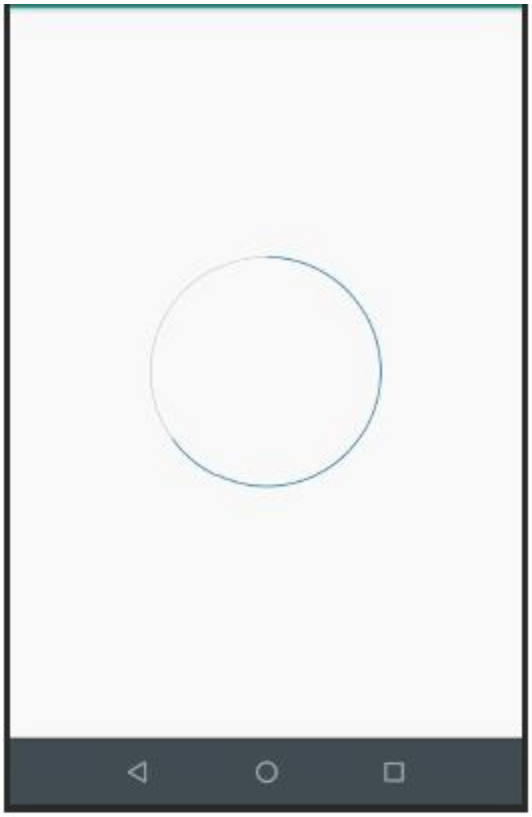
```
<? xml version= "1.0" encoding= "utf-8" ?>
<RelativeLayout xmlns: android = "http://schemas.android.com/apk/res/android"
   xmlns: tools = "http://schemas.android.com/tools"
   android :layout_width= "match_parent"
   android :layout_height= "match_parent"
   android :layout_margin= "16dp"
   tools :context= ".MainActivity" >
   <ProgressBar
      android :id= "@+id/progressBar"
      style= "?android:attr/progressBarStyleHorizontal"
      android :layout_width= "200dp"
      android :layout_height= "200dp"
      android :layout_centerInParent= "true"
      android :background= "@drawable/circular_shape"
      android :indeterminate= "false"
      android :max= "100"
      android :progress= "65"
      android :progressDrawable= "@drawable/circular_progress_bar" />
</RelativeLayout>
```

Java file

```
package app.tutorialspoint.com.sample ;
```

```
import android.support.v7.app.AppCompatActivity ;
import android.os.Bundle ;
public class MainActivity extends AppCompatActivity {
  @Override
  protected void onCreate (Bundle savedInstanceState) {
    super .onCreate(savedInstanceState) ;
    setContentView(R.layout. activity_main ) ;
  }
}
```

Output:



2.Write a program for horizontal progress bar.

```
<RelativeLayout xmlns:androclass="http://schemas.android.com/apk/res/and
roid"

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

   android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"

    tools:context=".MainActivity" >


    <Button

        android:id="@+id/button1"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:layout_alignParentTop="true"

        android:layout_centerHorizontal="true"

        android:layout_marginTop="116dp"

        android:text="download file" />


</RelativeLayout>
```

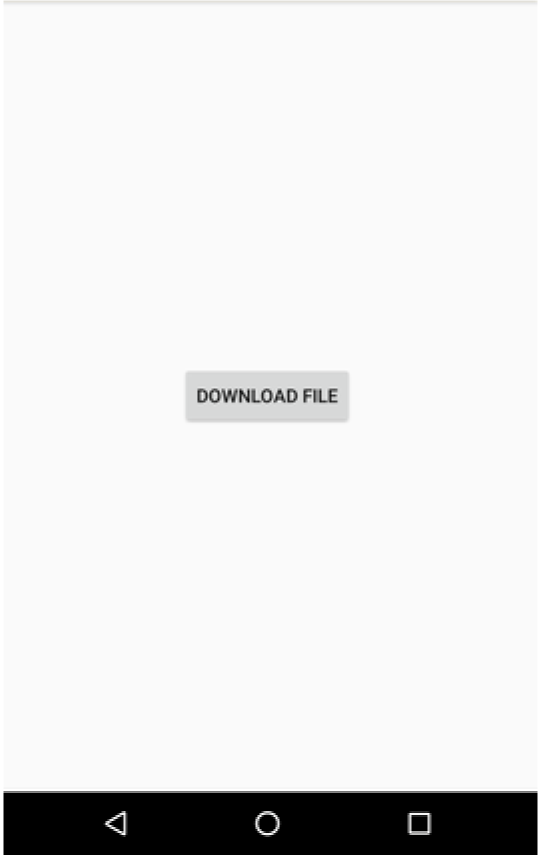
Java file

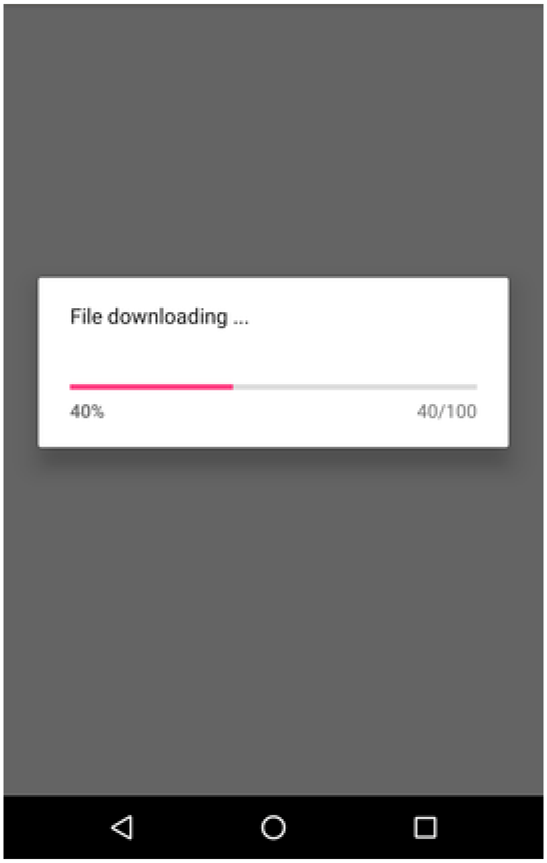
```
package example.javatpoint.com.progressbar;


import android.app.ProgressDialog;

import android.os.Handler;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;
```

```
public class MainActivity extends AppCompatActivity {
    Button btnStartProgress;
    ProgressDialog progressBar;
    private int progressBarStatus = 0;
    private Handler progressBarHandler = new Handler();
    private long fileSize = 0;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        addListenerOnButtonClick();
    }
    public void addListenerOnButtonClick() {
        btnStartProgress = findViewById(R.id.button);
        btnStartProgress.setOnClickListener(new View.OnClickListener(){

            @Override
            public void onClick(View v) {
                                progressBar = new ProgressDialog(v.getContext());
      progressBar.setCancelable(true);
```

```
                progressBar.setProgressStyle(ProgressDialog.STYLE_HORIZONTA
L);
                progressBar.setProgress(0);
                    progressBar.setMax(100);
                    progressBar.show();
                    progressBarStatus = 0;
                 fileSize = 0;


                    new Thread(new Runnable() {
                        public void run() {
                            while (progressBarStatus < 100) {
                                progressBarStatus = doOperation();
                                try {
                                    Thread.sleep(1000);
                                } catch (InterruptedException e) {
                                    e.printStackTrace();
                                }
                                progressBarHandler.post(new Runnable() {
                                    public void run() {
                                        progressBar.setProgress(progressBarStatus);
                                    }
                                });
                            }
                            if (progressBarStatus >= 100) {
                                try {
                                    Thread.sleep(1000);
                                } catch (InterruptedException e) {
                                    e.printStackTrace();
                                }
                                progressBar.dismiss();
                            }
                        }
                    }).start();
                        });
        }
        public int doOperation() {
                    while (fileSize <= 10000) {
                 fileSize++;
                 if (fileSize == 1000) {
                    return 10;
                 } else if (fileSize == 2000) {
                    return 20;
                 } else if (fileSize == 3000) {
                return 30;
            } else if (fileSize == 4000) {
                return 40; // you can add more else if
            }
    {
    }
```

Output:




File downloading ...

Practical No 14

Develop a program to implement List view,Grid view ,Image view and scroll view.

Theory questions:

1.List all attributes of image view:

android:adjustViewBounds	Set this to true if you want the ImageView to adjust its bounds to preserve the aspect ratio of its drawable.
android:baseline	The offset of the baseline within this view.
android:baselineAlignBottom	If true, the image view will be baseline aligned with based on its bottom edge.
android:cropToPadding	If true, the image will be cropped to fit within its padding.
android:maxHeight	An optional argument to supply a maximum height for this view.
android:maxWidth	An optional argument to supply a maximum width for this view.
android:scaleType	Controls how the image should be resized or moved to match the size of this ImageView.
android:src	Sets a drawable as the content of this ImageView.
android:tint	The tinting color for the image.
android:tintMode	Blending mode used to apply the image tint.

2.Describe android:stretchMode attribute of grid view in detail.

Defines how columns should stretch to fill the available empty space, if any.

Program:

1.write program for list view.

Xml file

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

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="List view"
        android:textSize="40dp"/>
    <ListView
        android:id="@+id/listView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >
    </ListView>

</LinearLayout>
```

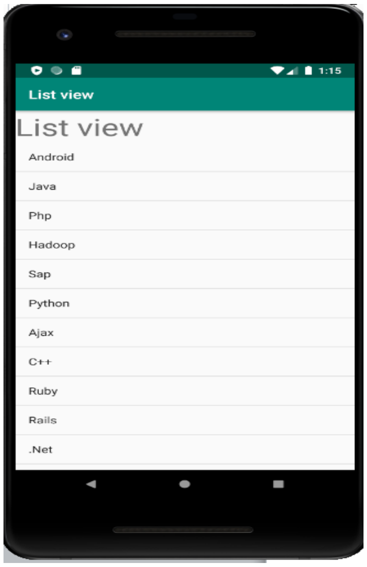
String.xml

```
<resources>
    <string name="app_name">List view</string>
    <string-array name="array_technology">
        <item>Android</item>
        <item>Java</item>
        <item>Php</item>
        <item>Hadoop</item>
        <item>Sap</item>
        <item>Python</item>
        <item>Ajax</item>
        <item>C++</item>
        <item>Ruby</item>
        <item>Rails</item>
        <item>.Net</item>
        <item>Perl</item>
    </string-array>
</resources>
```

MainActivity.java

```
package com.example.listview;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    ListView listView;
    TextView textView;
    String[] listItem;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        textView=(findViewById(R.id.textView));
        listItem = getResources().getStringArray(R.array.array_technology);

        final ArrayAdapter adapter = new ArrayAdapter<String>(this,

                android.R.layout.simple_list_item_1, android.R.id.text1,listItem);

         listView = (ListView) findViewById(R.id.listView);
        listView.setAdapter(adapter);
        listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position,
long id)
            {

                Toast.makeText(MainActivity.this, adapter.getItem(position).toString(),
Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

Output:



List view

List view

Android
Java
Php
Hadoop
Sap
Python
Ajax
C++
Ruby
Rails
.Net

2.Write a program to display an image using image view and a button named as “change image’ .once you click on button another image should get displayed.

Program:

Activity_main.xml

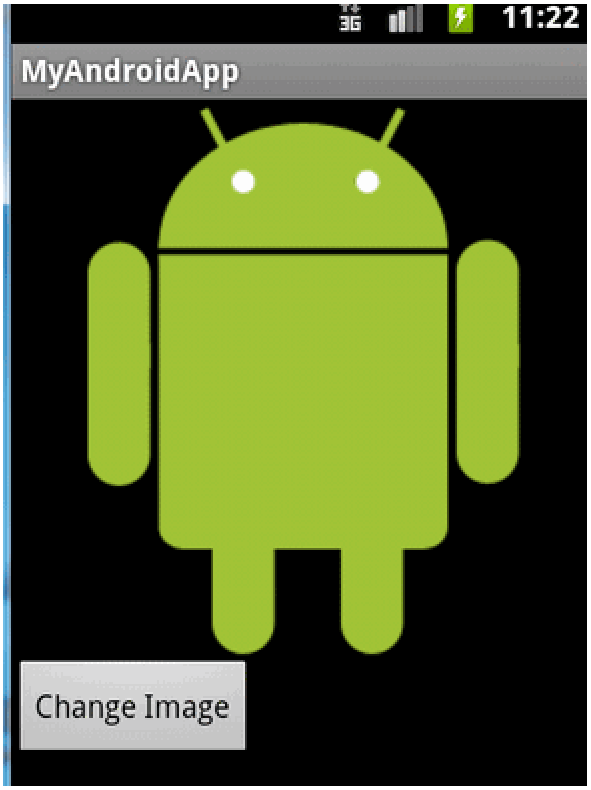
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical">

    <Button
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:text="Upload Image"
        android:padding="16dp"
        android:textColor="#F3EDED"
        android:textSize="21dp"
        android:background="@color/colorAccent"
        android:id="@+id/uploadImage"/>


    <ImageView
```

```
        android:id="@+id/imageView2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="30dp"/>

</LinearLayout>
```

MainActivity.java

```
package com.example.importimage;

import android.content.Intent;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import java.io.IOException;

public class MainActivity extends AppCompatActivity {

    private static final String TAG = MainActivity.class.getSimpleName();

    private int PICK_IMAGE_REQUEST = 1;


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

        Button selectImage = findViewById(R.id.uploadImage);

        selectImage.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                chooseImage();
            }
        });
    }

    public void chooseImage() {
        Intent intent = new Intent();
        intent.setType("image/*");
        intent.setAction(Intent.ACTION_GET_CONTENT);
        startActivityForResult(Intent.createChooser(intent, "Select Picture"),
PICK_IMAGE_REQUEST);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable
Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        if (requestCode == PICK_IMAGE_REQUEST && resultCode ==
RESULT_OK && data != null && data.getData() != null) {

            Uri uri = data.getData();

            try {
                Bitmap bitmap =
MediaStore.Images.Media.getBitmap(getContentResolver(), uri);
                // Log.d(TAG, String.valueOf(bitmap));

                ImageView imageView = findViewById(R.id.imageView2);
                imageView.setImageBitmap(bitmap);
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

Output:



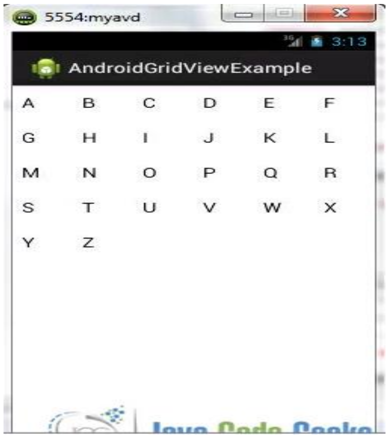
3.write a program to display 15 buttons using grid view.

```
<?xml version="1.0" encoding="utf-8"?>
<GridView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/gridView"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:columnWidth="50dp"
    android:gravity="center"
    android:numColumns="auto_fit"
    android:stretchMode="columnWidth" >

</GridView>
```

Main_Activity.java

```
package com.javacodegeeks.android.androidgridviewexample;

import android.app.Activity;
import android.os.Bundle;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.GridView;
import android.widget.TextView;
import android.widget.Toast;
import android.view.View;
import android.widget.AdapterView.OnItemClickListener;

public class MainActivity extends Activity {

    GridView grid;

    static final String[] letters = new String[] {
            "A", "B", "C", "D", "E",
            "F", "G", "H", "I", "J",
            "K", "L", "M", "N", "O",
            "P", "Q", "R", "S", "T",
            "U", "V", "W", "X", "Y", "Z"};

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.main);

        grid = (GridView) findViewById(R.id.gridView);

        ArrayAdapter adapter = new ArrayAdapter(this,
android.R.layout.simple_list_item_1, letters);

        grid.setAdapter(adapter);

        grid.setOnItemClickListener(new OnItemClickListener() {
            public void onItemClick(AdapterView parent, View v, int position, long id)
{
               Toast.makeText(getApplicationContext(),
                ((TextView) v).getText(), Toast.LENGTH_SHORT).show();
            }
        });

    }
}
```

Output:



```
A  B  C  D  E  F
G  H  I  J  K  L
M  N  O  P  Q  R
S  T  U  V  W  X
Y  Z
```

4.Write a program for scroll view

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.test.scrollviews.MainActivity">


    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="Vertical ScrollView example"
        android:id="@+id/textView"
        android:layout_gravity="center_horizontal"
        android:layout_centerHorizontal="true"
        android:layout_alignParentTop="true" />


    <ScrollView android:layout_marginTop="30dp"
```

```
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"        android:id="@+id/scrollView">


         <LinearLayout
           android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:orientation="vertical" >

            <Button
                android:layout_width="fill_parent"                android:layout_height="w
rap_content"
                android:text="Button 1" />
            <Button
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:text="Button 2" />
            <Button
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:text="Button 3" />
            <Button
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:text="Button 4" />
            <Button
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:text="Button 5" />
            <Button
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:text="Button 6" />
            <Button
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:text="Button 7" />
            <Button
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
```

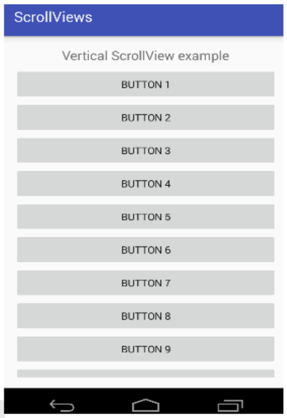
```
    android:text="Button 8" />
<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Button 9" />
<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Button 10" />
<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Button 11" />
<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Button 12" />
<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
   android:text="Button 13" />
<Button
   androd:layout_width="fill_parent"
    andrid:layout_height="wrap_content"
    android:text="Button 14" />
<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Button 15" />
<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Button 16" />
<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Button 17" />
<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
```

```
                android:text="Button 18" />

            <Button
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:text="Button 19" />
            <Button
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:text="Button 20" />
        </LinearLayout>

    </ScrollView>
_
</RelativeLayout>
```

Main_Activity.java

```
package com.example.test.scrollviews;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

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

Output:



Practical No-15

1.List all predefined constants to specify overall positioning of toast .Which method is used to change the positioning of toast message on screen .

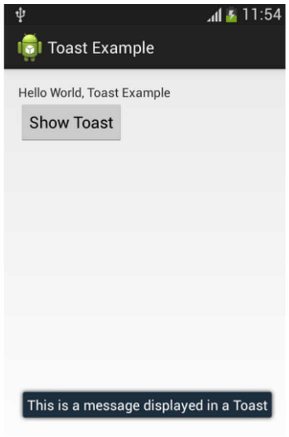
public static Toast makeText(Context context, CharSequence text, int duration)	public static Toast makeText(Context context, CharSequence text, int duration)
public static Toast makeText(Context context, CharSequence text, int duration)	public static Toast makeText(Context context, CharSequence text, int duration)
public static Toast makeText(Context context, CharSequence text, int duration)	public static Toast makeText(Context context, CharSequence text, int duration)

To change the positioning on the screen of a Toast message using the setGravity() method.

2.List two constants of toastclass.

public static final int LENGTH_LONG	displays for a long time
public static final int LENGTH_SHORT	displays for a short time

1.Write aprogram to diplay following output



Hello World, Toast Example

This is a message displayed in a Toast

```
<?xml version="1.0" encoding="utf-8"?>
   <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
              android:id="@+id/toast_root_view"
              android:orientation="vertical" android:layout_width="match_parent"
```

```
                 android:layout_height="match_parent"
                 android:background="@android:color/background_dark"
                 android:padding="16dp">
 <TextView
                android:id="@+id/toast_header"
                android:textSize="20dp"
                android:textColor="@android:color/primary_text_dark"
                android:layout_width="match_parent"
                android:layout_height="match_parent" />
<TextView
                android:id="@+id/toast_body"
                android:textColor="@android:color/primary_text_dark"
                android:layout_width="match_parent"
                android:layout_height="match_parent" />

    </LinearLayout>
```

JAVA: file

```
LayoutInflater inflater = getLayoutInflater();

View toastLayout = inflater.inflate(R.layout.my_toast,
        (ViewGroup) findViewById(R.id.toast_root_view));

TextView header = (TextView) toastLayout.findViewById(R.id.toast_header);
header.setText("Message for you:");

TextView body = (TextView) toastLayout.findViewById(R.id.toast_body);
body.setText("You have got mail!");

Toast toast = new Toast(getApplicationContext());
toast.setGravity(Gravity.CENTER, 0, 0);
toast.setDuration(Toast.LENGTH_LONG);
toast.setView(toastLayout);
toast.show();
```

2)WAP to display three checkbox and one button named order as show below. Once you click on button it should toast different selected checkboxes along with items individual and total price.

XML file

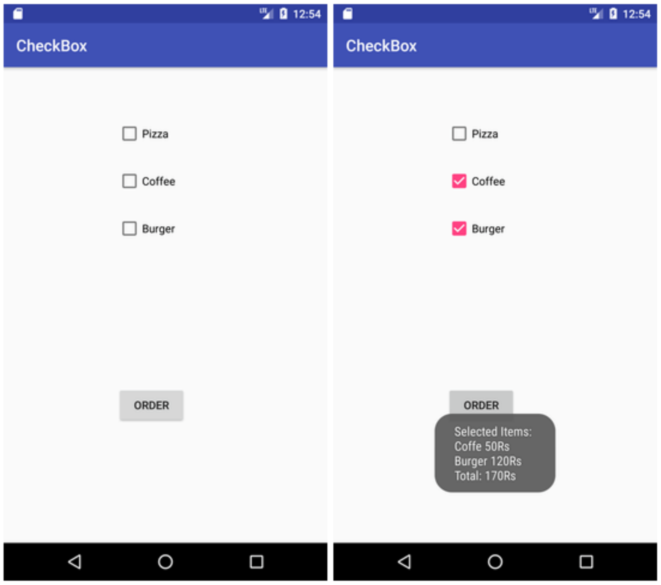
```
<?xml version="1.0" encoding="utf-8"?>
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
     xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context="example.javatpoint.com.checkbox.MainActivity">
      <CheckBox
           android:id="@+id/checkBox"
           android:layout_width="wrap_content"
           android:layout_height="wrap_content"
           android:layout_marginLeft="144dp"
           android:layout_marginTop="68dp"
           android:text="Pizza"
           app:layout_constraintStart_toStartOf="parent"
           app:layout_constraintTop_toTopOf="parent" />

        <CheckBox
           android:id="@+id/checkBox2"
           android:layout_width="wrap_content"
           android:layout_height="wrap_content"
           android:layout_marginLeft="144dp"
           android:layout_marginTop="28dp"
           android:text="Coffee"
           app:layout_constraintStart_toStartOf="parent"
           app:layout_constraintTop_toBottomOf="@+id/checkBox" />

        <CheckBox
           android:id="@+id/checkBox3"
           android:layout_width="wrap_content"
           android:layout_height="wrap_content"
           android:layout_marginLeft="144dp"
```

```
            android:layout_marginTop="28dp"
            android:text="Burger"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/checkBox2" />

    <Button
            android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
            android:layout_marginLeft="144dp"
            android:layout_marginTop="184dp"
            android:text="Order"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/checkBox3" />

 </LinearLayout >
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="example.javatpoint.com.checkbox.MainActivity">


    <CheckBox
 android:id="@+id/checkBox"
        android:layout_width="wrap_content"

        android:layout_height="wrap_content"
   android:layout_marginLeft="144dp"
        android:layout_marginTop="68dp"
        android:text="Pizza"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <CheckBox
        android:id="@+id/checkBox2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="144dp"
```

```
      android:layout_marginTop="28dp"
        android:text="Coffee"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/checkBox" />

    <CheckBox
        android:id="@+id/checkBox3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="144dp"
        android:layout_marginTop="28dp"
        android:text="Burger"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/checkBox2" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="144dp"
        android:layout_marginTop="184dp"
        android:text="Order"
        app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toBottomOf="@+id/checkBox3" />
  </LinearLayout>
```

Java File

```
package example.javatpoint.com.checkbox;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.Toast;
public class MainActivity extnds AppCompatActivity {
    CheckBox pizza,coffe,burger;
    Button buttonOrder;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        addListenerOnButtonClick();
    }
      public void addListenerOnButtonClick(){
        //Getting instance of CheckBoxes and Button from the activty_main.xml file
        pizza=(CheckBox)findViewById(R.id.checkBox);
        coffe=(CheckBox)findViewById(R.id.checkBox2);
        burger=(CheckBox)findViewById(R.id.checkBox3);
        buttonOrder=(Button)findViewById(R.id.button);

        //Applying the Listener on the Button click
        buttonOrder.setOnClickListener(new View.OnClickListener(){

            @Override
            public void onClick(View view) {
                int totalamount=0;
                StringBuilder result=new StringBuilder();
                result.append("Selected Items:");
                if(pizza.isChecked()){
                    result.append("\nPizza 100Rs");
                   totalamount+=100;
                }
                ifcoffe.isChecked()){
                   result.append("\nCoffe 50Rs");
                    totalamount+=50;
                }
                if(burger.isChecked()){
                    result.append("\nBurger 120Rs");
                    totalamount+=120;
                }
                result.append("\nTotal: "+totalamount+"Rs");
                //Displaying the message on the toast
                Toast.makeText(getApplicationContext(), result.toString(), Toast.LENG
TH_LONG
                              ).show();
            }
           });
        }
    }
```



Selected Items:
Coffe 50Rs
Burger 120Rs
Total: 170Rs

Practical No:-16:

Develop a Program to implement Date and Time Picker.

1.Write an xml Timepicker tag with all its attributes.

```
<TimePicker android:id="@+id/timePicker1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:timePickerMode="clock" />
```

2.List and explain all methods of TimePickerclass.

1.is24HourView():- This method returns true if this is in 24 hour view else false.
2.isEnabled():- This method returns the enabled status for this view
3.setCurrentHour(Integer currentHour):-This method sets the current hour
4.setCurrentMinute(Integer currentMinute):-This method sets the current minute
5.setEnabled(boolean enabled):-This method set the enabled state of this view
6.setIs24HourView(Boolean is24HourView):-This method set whether in 24 hour or AM/PM mode
7.setOnTimeChangedListener(TimePicker.OnTimeChangedListener onTimeChangedListener):-This method Set the callback that indicates the time has been adjusted by the user

3.List and explain any five methods of DatePickerclass.

1. setSpinnersShown(boolean shown):- This method is used to determine whether the spinner of the date picker is shown or not. In this method you have to set a Boolean value either true or false. True indicates spinner is shown, false value indicates spinner is not shown. Default value for this function is true.

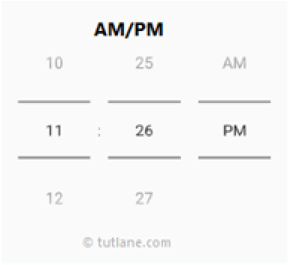
2. getDayOfMonth(): -This method is used to get the selected day of the month from a date picker. This method returns an integer value.
3. getMonth():-This method is used to get the selected month from a date picker. This method returns an integer value.
4. getYear():-This method is used to get the selected year from a date picker. This method returns an integer value.
5. getFirstDayOfWeek():-This method is used to get the first day of the week. This method returns an integer value.

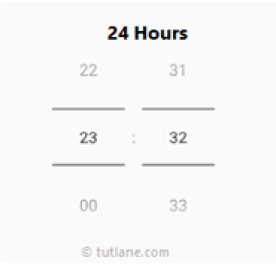
PROGRAM:

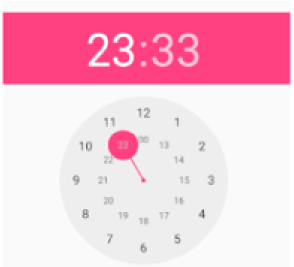
1 .Write a program to display the following output. Use a TimePicker with a spinner.

```
<LinearLayout
   android:layout_width="match_parent"
   android:layout_height=”match_parent"
   tools:context=”.MainActivity”/>
<TimerPicker
   android:id="@+id/datePicker1"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:timePickerMode="spinner"/>
</LinearLayout>
<LinearLayout
   android:layout_width="match_parent"
   android:layout_height=”match_parent"
   tools:context=”.MainActivity”/>
<TimerPicker
   android:id="@+id/datePicker1"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:timePickerMode="spinner"/>
</LinearLayout>
public class MainActivity extends AppCompatActivity
{
  TimePicker picker;
 @Override
   protected void onCreate(Bundle savedInstanceState)
{
      super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
     TimePicker picker=(TimePicker)findViewById(R.id.timePicker1);
     picker.setIs24HourView(true);
 }
}
```

OUTPUT:







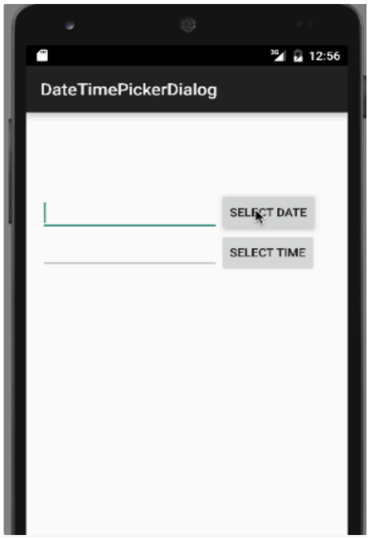
2.Write a program to display following output. Select and display date and time on click of “select date ”,”select time ”button respectively.

```
<RelativeLayout xmlns:android="https://schemas.android.com/apk/res/android"
   xmlns:tools="https://schemas.android.com/tools"
android:layout_width="match_parent"
   android:layout_height="match_parent"
android:paddingLeft="@dimen/activity_horizontal_margin"
   android:paddingRight="@dimen/activity_horizontal_margin"
   android:paddingTop="@dimen/activity_vertical_margin"
   android:paddingBottom="@dimen/activity_vertical_margin"
tools:context=".MainActivity">
<EditText
      android:layout_width="200dp"
      android:layout_height="wrap_content"
      android:id="@+id/in_date"
      android:layout_marginTop="82dp"
      android:layout_alignParentTop="true"
      android:layout_alignParentLeft="true"
      android:layout_alignParentStart="true" />
<Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="SELECT DATE"
      android:id="@+id/btn_date"
      android:layout_alignBottom="@+id/in_date"
      android:layout_toRightOf="@+id/in_date"
      android:layout_toEndOf="@+id/in_date" />
<EditText
```

```
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:id="@+id/in_time"
        android:layout_below="@+id/in_date"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />
<Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="SELECT TIME"
        android:id="@+id/btn_time"
        android:layout_below="@+id/btn_date"
        android:layout_alignLeft="@+id/btn_date"
        android:layout_alignStart="@+id/btn_date" />
</RelativeLayout>
public class MainActivity extends AppCompatActivity implements
View.OnClickListener {
Button btnDatePicker, btnTimePicker;
    EditText txtDate, txtTime;
    private int mYear, mMonth, mDay, mHour, mMinute;
 @Override
    protected void onCreate(Bundle savedInstanceState)
 {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
btnDatePicker=(Button)findViewById(R.id.btn_date);
       btnTimePicker=(Button)findViewById(R.id.btn_time);
        txtDate=(EditText)findViewById(R.id.in_date);
        txtTime=(EditText)findViewById(R.id.in_time);
        btnDatePicker.setOnClickListener(this);
        btnTimePicker.setOnClickListener(this);

    }

    @Override
    public void onClick(View v)
 {
    if (v == btnDatePicker)
 {

      // Get Current Date
      final Calendar c = Calendar.getInstance();
      mYear = c.get(Calendar.YEAR);
      mMonth = c.get(Calendar.MONTH);
      mDay = c.get(Calendar.DAY_OF_MONTH);


      DatePickerDialog datePickerDialog = new DatePickerDialog(this,
          new DatePickerDialog.OnDateSetListener() {

            @Override
            public void onDateSet(DatePicker view, int year,
                       int monthOfYear, int dayOfMonth) {

              txtDate.setText(dayOfMonth + "-" + (monthOfYear + 1) + "-" +
year);

            }
          }, mYear, mMonth, mDay);
      datePickerDialog.show();
    }
    if (v == btnTimePicker) {

      // Get Current Time
      final Calendar c = Calendar.getInstance();
      mHour = c.get(Calendar.HOUR_OF_DAY);
      mMinute = c.get(Calendar.MINUTE);

      // Launch Time Picker Dialog
      TimePickerDialog timePickerDialog = new TimePickerDialog(this,
          new TimePickerDialog.OnTimeSetListener() {

            @Override
            public void onTimeSet(TimePicker view, int hourOfDay,
                       int minute) {

              txtTime.setText(hourOfDay + ":" + minute);
```

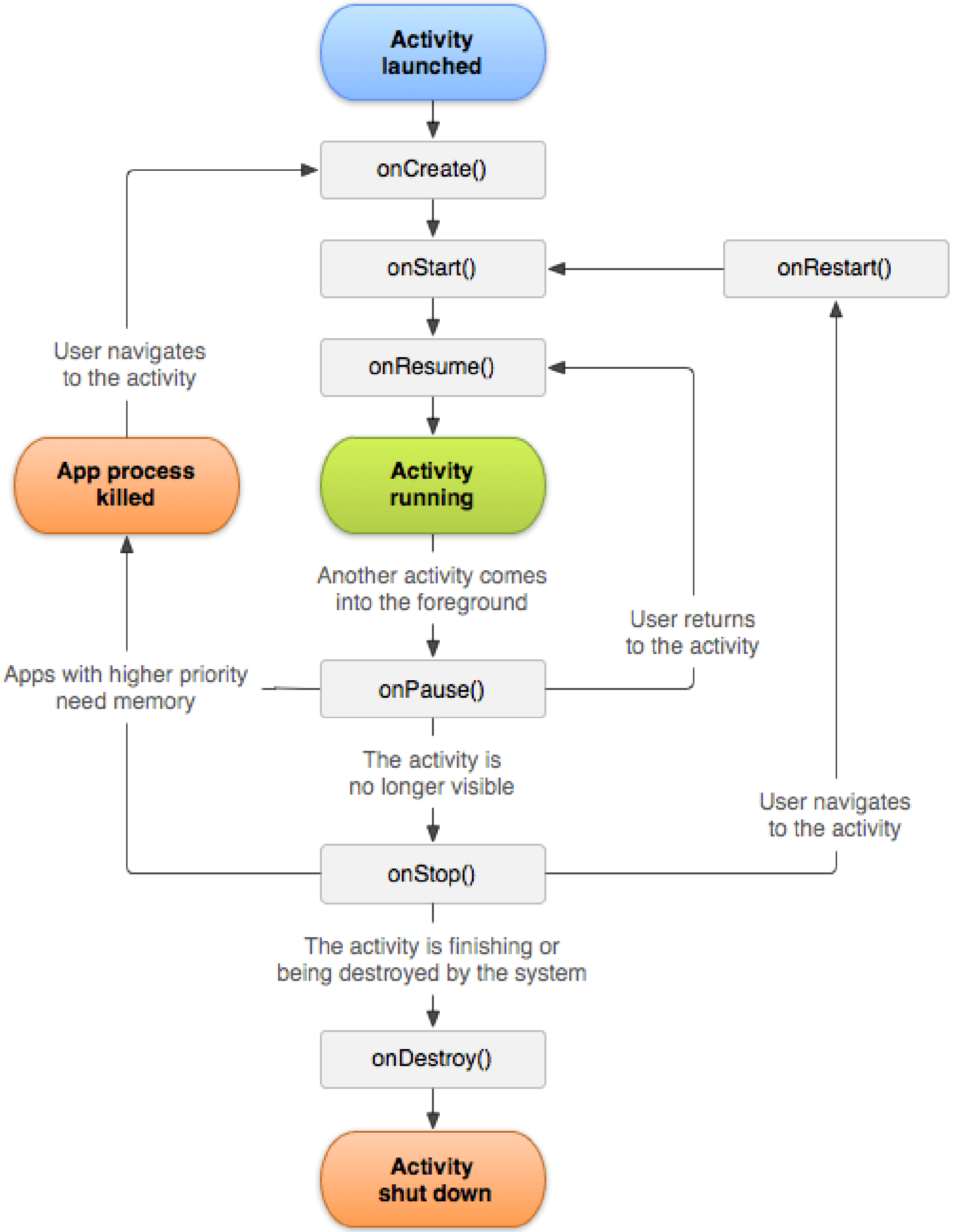
```
                    }
                }, mHour, mMinute, false);
            timePickerDialog.show();
        }
    }
}
```

OUTPUT:

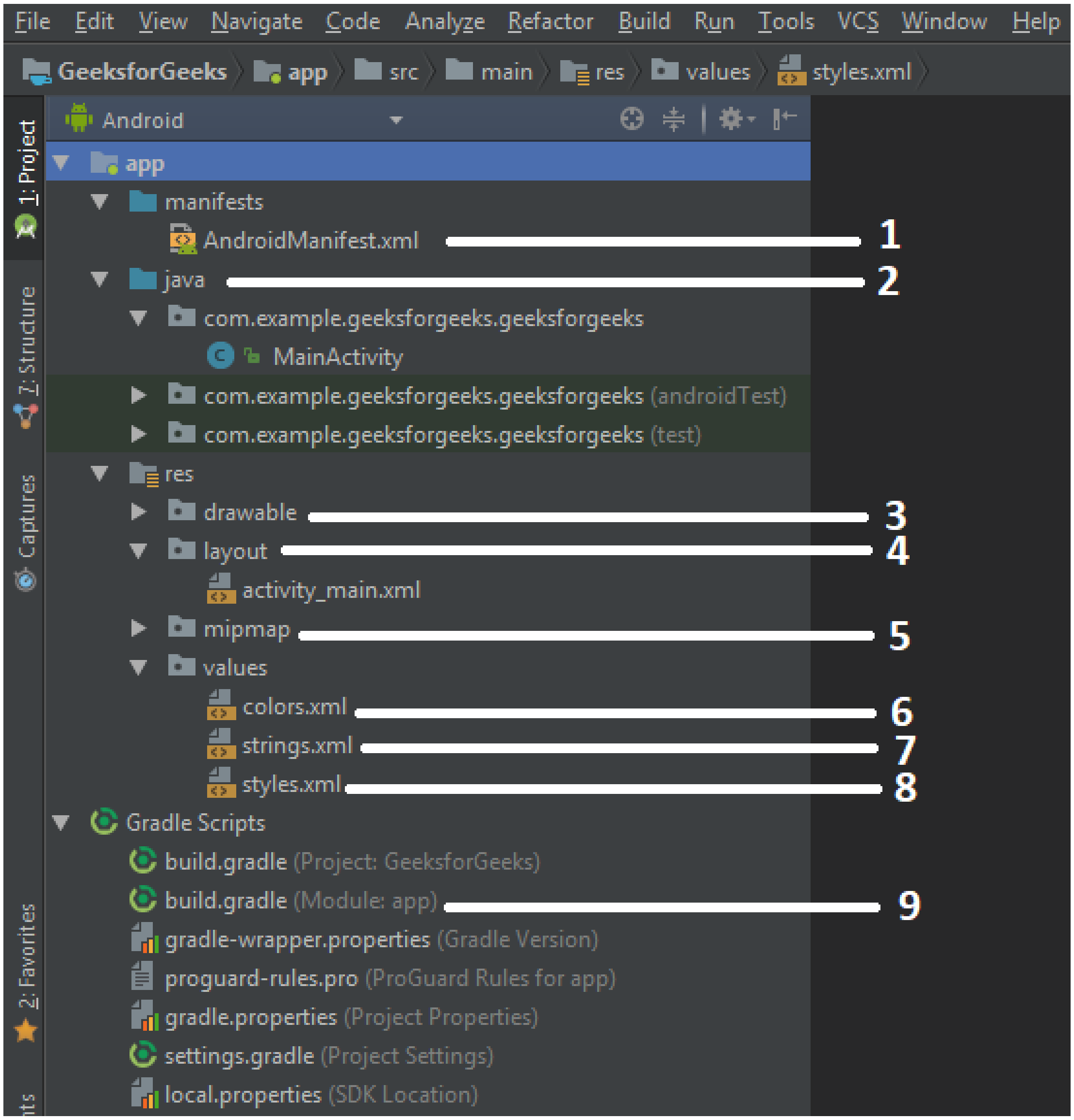


Practical No :-17

1.Draw the activity life cycle diagram.
Ans:


Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

2.Give the hierarchy of directory structure where you store activity file.
Ans:


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
GeeksforGeeks app src main res values styles.xml
1: Project
7: Structure
Captures
2: Favorites
Android
app
manifests
AndroidManifest.xml
1
java
2
com.example.geeksforgeeks.geeksforgeeks
MainActivity
com.example.geeksforgeeks.geeksforgeeks (androidTest)
com.example.geeksforgeeks.geeksforgeeks (test)
res
drawable
3
layout
4
activity_main.xml
mipmap
5
values
colors.xml
6
strings.xml
7
styles.xml
8
Gradle Scripts
build.gradle (Project: GeeksforGeeks)
build.gradle (Module: app)
9
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)

3.Difference between onStop() & onDestroy() methods,also between onPause() & onResume() methods.

Ans:

onStop() & onDestroy()

onPause() is called when an activity is about to lose focus. onStop() is called when the activity is has already lost the focus and it is no longer in the screen. But onPause() is called when the activity is still in the screen, once the method execution is completed then the activity loses focus. So, onPause() is logically before onStop().

From onPause() it is possible to call onResume() but it is not possible once onStop() is called. Once onStop() is called then onRestart() can be called.

onDestroy() is last in the order after onStop(). onDestory() is called just before an activity is destroyed and after that it is gone it is not possible to resurrect this.

onPause() & onResume

When the activity enters the Resumed state, it comes to the foreground, and then the system invokes the onResume() callback. This is the state in which the app interacts with the user. The app stays in this state until something happens to take focus away from the app. Such an event might be, for instance, receiving a phone call, the user’s navigating to another activity, or the device screen’s turning off.

When an interruptive event occurs, the activity enters the Paused state, and the system invokes the onPause() callback.

If the activity returns to the Resumed state from the Paused state, the system once again calls onResume() method. For this reason, you should implement onResume() to initialize components that you release during onPause(), and perform any other initializations that must occur each time the activity enters the Resumed state.

The system calls this method as the first indication that the user is leaving your activity (though it does not always mean the activity is being destroyed); it indicates that the activity is no longer in the foreground (though it may still be visible if the user is in multi-window mode). Use the onPause() method to pause or adjust operations that should not continue (or should continue in moderation) while the Activity is in the Paused state, and that you expect to resume shortly.

1.Write a program to create a HelloWorld Activity using all lifecycle methods to display message using Log.d

File: activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

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

```
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Hello World!"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>
```

Android Activity Lifecycle Example

It provides the details about the invocation of life cycle methods of activity. In this example, we are displaying the content on the logcat.

File: MainActivity.java

```
package com.example.log;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d("lC","oncreate");
    }
    protected void onStart(){
        super.onStart();
        Log.d("LC","Onstart");
    }
    protected void onStop(){
        super.onStop();
        Log.d("LC","Onstop");
    }
    protected void onResume(){
        super.onResume();
        Log.d("LC","onResume");
    }
    protected void onPause(){
        super.onPause();
        Log.d("Lc","onPause");
    }
    protected void onRestart(){
        super.onRestart();
        Log.d("LC","onRestart");
    }
    protected void onDestroy(){
        super.onDestroy();
        Log.d("Lc","onDestroy");
    }
}
```

Practical :18

WAP to create a text field and a button "Navigate". when you enter "www.google.com"and press navigate button it should open google page.

Java file

```
package com.example.emplicityintent;

import androidx.appcompat.app.AppCompatActivity;

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

public class MainActivity extends AppCompatActivity {
EditText et1;
Button button;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        et1=findViewById(R.id.et1);
        button=findViewById(R.id.button);

        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String url=et1.getText().toString();
                Intent intent=new Intent(Intent.ACTION_VIEW, Uri.parse(url));
                startActivity(intent);
            }
        });
    }
}
```

xml file

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

    <EditText
        android:id="@+id/et1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPersonName"
        android:text="Name"
        tools:layout_editor_absoluteX="125dp"
        tools:layout_editor_absoluteY="135dp" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button"
        tools:layout_editor_absoluteX="146dp"
        tools:layout_editor_absoluteY="269dp" />

</LinearLayout>
```

Practical no-20

2) Write a program to display the following output

java file

```
package com.example.third;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
public class Main4Activity extends AppCompatActivity
{
Button start,stop;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main4);
start=findViewById(R.id.ss);
stop=findViewById(R.id.ss1);
start.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
Intent intent = new Intent(Main4Activity.this, MyBackgroundService.class);
startService(intent);
}
});
stop.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
Intent intent = new Intent(Main4Activity.this, MyBackgroundService.class);
stopService(intent);
}
});
}
}
```

service file

```
package com.example.third;
import android.app.Service;
import android.content.Intent;
import android.media.MediaPlayer;
import android.os.IBinder;
import android.widget.Toast;
public class MyBackgroundService extends Service
{
    MediaPlayer mediaPlayer;
    public MyBackgroundService() {
    }
   @Override
   public IBinder onBind(Intent intent) {
        // TODO: Return the communication channel to the service.
        return  null;
    }
    @Override
    public void onCreate() {
        Toast.makeText(this, "Service Created ", Toast.LENGTH_SHORT).show();
        mediaPlayer=MediaPlayer.create(this,R.raw.yes);
        mediaPlayer.setLooping(false);
    }
@Override
    public void onStart(Intent intent, int startId) {
        Toast.makeText(this, "Service Started", Toast.LENGTH_SHORT).show();
        mediaPlayer.start();
    }@Override
    public void onDestroy() {
        super.onDestroy();
        Toast.makeText(this, "Service Stopped", Toast.LENGTH_SHORT).show();
        mediaPlayer.stop();
    }}
```

xml file

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:orientation="vertical"
    android:layout_height="match_parent"
    tools:context=".Main4Activity">
    <Button
        android:layout_width="match_parent"
        android:text="Start Service"
        android:id="@+id/ss"
        android:layout_height="wrap_content"/>
    <Button
        android:layout_width="match_parent"
        android:text="Stop Service"
        android:id="@+id/ss1"
        android:layout_height="wrap_content"/>
</LinearLayout>
```

1) Write a program to start a WiFi using services

xml file

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="match_parent"
    tools:context=".Main3Activity"
    android:layout_height="match_parent">
    <Button
        android:id="@+id/button1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="50dp"
        android:text="start Wifi"/>
    <Button
        android:id="@+id/button2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:text="Disabel Wifi"/>
</Linearlayout>
```

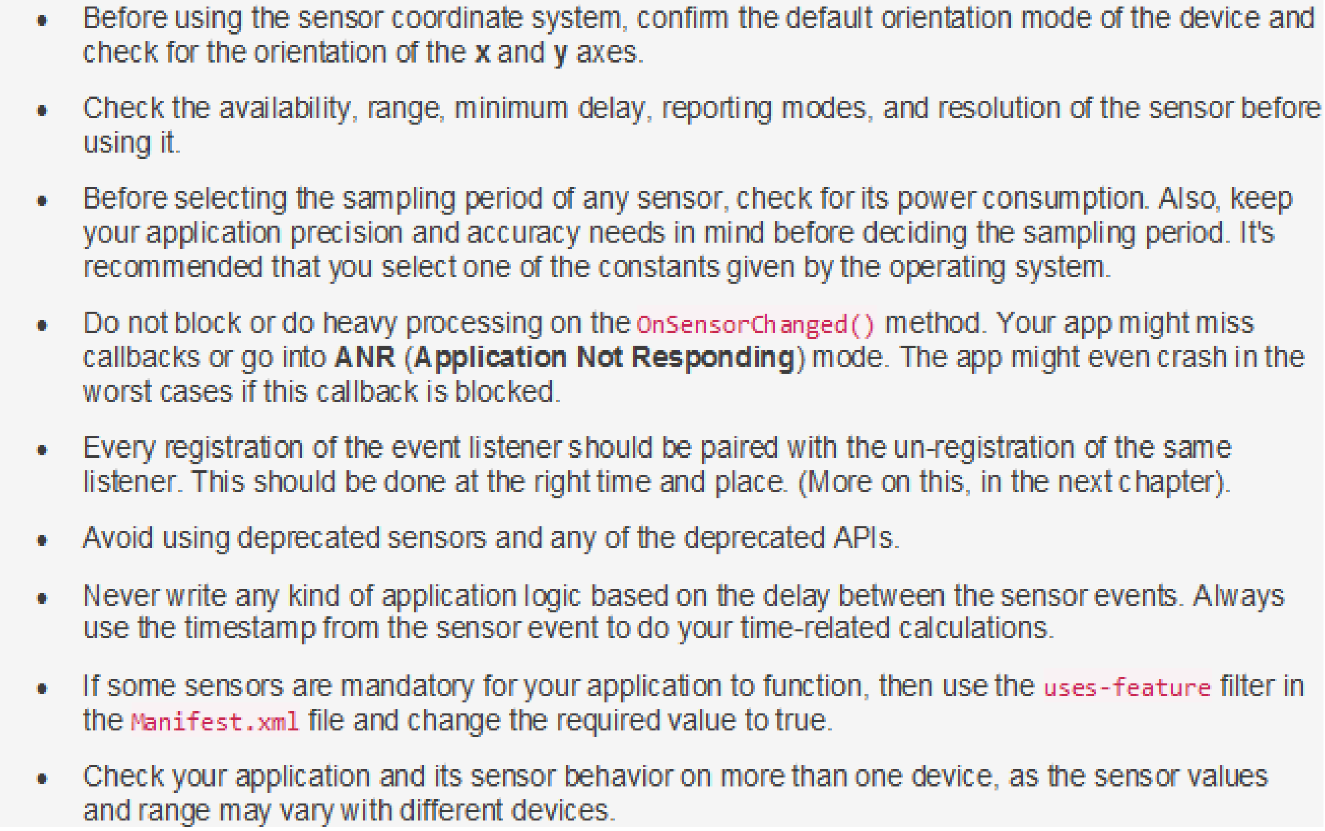
java file

```
package com.example.third;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Context;
import android.content.Intent;
import android.net.wifi.WifiManager;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
```

```
public class Main3Activity extends AppCompatActivity{
Button enableButton,di;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main3);
enableButton=findViewById(R.id.button1);
di=findViewById(R.id.button2);
enableButton.setOnClickListener(new View.OnClickListener(){
public void onClick(View v){
WifiManager
wifi=(WifiManager)getApplicationContext().getSystemService(Context.WIFI_SE
RVICE);
wifi.setWifiEnabled(true); }});
di.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
WifiManager
wifi=(WifiManager)getApplicationContext().getSystemService(Context.WIFI_SE
RVICE);
wifi.setWifiEnabled(false);
}
});}}
```

Practical no-22

1) List the best practices for accessing and using sensors in android.



- Before using the sensor coordinate system, confirm the default orientation mode of the device and check for the orientation of the **x** and **y** axes.
- Check the availability, range, minimum delay, reporting modes, and resolution of the sensor before using it.
- Before selecting the sampling period of any sensor, check for its power consumption. Also, keep your application precision and accuracy needs in mind before deciding the sampling period. It's recommended that you select one of the constants given by the operating system.
- Do not block or do heavy processing on the `OnSensorChanged()` method. Your app might miss callbacks or go into **ANR** (**Application Not Responding**) mode. The app might even crash in the worst cases if this callback is blocked.
- Every registration of the event listener should be paired with the un-registration of the same listener. This should be done at the right time and place. (More on this, in the next chapter).
- Avoid using deprecated sensors and any of the deprecated APIs.
- Never write any kind of application logic based on the delay between the sensor events. Always use the timestamp from the sensor event to do your time-related calculations.
- If some sensors are mandatory for your application to function, then use the `uses-feature` filter in the `Manifest.xml` file and change the required value to true.
- Check your application and its sensor behavior on more than one device, as the sensor values and range may vary with different devices.

2) Write a program to display the list of sensors supported by the mobile device.

xml file

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".Main7Activity">
    <ListView
        android:layout_width="match_parent"
        android:id="@+id/list"
        android:layout_height="match_parent"/>
</LinearLayout>
```

java file

```
package com.example.third;
```

```
import androidx.appcompat.app.AppCompatActivity;
import android.content.Context;
import android.hardware.Sensor;
import android.hardware.SensorManager;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import java.util.List;
public class Main7Activity extends AppCompatActivity
{
    ListView listView;
    SensorManager manager;
    List<Sensor> sensors;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main7);
        listView=findViewById(R.id.list);
        manager=(SensorManager)getSystemService(Context.SENSOR_SERVICE);
        sensors=manager.getSensorList(Sensor.TYPE_ALL);
        listView.setAdapter(new
ArrayAdapter<Sensor>(this,android.R.layout.simple_list_item_1,sensors));
    }
}
```

1)Write a Program to change the background color when device is shuffled

xml file

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">
      <ImageView
        android:layout_width="match_parent"
        android:src="@color/colorPrimaryDark"
        android:layout_height="200dp"
        android:id="@+id/img1">
    </ImageView>
    <ImageView
        android:layout_width="match_parent"
        android:src="@color/colorPrimaryDark"
        android:layout_height="200dp"
        android:id="@+id/img2"
        android:layout_marginTop="30dp">
    </ImageView>
  </LinearLayout>
```

java file

```
package com.example.fifth;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.graphics.Color;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity
{
    ImageView img1,img2;
    private SensorManager sensorManager;
    private Sensor proximitysensor;
    Button b1,b2;
    private SensorEventListener proximitySensorListener;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        img1 = findViewById(R.id.img1);
        img2 = findViewById(R.id.img2);
        b1 = findViewById(R.id.b1);
        b2 = findViewById(R.id.b2);
        b1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new Intent(MainActivity.this,Main2Activity.class);
                startActivity(intent);
            }
        });
        b2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new Intent(MainActivity.this,Main3Activity.class);
                startActivity(intent);
            }
        });
sensorManager=(SensorManager)getSystemService(SENSOR_SERVICE);

proximitysensor=sensorManager.getDefaultSensor(Sensor.TYPE_PROXIMITY);
        if (proximitysensor==null){
            Toast.makeText(this, "Noo", Toast.LENGTH_SHORT).show();
            finish();
        }
        proximitySensorListener=new SensorEventListener() {
            @Override
            public void onSensorChanged(SensorEvent event) {
                if(event.values[0]<proximitysensor.getMaximumRange()){
                    getWindow().getDecorView().setBackgroundColor( Color.RED);
                }
                else {
                    getWindow().getDecorView().setBackgroundColor( Color.GREEN);
                }}
            @Override
            public void onAccuracyChanged(Sensor sensor, int accuracy) {
            } };

sensorManager.registerListener(proximitySensorListener,proximitysensor,2+1000
*1000);
} @Overrideg
    protected void onPause() {
        super.onPause();
        sensorManager.unregisterListener(proximitySensorListener);
    }}
```

Practical No 23:

1) List all the methods related to camera class.
    1. Camera.open()
    2. Camera.startPreview()
    3. Camera.setPreviewDisplay()
    4. Camera.unlock()
    5. MediaRecorder.start()
    6. MediaRecorder.stop()
    7. MediaRecorder.reset()
    8. MediaRecorder.release()
    9. MediaRecorder.prepare()
    10. Camera.lock()
    11. Camera.stopPreview()
    12. Camera.release()

2) Explain the method that is used to detect the face in android.

1. Eye: 1)getIsLeftEyeOPenProbability
2) getIsRightEyeOPenProbability

2. Smiling: 1)getIsSmilingProbability

.

Practical No:-24

Theory Questions:-

1.Name the methods which are used to enable and disable Bluetooth adapter.

1)enable()

2)disable()

2.Explain the purpose of ACTION_REQUEST_DISCOVERABLE Constant.

ACTION_REQUEST_DISCOVERABLE

This constant is used for turn on discovering of Bluetooth.

3.List the uses of setName(String name) method.

The setName() method of thread class is used to change the name of the thread.

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

 <TextView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="Bluetooth" android:textStyle="bold"
  android:layout_marginTop="30dp" android:textSize="30dp"
  android:layout_marginLeft="30dp" />
  android:layout_height="wrap_content"
```

```
   android:text="Turn On" android:layout_marginLeft="30dp"
   android:layout_marginTop="70dp" />
 <Button
   android:id="@+id/btnvisible"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="Get visible"
   android:layout_marginLeft="30dp"
   android:layout_marginTop="110dp" />
 <Button
  android:id="@+id/btnlist"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="List Devices"
  android:layout_marginLeft="30dp"
  android:layout_marginTop="150dp" />
 <Button
 android:id="@+id/btnOFF"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:layout_marginTop="190dp"
 android:layout_marginLeft="30dp"
 android:text="Turn OFF" />
 <ListView
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
android:id="@+id/listView"
```

```
 android:layout_marginTop="40dp"
 android:layout_alignParentBottom="true"
 android:layout_below="@id/btnOFF" />

 </RelativeLayout>
```

Manifest file:-

```
      <?xml version="1.0" encoding="utf-8" ?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="com.example.startservice">
  <uses-permission android:name="android.permission.BLUETOOTH" />
  <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
  <uses-permission
      android:name="android.permission.ACCESS_COARSE_LOCATION" />
 <application android:allowBackup="true" android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:roundIcon="@mipmap/ic_launcher_round" android:supportsRtl="true"
android:theme="@style/AppTheme">
      <activity android:name=".MainActivity">
      <intent-filter>
  <action android:name="android.intent.action.MAIN" />
  <category android:name="android.intent.category.LAUNCHER" />
     </intent-filter>
     </activity>
     </application>
     </manifest>
```

Mainactivity.java

```
package com.example.startservice;
import androidx.appcompat.app.AppCompatActivity;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.widget.Toast;
import java.util.ArrayList;
import java.util.Set;
public class MainActivity extends AppCompatActivity
 {
  private Set<BluetoothDevice> pairedDevices;
    private BluetoothAdapter BA;
    ListView lv;
 @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button btntOn = (Button)findViewById(R.id.btnOn);
        Button btntOff = (Button)findViewById(R.id.btnOFF);
        Button btnlist = (Button)findViewById(R.id.btnlist);
        Button btnget = (Button)findViewById(R.id.btnvisible);
        lv = (ListView)findViewById(R.id.listView);

  final BluetoothAdapter bAdapter = BluetoothAdapter.getDefaultAdapter();
btntOn.setOnClickListener(new View.OnClickListener() {
                public void onClick(View v) {
               if(bAdapter == null)
               {
Toast.makeText(getApplicationContext(),"Bluetooth Not
Supported",Toast.LENGTH_SHORT).show();
               }
               else{
                   if(!bAdapter.isEnabled()){
                       startActivityForResult(new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE),1);
                       Toast.makeText(getApplicationContext(),"Bluetooth Turned
ON",Toast.LENGTH_SHORT).show();
                   }
               }
           }
        });
        btntOff.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View v) {
               bAdapter.disable();
        Toast.makeText(getApplicationContext(),"Bluetooth Turned OFF",
Toast.LENGTH_SHORT).show();
      }
    });
  btnlist.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
pairedDevices = BA.getBondedDevices();

        ArrayList list = new ArrayList();

        for(BluetoothDevice bt : pairedDevices)
          list.add(bt.getName());
        Toast.makeText(getApplicationContext(), "Showing Paired
Devices",Toast.LENGTH_SHORT).show();

        final ArrayAdapter adapter = new
ArrayAdapter(MainActivity.this,android.R.layout.simple_list_item_1, list);

        lv.setAdapter(adapter);
      }
    });



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

```
                Intent getVisible = new
Intent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);
                startActivityForResult(getVisible, 0);
            }
        });
    }
}
```

Practical No:-25

1.write the steps to perform Tween Animation.

Step 1 : Right click res folder and choose new folder option then type the name of the folder in the dialog box appears and press Enter.

Step 2 : Now right click the anim folder and select Android XML file from the options.

Step 3 : A dialog box will appear.

2.Explain the use of from XScale and fromYscale method in detail.

1)fromXScale: Horizontal scaling factor to apply at the start of the animation.

2) YScale: Vertical scaling factor to apply at the start of the animation.

Program Animation

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

<RelativeLayout

    <TextView

       android:layout_width="wrap_content"

       android:layout_height="wrap_content"

       android:text="ANIMATION"

       android:id="@+id/textView"

       android:textSize="35dp"

       android:layout_alignParentTop="true"

       android:layout_centerHorizontal="true" />
```

```
<ImageView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/imageView"
    android:src="@drawable/img"
    android:layout_below="@+id/textView"
    android:layout_alignRight="@+id/textView"
    android:layout_alignEnd="@+id/textView"
    android:layout_alignLeft="@+id/textView"
    android:layout_alignStart="@+id/textView"/>

<Button
    android:layout_width="120dp"
    android:layout_height="wrap_content"
    android:text="zoom"
    android:id="@+id/button"
    android:layout_below="@+id/imageView"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_marginTop="40dp"
    android:onClick="zoom"/>

<Button
    android:layout_width="120dp"
    android:layout_height="wrap_content"
    android:text="clockwise"
    android:id="@+id/button2"
    android:layout_alignTop="@+id/button"
    android:layout_centerHorizontal="true"
    android:onClick="clockwise"/>

<Button
    android:id="@+id/button3"
    android:layout_width="120dp"
    android:layout_height="wrap_content"
    android:layout_alignTop="@+id/button2"
    android:layout_alignParentRight="true"
    android:onClick="fade"
    android:text="fade" />

<Button
    android:layout_width="120dp"
    android:layout_height="wrap_content"
    android:text="blink"
    android:onClick="blink"
    android:id="@+id/button4"
    android:layout_below="@+id/button"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

<Button
    android:layout_width="120dp"
    android:layout_height="wrap_content"
```

```
        android:text="move"
        android:onClick="move"
        android:id="@+id/button5"
        android:layout_below="@+id/button2"
        android:layout_alignRight="@+id/button2"
        android:layout_alignEnd="@+id/button2"
        android:layout_alignLeft="@+id/button2"
        android:layout_alignStart="@+id/button2" />
    <Button
        android:id="@+id/button6"
        android:layout_width="120dp"
        android:layout_height="wrap_content"
        android:layout_below="@+id/button3"
        android:layout_alignLeft="@+id/button3"
        android:layout_marginLeft="2dp"
        android:layout_marginTop="0dp"
        android:layout_toRightOf="@+id/textView"
        android:onClick="slide"
        android:text="slide" />

      <Button
        android:layout_width="120dp"
        android:layout_height="wrap_content"
        android:text="Zoom out"
        android:onClick="ZoomOut"
        android:id="@+id/button7"
        android:layout_below="@+id/button4"
```

```
  />
<Button
   android:layout_width="120dp"
   android:layout_height="wrap_content"
   android:text="fade out"
   android:onClick="slide"
   android:id="@+id/button8"
   android:layout_alignLeft="@+id/button2"
   android:layout_below="@+id/button5"
   android:layout_toRightOf="@+id/button7"
    />
<Button
   android:layout_width="120dp"
   android:layout_height="wrap_content"
   android:text="Rotate"
   android:onClick="rotate"
   android:id="@+id/button9"
   android:layout_alignLeft="@+id/button6"
   android:layout_below="@+id/button6"
   android:layout_toRightOf="@+id/button8"
   />
<Button
   android:layout_width="120dp"
   android:layout_height="wrap_content"
   android:text="Anticlockwise"
   android:onClick="anticlockwise"
```

```
        android:id="@+id/button10"
        android:layout_below="@+id/button7 />

</RelativeLayout>
```

Mainactivity.java

```
package com.example.animation;
import android.os.Bundle;
import android.view.View;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.ImageView;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity
{

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

    public void clockwise(View view)
```

```
{
        ImageView image = (ImageView)findViewById(R.id.imageView);
        Animation animation =
AnimationUtils.loadAnimation(getApplicationContext(),
         R.anim.clockwise);
        image.startAnimation(animation);
   }

    public void zoom(View view)
 {
        ImageView image = (ImageView)findViewById(R.id.imageView);
        Animation animation1 =
AnimationUtils.loadAnimation(getApplicationContext(),
        R.anim.zoom);
        image.startAnimation(animation1);
    }

    public void fade(View view)
{
        ImageView image = (ImageView)findViewById(R.id.imageView);
        Animation animation1 =
AnimationUtils.loadAnimation(getApplicationContext(),
        R.anim.fade);
        image.startAnimation(animation1);
}
```

```
    public void blink(View view)
{
        ImageView image = (ImageView)findViewById(R.id.imageView);
        Animation animation1 =
AnimationUtils.loadAnimation(getApplicationContext(),
         R.anim.blink);
         image.startAnimation(animation1);
    }

    public void move(View view)
{
        ImageView image = (ImageView)findViewById(R.id.imageView);
        Animation animation1 =
AnimationUtils.loadAnimation(getApplicationContext(),
        R.anim.move);
        image.startAnimation(animation1);
    }

    public void slide(View view)
 {
        ImageView image = (ImageView)findViewById(R.id.imageView);
        Animation animation1 =
AnimationUtils.loadAnimation(getApplicationContext(),
        R.anim.slide);
        image.startAnimation(animation1);
    }
    public void ZoomOut(View view){
        ImageView image = (ImageView)findViewById(R.id.imageView);
        Animation animation1 =
AnimationUtils.loadAnimation(getApplicationContext(),
        R.anim.zoomout);
       image.startAnimation(animation1);
}
    public void fadeout(View view)
{
        ImageView image = (ImageView)findViewById(R.id.imageView);
        Animation animation1 =
AnimationUtils.loadAnimation(getApplicationContext(),
        R.anim.fadeout);
        image.startAnimation(animation1);


     }
    public void rotate(View view)
{
        ImageView image = (ImageView)findViewById(R.id.imageView);
        Animation animation1 =
AnimationUtils.loadAnimation(getApplicationContext(),
        R.anim.rotate);
        image.startAnimation(animation1);
 }
    public void anticlockwise(View view)
```

```
{
        ImageView image = (ImageView)findViewById(R.id.imageView);
        Animation animation1 =
AnimationUtils.loadAnimation(getApplicationContext(),
        R.anim.anticlockwise);
        image.startAnimation(animation1);
    }

}
```

res/anim:-

1)Zoom.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
<scale xmlns:android="http://schemas.android.com/apk/res/android"
            android:fromXScale="0.5"
            android:toXScale="3.0"
            android:fromYScale="0.5"
            android:toYScale="3.0"
            android:duration="5000"
            android:pivotX="50%"
            android:pivotY="50%" >
        </scale>
<scale xmlns:android="http://schemas.android.com/apk/res/android"
            android:startOffset="5000"
            android:fromXScale="3.0"
            android:toXScale="0.5"
            android:fromYScale="3.0"
            android:toYScale="0.5"
            android:duration="5000"
            android:pivotX="50%"
            android:pivotY="50%" >
        </scale>
</set>
```

2)zoomout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<scale  xmlns:android="http://schemas.android.com/apk/res/android"
    android:fromXScale="5"
    android:toXScale="1"
    android:fromYScale="5"
    android:toYScale="1"
    android:pivotX="50%"
    android:pivotY="50%"
    android:duration="1000"
    android:fillAfter="true">
</scale>
```

3)clockwise.xml

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

```
<rotate xmlns:android="http://schemas.android.com/apk/res/android"
        android:fromDegrees="0"
        android:toDegrees="360"
        android:pivotX="50%"
        android:pivotY="50%"
        android:duration="5000" >
    </rotate>
 <rotate xmlns:android="http://schemas.android.com/apk/res/android"
        android:startOffset="5000"
        android:fromDegrees="360"
        android:toDegrees="0"
        android:pivotX="50%"
        android:pivotY="50%"
        android:duration="5000" >
    </rotate>
</set>
```

4)anticlockwise

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
android:interpolator="@android:anim/cycle_interpolator">
    <rotate android:fromDegrees="360"
        android:toDegrees="0"
        android:pivotX="50%"
        android:pivotY="50%"
        android:duration="5000" />
  </set>
```

5)Fade.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/accelerate_interpolator" >
 <alpha
        android:fromAlpha="0"
        android:toAlpha="1"
        android:duration="2000" >
    </alpha>
<alpha
        android:startOffset="2000"
        android:fromAlpha="1"
        android:toAlpha="0"
        android:duration="2000"
    </alpha>
</set>
```

6)fadeout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/linear_interpolator">
    <alpha
```

```
        android:duration="2000"
        android:fromAlpha="1.0"
        android:toAlpha="0.1" >
    </alpha>
</set>
```

7)blink.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <alpha android:fromAlpha="0.0"
        android:toAlpha="1.0"
        android:interpolator="@android:anim/accelerate_interpolator"
        android:duration="600"
        android:repeatMode="reverse"
        android:repeatCount="infinite"/>
</set>
```

8)move.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/linear_interpolator"
    android:fillAfter="true">
 <translate
        android:fromXDelta="0%p"
        android:toXDelta="75%p"
        android:duration="800" />
</set>
```

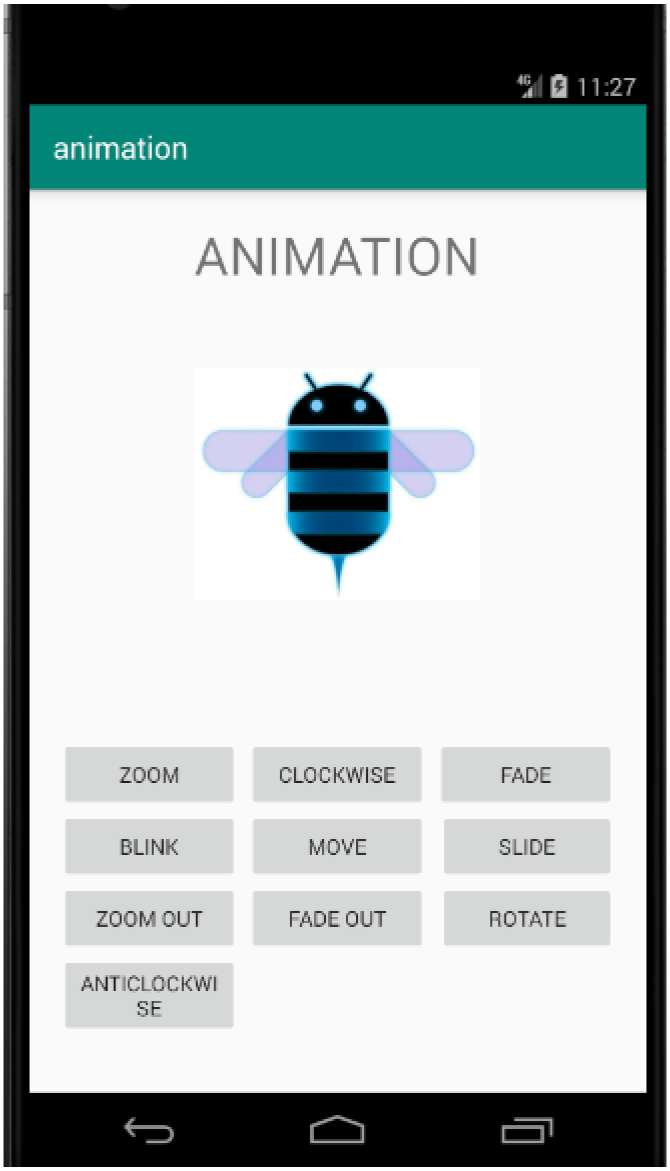
9)rotate

```
<?xml version="1.0" encoding="utf-8"?>
<rotate
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:fromDegrees="0"
    android:toDegrees="360"
    android:pivotX="50%"
    android:pivotY="50%"
    android:repeatCount="infinite"
    android:duration="1200" />
```

10)slide.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:fillAfter="true" >
  <scale
        android:duration="500"
        android:fromXScale="1.0"
        android:fromYScale="1.0"
        android:interpolator="@android:anim/linear_interpolator"
        android:toXScale="1.0"
        android:toYScale="0.0" />
</set>
```

Output



Practical No 26

1. List the basic methods used in an android AsyncTaskclass.

   Ans:

   1. DolonBackground ()

   2. OnPreExecute ()

   3.onPostExecute ()

   4. onProgressUpdate ()

2. Difference between AsyncTask and Services.

Parameter	Service	AsyncTask
1.Trigger	Call to method onstartservice()	Call to method execute ()
2.Trigger from(Thread)	Any thread	Main thread
3.Runs on(Thread)	Main thread	Worker thread
4.Limitations	May block main thread.	One instance can only be execute one.

3. Name the method used, if a process takes a long time to do its work?

DolonBackground ()

This method used in AsyncTask, this is a method or process takes a long time to do its work.

Practical no.27

1) write a program to create the login form and display login successessful/Unsuccessful toast message.

xml file

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".Main12Activity">
    <TextView
        android:layout_width="match_parent"
        android:text="login"
        android:textSize="30dp"
        android:textColor="#000"
        android:textStyle="bold"
        android:gravity="center"
        android:layout_height="wrap_content"/>
    <EditText
        android:layout_width="match_parent"
        android:id="@+id/e1"
        android:layout_marginTop="40dp"
        android:hint="Enter username"
        android:layout_height="wrap_content"/>
    <EditText
        android:layout_width="match_parent"
        android:id="@+id/e2"
        android:hint="Enter username"
        android:layout_height="wrap_content"/>
    <Button
        android:layout_width="wrap_content"
        android:id="@+id/b"
        android:text="login"
        android:layout_gravity="center"
        android:layout_height="wrap_content"/>
</LinearLayout>
```

java file

```
package com.example.third;
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 Main12Activity extends AppCompatActivity
{
    EditText e1,e2;
    Button b;
 @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main12);
        e1=findViewById(R.id.e1);
        e2=findViewById(R.id.e2);
        b=findViewById(R.id.b);
        b.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (e1.getText().toString().equals("")&&
e2.getText().toString().equals(""))
                {
                    Toast.makeText(Main12Activity.this, "Login Unsuccessful",
Toast.LENGTH_SHORT).show();
                }
                else {
                    Toast.makeText(Main12Activity.this, "Login Success",
Toast.LENGTH_SHORT).show();
  }

            }
        });
    }
}
```

Practical No-28

Theory:

1.Explain validation of user input ?

Ans:Input validation climinaty the errors that can be done by user while giving inputs to our app.For example if we want to get the user emails we can check the entered email is a valid email or not before storing it inside the database.

2.List and explain various GUI componentes used to design the login form with validation.

Ans:1)Edit Text View, 2)Text View, 3)Buttons

3)Difference between Text View and Edit Text View.

Ans: EditText is used for user input.TextView is the widget used when you want the user to View the Text (such as label, etc) and EditText used when you want the user to be able to edit the text. The text in either widget can be set programmatically or via xml using the android:text parameter

```
<TextView
android:id="@+id/text_view_id"
android:layout_height="wrap_content"
android:layout_width="wrap_content"
android:text="hello" />
<EditText
android:id="@+id/editText_id"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:inputType="textPersonName"
 android:text="" />
```

Program:

1. Write a program to create the login form with neccessary validations like length of uesrname and password, empt text fields, count of unsuccessful login attempts. Display the login successful/Unsuccessful toastmessage.

Ans:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/LinearLayout1"
```

```
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical"
   tools:context="com.androidcoding.abhi.simple_login.MainActivity" >
<TextView
       android:id="@+id/textView"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="Login"
       android:layout_marginTop="80dp"
       android:layout_gravity="center"
       android:textAppearance="?android:attr/textAppearanceLarge" />
 <LinearLayout
     android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:orientation="vertical"
       android:layout_marginTop="50dp"
       android:background="#9DA7D0"
<EditText
       android:id="@+id/uname"
       android:layout_width="250dp"
       android:layout_height="40dp"
       android:layout_gravity="center"
       android:layout_marginTop="40dp"
       android:background="#ffffff"
       android:hint="username..."/ >
<EditText
```

```
      android:id="@+id/pwd"
      android:layout_width="250dp"
      android:layout_height="40dp"
      android:layout_gravity="center"
      android:layout_marginTop="10dp"
      android:layout_marginBottom="20dp"
      android:inputType="textPassword"
      android:background="#ffffff"
      android:hint="password..."/ >
<LinearLayout
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:orientation="horizontal"
         android:layout_marginTop="10dp"
         android:layout_marginBottom="20dp"
         android:gravity="center" >
<Button
         android:id="@+id/loginbtn"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="Login" />
 <Button
         android:id="@+id/clrbtn"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="Clear" />
      </LinearLayout>
 <TextView
            android:id="@+id/textView1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textColor="#ffffff"
            android:textSize="20dp"
            android:text=" Username = androidcoding \n  Password = androidcoding"/>
</LinearLayout>
</LinearLayout>
```

Java File:

```
public class MainActivity extends Activity {
Button loginbtn, clrbtn;
EditText uname, pwd;
String getuname, getpwd;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
loginbtn = (Button)findViewById(R.id.loginbtn);
clrbtn = (Button)findViewById(R.id.clrbtn);
uname = (EditText)findViewById(R.id.uname);
pwd = (EditText)findViewById(R.id.pwd);
loginbtn.setOnClickListener(new OnClickListener() {
public void onClick(View v) {
getuname = uname.getText().toString();
 getpwd = pwd.getText().toString();
 if(getuname.matches("androidcoding") && getpwd.matches("androidcoding")){
 Intent i = new Intent(MainActivity.this,HomeScreen.class);
            startActivity(i);
            } else {
Toast.makeText(MainActivity.this,
            "enter correct username & password", Toast.LENGTH_LONG).show();
}
}
 });

clrbtn.setOnClickListener(new OnClickListener() {
 public void onClick(View v) {

            uname.setText("");
            pwd.setText("");
 }
  });

}
```

Practical No-29

Theory:

1.Explain the use of SmsManagerClass.

Ans:SmsManager that supports both GSM and CDMA. Manages SMS operations such as sending data, text, and pdu SMS messages. Get this object by calling the static method SmsManager. getDefault()

2.List change that are need to be done in AndroidManifest.XML file to send and receive masseges.

Ans:Use an implicit Intent to launch a messaging app with the ACTION_SENDTO intent action. This is the simplest choice for sending messages. The user can add a picture or other attachment in the messaging app, if the messaging app supports adding attachments. Your app doesn't need code to request permission from the user .If the user has multiple SMS messaging apps installed on the Android phone, the App chooser will appear with a list of these apps, and the user can choose which one to use. (Android smartphones will have at least one, such as Messenger.) The user can change the message in the messaging app before sending it. The user navigates back to your app using the Back button. Send the SMS message using the sendTextMessage() method or other methods of the SmsManager class. This is a good choice for sending messages from your app without having to use another installed app. Your app must ask the user for permission before sending the SMS message, if the user hasn't already granted permission. The user stays in your app during and after sending the message. You can manage SMS operations such as dividing a message into fragments, sending a multipart message, get carrier-dependent configuration values, and so on.To receive SMS messages, use the onReceive() method of the BroadcastReceiver class.

Program:

1.Write a program to send and receive SMS, make use of following GUI

AndroidManifest.xml:

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



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

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

</manifest>
```

Activity_main.xml:

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

<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/colorPrimary"
    tools:context=".MainActivity">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">


    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
```

```
      android:layout_height="wrap_content"
      android:text="Sending SMS Example"
      android:layout_alignParentTop="true"
      android:layout_centerHorizontal="true"
      android:textSize="30dp" />

   <TextView
      android:id="@+id/textView2"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Tutorials point "
      android:textColor="#ff87ff09"
      android:textSize="30dp"
      android:layout_below="@+id/textView1"
      android:layout_alignRight="@+id/imageButton"
      android:layout_alignEnd="@+id/imageButton" />

   <ImageButton
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:id="@+id/imageButton"
      android:src="@drawable/ic_launcher_background"
      android:layout_below="@+id/textView2"
      android:layout_centerHorizontal="true" />

   <EditText
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:id="@+id/editText"
      android:hint="Enter Phone Number"
      android:phoneNumber="true"
      android:textColorHint="@color/abc_primary_text_material_dark"
      android:layout_below="@+id/imageButton"
      android:layout_centerHorizontal="true" />

   <EditText
      android:id="@+id/editText2"
      android:layout_width="182dp"
      android:layout_height="wrap_content"
      android:layout_below="@+id/editText"
```

```
        android:layout_alignStart="@+id/editText"
        android:layout_alignLeft="@+id/editText"
        android:layout_alignEnd="@+id/imageButton"
        android:layout_alignRight="@+id/imageButton"
        android:layout_marginEnd="-30dp"
        android:layout_marginRight="-30dp"
        android:hint="Enter SMS"
        android:textColorHint="@color/abc_primary_text_material_dark" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Send Sms"
        android:id="@+id/btnSendSMS"
        android:layout_below="@+id/editText2"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="48dp" />
        </RelativeLayout>
    </ScrollView>


</RelativeLayout>
```

MainActivity.java

```
package com.example.sms_send;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.Manifest;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
   private static final int MY_PERMISSIONS_REQUEST_SEND_SMS =0 ;
   Button sendBtn;
   EditText txtphoneNo;
   EditText txtMessage;
   String phoneNo;
   String message;

 @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate (savedInstanceState);
      setContentView (R.layout.activity_main);
      sendBtn = (Button) findViewById(R.id.btnSendSMS);
      txtphoneNo = (EditText) findViewById(R.id.editText);
      txtMessage = (EditText) findViewById(R.id.editText2);

      sendBtn.setOnClickListener(new View.OnClickListener() {
         public void onClick(View view) {
            sendSMSMessage();
         }});
}
   private void sendSMSMessage() {

      phoneNo = txtphoneNo.getText().toString();
      message = txtMessage.getText().toString();

      if (ContextCompat.checkSelfPermission(this,
            Manifest.permission.SEND_SMS)
            != PackageManager.PERMISSION_GRANTED) {
         if (ActivityCompat.shouldShowRequestPermissionRationale(this,
               Manifest.permission.SEND_SMS)) {
         } else {
                ActivityCompat.requestPermissions(this,
                        new String[]{Manifest.permission.SEND_SMS},
                        MY_PERMISSIONS_REQUEST_SEND_SMS);
            }
        }

    }
    @Override
    public void onRequestPermissionsResult(int requestCode,String permissions[],
int[] grantResults) {
        while (MY_PERMISSIONS_REQUEST_SEND_SMS==0) {

                if (grantResults.length > 0
                        && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                    SmsManager smsManager = SmsManager.getDefault();
                    smsManager.sendTextMessage(phoneNo, null, message, null, null);
                    Toast.makeText(getApplicationContext(), "SMS sent.",
                            Toast.LENGTH_LONG).show();
                } else {
                    Toast.makeText(getApplicationContext(),
                            "SMS faild, please try again.", Toast.LENGTH_LONG).show();
                    return;
                }

        }

    }

}
```

1)Write a program to send and receive SMS, make use of following GUI

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

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

```
   package="com.example.sms_send">

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

<application

      android:allowBackup="true"

      android:icon="@mipmap/ic_launcher"

      android:label="@string/app_name"

      android:roundIcon="@mipmap/ic_launcher_round"

      android:supportsRtl="true"

      android:theme="@style/AppTheme">

      <activity android:name=".MainActivity">

         <intent-filter>

            <action android:name="android.intent.action.MAIN" />



            <category android:name="android.intent.category.LAUNCHER" />

         </intent-filter>

      </activity>

   </application>

</manifest>
```

Activity_main.xml:

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

```
<RelativeLayout

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

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

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

   android:layout_width="match_parent"

   android:layout_height="match_parent"

   android:background="@color/colorPrimary"

   tools:context=".MainActivity">
  <ScrollView

      android:layout_width="match_parent"

      android:layout_height="wrap_content">


      <RelativeLayout

         android:layout_width="match_parent"

         android:layout_height="wrap_content">




   <TextView

      android:id="@+id/textView1"

      android:layout_width="wrap_content"

      android:layout_height="wrap_content"
```

```
    android:text="Sending SMS Example"

    android:layout_alignParentTop="true"

    android:layout_centerHorizontal="true"

    android:textSize="30dp" />


<TextView

    android:id="@+id/textView2"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:text="Tutorials point "

    android:textColor="#ff87ff09"

    android:textSize="30dp"

    android:layout_below="@+id/textView1"

    android:layout_alignRight="@+id/imageButton"

    android:layout_alignEnd="@+id/imageButton" />


<ImageButton

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:id="@+id/imageButton"
    android:src="@drawable/ic_launcher_background"

    android:layout_below="@+id/textView2"

    android:layout_centerHorizontal="true" />


<EditText

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:id="@+id/editText"

    android:hint="Enter Phone Number"

    android:phoneNumber="true"

    android:textColorHint="@color/abc_primary_text_material_dark"

    android:layout_below="@+id/imageButton"

    android:layout_centerHorizontal="true" />


<EditText

    android:id="@+id/editText2"

    android:layout_width="182dp"

    android:layout_height="wrap_content"

    android:layout_below="@+id/editText"

    android:layout_alignStart="@+id/editText"
```

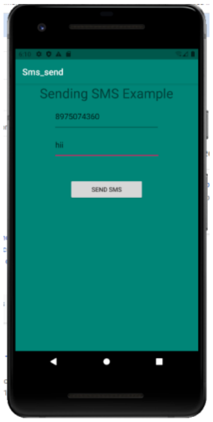
```
        android:layout_alignLeft="@+id/editText"

        android:layout_alignEnd="@+id/imageButton"

        android:layout_alignRight="@+id/imageButton"

        android:layout_marginEnd="-30dp"

        android:layout_marginRight="-30dp"

        android:hint="Enter SMS"

        android:textColorHint="@color/abc_primary_text_material_dark" />


    <Button

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:text="Send Sms"

        android:id="@+id/btnSendSMS"

        android:layout_below="@+id/editText2"

        android:layout_centerHorizontal="true"

        android:layout_marginTop="48dp" />

        </RelativeLayout>

    </ScrollView>
</RelativeLayout>
```

MainActivity.java

```
package com.example.sms_send;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import android.Manifest;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
   private static final int MY_PERMISSIONS_REQUEST_SEND_SMS =0 ;
   Button sendBtn;
   EditText txtphoneNo;
   EditText txtMessage;
   String phoneNo;
   String message;
@Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate (savedInstanceState);
      setContentView (R.layout.activity_main);
      sendBtn = (Button) findViewById(R.id.btnSendSMS);
      txtphoneNo = (EditText) findViewById(R.id.editText);
      txtMessage = (EditText) findViewById(R.id.editText2);

      sendBtn.setOnClickListener(new View.OnClickListener() {
         public void onClick(View view) {
            sendSMSMessage();
         }
});
 }
   private void sendSMSMessage() {

      phoneNo = txtphoneNo.getText().toString();
      message = txtMessage.getText().toString();
```

```
        if (ContextCompat.checkSelfPermission(this,
                Manifest.permission.SEND_SMS)
                != PackageManager.PERMISSION_GRANTED) {
            if (ActivityCompat.shouldShowRequestPermissionRationale(this,
                    Manifest.permission.SEND_SMS)) {
            } else {
                ActivityCompat.requestPermissions(this,
                        new String[]{Manifest.permission.SEND_SMS},
                        MY_PERMISSIONS_REQUEST_SEND_SMS);
            }
        }

    }
    @Override
    public void onRequestPermissionsResult(int requestCode,String permissions[],
int[] grantResults) {
        while (MY_PERMISSIONS_REQUEST_SEND_SMS==0) {

                if (grantResults.length > 0
                        && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                    SmsManager smsManager = SmsManager.getDefault();
                    smsManager.sendTextMessage(phoneNo, null, message, null, null);
                    Toast.makeText(getApplicationContext(), "SMS sent.",
                            Toast.LENGTH_LONG).show();
                } else {
                    Toast.makeText(getApplicationContext(),
                            "SMS faild, please try again.", Toast.LENGTH_LONG).show();
                    return;
                }

        }

    }

}
```

Output:



6:10

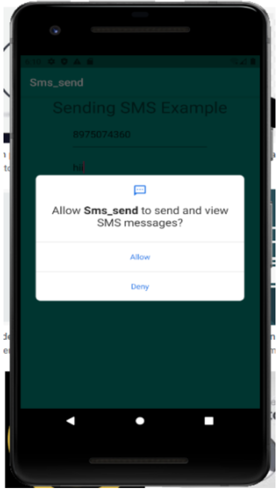
Sms_send

Sending SMS Example

8975074360

hii

Fig:GUI of Sending Message



Sms_send

Sending SMS Example

8975074360

hii

Allow **Sms_send** to send and view SMS messages?

Fig:Permission for Sending Message

Practical no. 30

Practical related Q:-

1. Why it becomes neccessary to have inbuilt email module in mobile applications.

Ans:-Email is messages distributed by electronic means from one system user to one or more recipients via a network.

Before starting Email Activity, You must know Email functionality with intent, Intent is carrying data from one component to another component with-in the application or outside the application.

To send an email from your application, you don’t have to implement an email client from the beginning, but you can use an existing one like the default Email app provided from Android, Gmail, Outlook, K-9 Mail etc. For this purpose, we need to write an Activity that launches an email client, using an implicit Intent with the right action and data. In this example, we are going to send an email from our app by using an Intent object that launches existing email clients.

Following section explains different parts of our Intent object required to send an email.

Intent Object - Action to send Email

You will use ACTION_SEND action to launch an email client installed on your Android device. Following is simple syntax to create an intent with ACTION_SEND action.

Intent emailIntent = new Intent(Intent.ACTION_SEND);

Intent Object - Data/Type to send Email

To send an email you need to specify mailto: as URI using setData() method and data type will be to text/plain using setType() method as follows −

emailIntent.setData(Uri.parse("mailto:"));

emailIntent.setType("text/plain");

2.List the extra fields that can be used in an application to send emails.

Ans:=

1 EXTRA_BCC

A String[] holding e-mail addresses that should be blind carbon copied.

2 EXTRA_CC

A String[] holding e-mail addresses that should be carbon copied.
3EXTRA_EMAIL
A String[] holding e-mail addresses that should be delivered to.
4 EXTRA_HTML_TEXT
A constant String that is associated with the Intent, used with ACTION_SEND to supply an alternative to
EXTRA_TEXT as HTML formatted text.
5 EXTRA_SUBJECT
A constant string holding the desired subject line of a message.
6 EXTRA_TEXT
A constant CharSequence that is associated with the Intent, used with ACTION_SEND to supply the
literal data to be sent.
7 EXTRA_TITLE
A CharSequence dialog title to provide to the user when used with a ACTION_CHOOSER.

Exercise
1.Write a program to send email.
Ans:=
Xml File:-

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

   <LinearLayout
   tools:context=".MainActivity"
   android:orientation="vertical"
   android:layout_height="match_parent"
   android:layout_width="match_parent"
   xmlns:tools="http://schemas.android.com/tools"
xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:android="http://schemas.android.com/apk/res/android">

   <EditText
      android:layout_height="wrap_content"
      android:layout_width="match_parent"
```

```
        tools:ignore="InvalidId"
        android:layout_marginTop="20dp"
        android:gravity="center"
        android:textSize="35dp"
        android:hint="Email Id"
        android:id="@+id/etmail"/>

    <EditText
        android:layout_height="wrap_content"
        android:layout_width="match_parent"
        tools:ignore="InvalidId"
        android:layout_marginTop="20dp"
        android:gravity="center"
        android:textSize="35dp"
        android:hint="Subject"
        android:id="@+id/etsubject"/>

    <EditText android:layout_height="200dp"
        android:layout_width="match_parent"
        tools:ignore="InvalidId"
        android:layout_marginTop="20dp"
        android:gravity="center"
        android:textSize="35dp"
        android:hint="Message"
        android:id="@+id/etmsg"/>

    <Button
        android:layout_height="wrap_content"
        android:layout_width="match_parent"
        android:textSize="30dp"
        android:id="@+id/btnsend"
        android:text="Send"/>

</LinearLayout>
```

Java File:-

```
package com.example.gmail;

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;

public class MainActivity extends AppCompatActivity {
    EditText et1,et2,et3;
    Button btn1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        et1=findViewById(R.id.etmail);
        et2=findViewById(R.id.etsubject);
        et3=findViewById(R.id.etmsg);
        btn1=findViewById(R.id.btnsend);
        btn1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                String id=et1.getText().toString();
                String subject=et2.getText().toString();
                String message=et3.getText().toString();
                Intent mail=new Intent(Intent.ACTION_SEND);
                mail.putExtra(Intent.EXTRA_EMAIL,id);
                mail.putExtra(Intent.EXTRA_SUBJECT,subject);
                mail.putExtra(Intent.EXTRA_TEXT,message);
                mail.setType("Message/rfc812");
                startActivity(Intent.createChooser(mail,"Send"));

            }
        });
    }

}
```

Manifest File:-

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

    <dist:module dist:instant="true" />

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

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

</manifest>
```

Paractical No. 31

Practical related Q:-

Q1. List the names of map type and write the syntax to change it.

Ans:- 1)Normal-googleMaps.setMapType(GoogleMap.MapType_Normal);
2)Hybrid-googleMap.setMapType(GoogleMap.MapType_HYBRID);
3)Satelite-googleMap.setMapType(GoogleMap.MapType_SATELITE);
4)Terrain-googleMap.setMapType(GoogleMap.MapType_TERRAIN);
5)None

Q2.Name the methods used to enable and disable zoom feature.

1. public void hide ()
2. public void setOnZoomInClickListener (View.OnClickListener listener)
3. public void setOnZoomOutClickListener (View.OnClickListener listener)
4. public void setIsZoomInEnabled (boolean isEnabled)
5. public void setIsZoomOutEnabled (boolean isEnabled)
6. public void show ()

Exercise:-

1. Write a program to locate users current location.

Xml file:-

```
<?xml version="1.0" encoding="utf-8"?>
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:map="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:id="@+id/map"
   android:name="com.google.android.gms.maps.SupportMapFragment"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".MapsActivity" />
```

Java File:-

```
package com.example.map;
```

```
import androidx.fragment.app.FragmentActivity;

import android.os.Bundle;

import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;

public class MapsActivity extends FragmentActivity implements
OnMapReadyCallback {

    private GoogleMap mMap;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        // Obtain the SupportMapFragment and get notified when the map is ready to
be used.
        SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager()
                .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
    }



      @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;

        // Add a marker in Sydney and move the camera
        LatLng sydney = new LatLng(17.6806, 75.3155);
        mMap.addMarker(new MarkerOptions().position(sydney).title("Marker in
Sydney"));
        mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
    }
}
```

Resource File:-

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="map_kay" translatable="false" />
    AIzaSyALMp5hkiAhmlFWfJ-cE6NFjhyF5LH4Adg
</resources>
```

Manifest File:-

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

<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" >
<meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value="@string/google_maps_key" />

        <activity
            android:name=".MapsActivity"
            android:label="@string/title_activity_maps">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```