Mobile Application Development

Regulation – R20

Lab manual for the Academic Year

(2022-2023)

II MCA I Semester

DEPARTMENT OF
INFORMATION TECHNOLOGY

VIGNAN'S
Foundation for Science, Technology & Research
(Deemed to be University)
-Estd. u/s 3 of UGC Act 1956

Vadlamudi, 522 213, AP, India

Name of the Faculty: Ms.T.Gayathri, Assistant Professor, Dept. of IT

Manual Structure

This manual is organized in such a way that the students can directly use it in the laboratory. Each laboratory exercise comprises of

1. Statement of the problem

2. XML file

4. Java file

5. Result

Scheme of evaluation of laboratory

The weight age for continuous in semester performance evaluation shall be 50 marks and the end semester performance shall be 50 marks.

The assessment for in semester performance will be based on the student attendance, regular work in the experiments, performance in viva-voce, result analysis of each experiment, quality of reports submitted regularly and in semester examination.

For in semester examination

Activity	**Marks**
1. Attendance	5
2. Performing experiment	10
3. Analysis of results and interpretation	5
4. Punctual submission of quality records	10
5. Laboratory in semester test	20
Total	50

For End semester examination

Activity	**Marks**
1. Design /Mathematical model	20
2. Implementation	20
3. Viva-voce	10
Total	50

Objective of Laboratory

1. To make students to implement different application programs by using the Activities, Intents, Dialog Objects and Fragments by using the Android.
2. At end of the course the student will be

   - Familiar with understanding and Linking of activities using intents.
   - Familiar with different Layouts such as Linear, absolute, Relative, Frame and Scroll Views.
   - Master analysing problems and writing program solutions to problems using the above design techniques.

Recommended System/ Software Requirements

- Intel based desktop PC with Java
- Android Studio

Experiments List

1. Installation of Android studio, its required tools and Android Virtual Device (Emulator).
2. a) Displaying the welcome message in AVD.
   b) Design an android application to display name, class, college, address etc. of a student.
3. a) Creating a basic Activity and applying themes, styles to it.
   b) Create an application to understand the Life Cycle of an Activity.
   c) Create an android application which display three buttons on main activity named RED, GREEN and BLUE. The background colour of the activity should be changed to appropriate colour when the user selects any of these buttons.
4. a) Displaying various types of Dialog objects.
   b) Develop a simple application showing a dialog box.
5. Linking activities with Intents.
   a) Create an application with Explicit Intents
   b) Create an application of Calling Built-In apps using Intents (Implicit Intents)
   c) Design an android application login form whenever user select login in the another displays the message as "WELCOME TO VIGNAN'S UNIVERSITY"
6. a) Passing data using Intent object.
   b)Develop an application of Basic Calculator app.
7. a) Usage of Fragments and adding them dynamically to the application.
   b) Usage of Fragments and adding them dynamically to the application.
8. a) Communication between fragments.
   b) Develop an application of BMI
9. Creating various layouts.
   a) Linear Layout
   b) Absolute Layout
   c) Table Layout
   d) Relative Layout
   e) Frame Layout
   f) Scroll View
   g) Design an application which contains some buttons named as relative layout, Linear layout, Table Layout…etc. Whenever the user selects the corresponding button it will opens a new activity which are designed according to the name displayed in the button.
10. a) Displaying Action bar.
    b) Design an Action bar which is similar to WHATS APP.
11. a) Handling view events.
    b) Create an application of scoreboard app
12. Create an application by following all the conditions
    - Registration form by including all basic views,
    - Display the dialog box as "Please verify all the details once again"
    - Successfully register, Open the activity is designed with fragments
    - Navigate to another activity present in different application.
    - Develop the action bar for that activity.

EXPERIMENT - 1

AIM: a) Installation of Android studio, its required tools and Android Virtual Device (Emulator).

Android Studio is one of the best and most popular IDE for development of Android application which is released by IntelliJ IDEA. Android studio has rich functionalists and easy to work. Android studio has great and awesome user interface, it is also smart with coding and will save your time.

Android programming is Java based programming so it requires JDK & JRE environment. You must have to install JDK & JRE for working with android. After installing JDK and JRE you need to configure Environment variables on your computer. Environment variables let to provide some specific path for android. There will be JDK/bin or JDK only. After installing JDK & JRE and configuring environment variables, the next step will be installation of SDK (Software Development Kit). This is another important component for android programming. If you are thinking to develop android apps, then you must install SDK on your computer.

Installing Android Studio on Windows

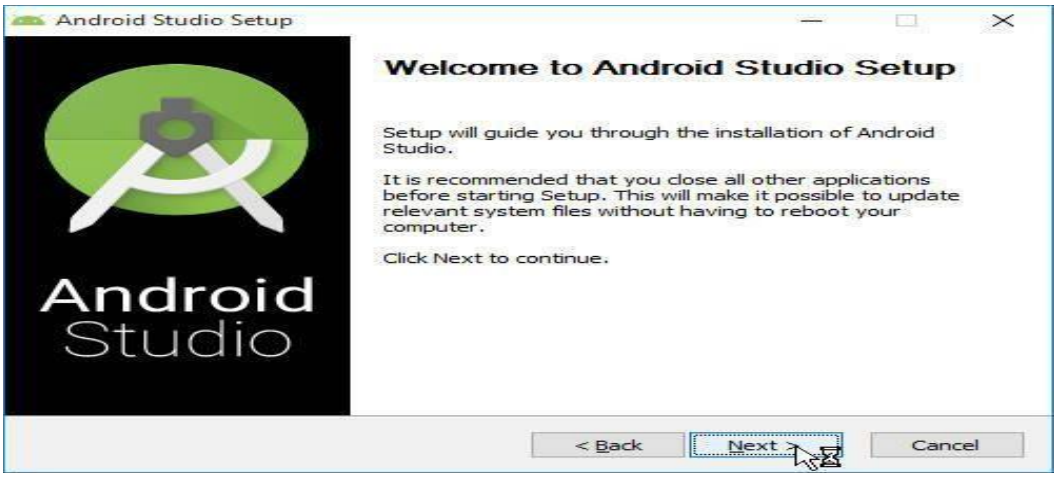
To download Android Studio form the website **https://developer.android.com/studio** There you will see android studio download button.

Android Studio
The official Android IDE
Android Studio IDE
Android SDK tools
Android 6.0 (Marshmallow) Platform
Android 6.0 emulator system image with Google APIs
DOWNLOAD ANDROID STUDIO FOR WINDOWS

After downloading android studio, follow below steps to install android studio on your computer.

Step1: First unzip the downloaded android studio file, open it and wait for few seconds, it will take some time to load on your screen.

Step2: After that the window will appear on your screen, where it will show you welcome to android studio setup. Go for "Next"



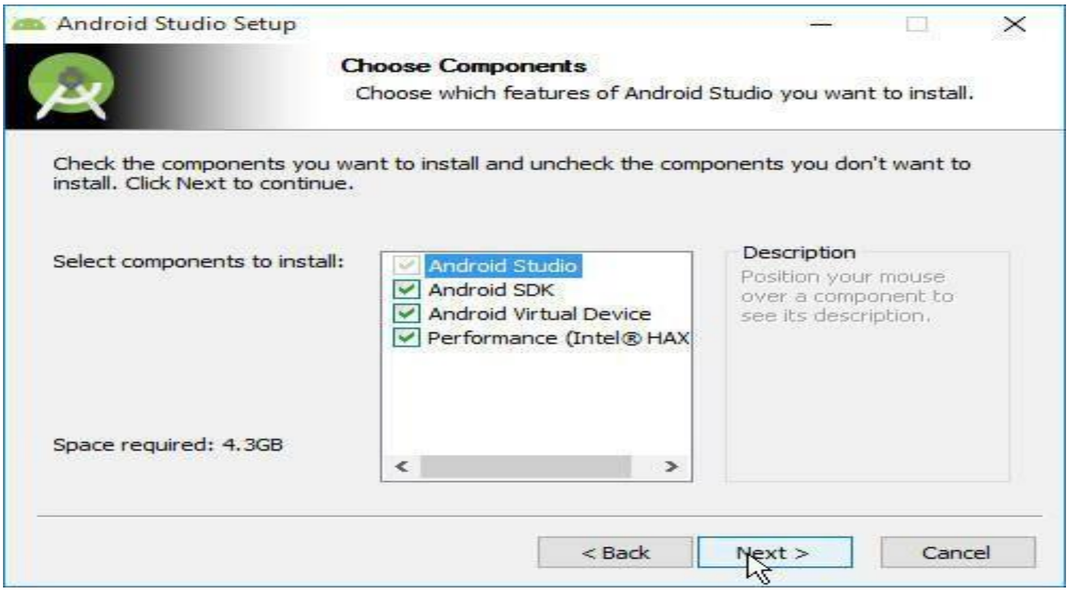
Welcome to Android Studio Setup

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.

Step3: Now choose what you want to install, all are most important component in android programming. Go for "Next"



Choose Components

Choose which features of Android Studio you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

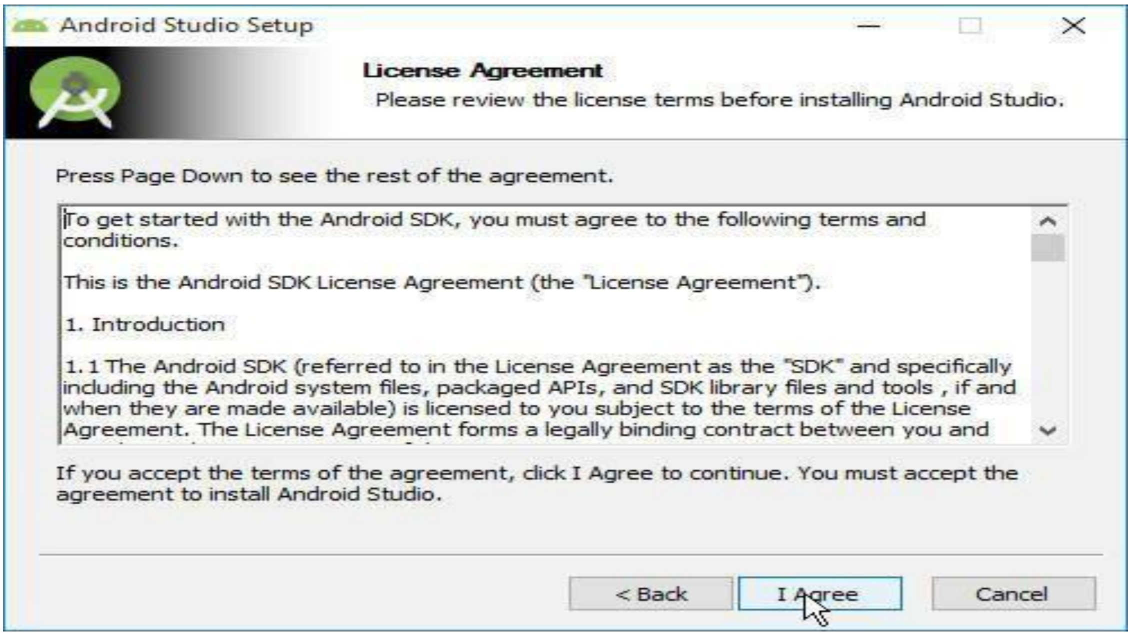
☑ Android Studio
☑ Android SDK
☑ Android Virtual Device
☑ Performance (Intel® HAX

Space required: 4.3GB

Description

Position your mouse over a component to see its description.

Step4: Go for "I Agree", It's for integrating SDK, it will confirm SDK installation.



License Agreement

Please review the license terms before installing Android Studio.

Press Page Down to see the rest of the agreement.

To get started with the Android SDK, you must agree to the following terms and conditions.

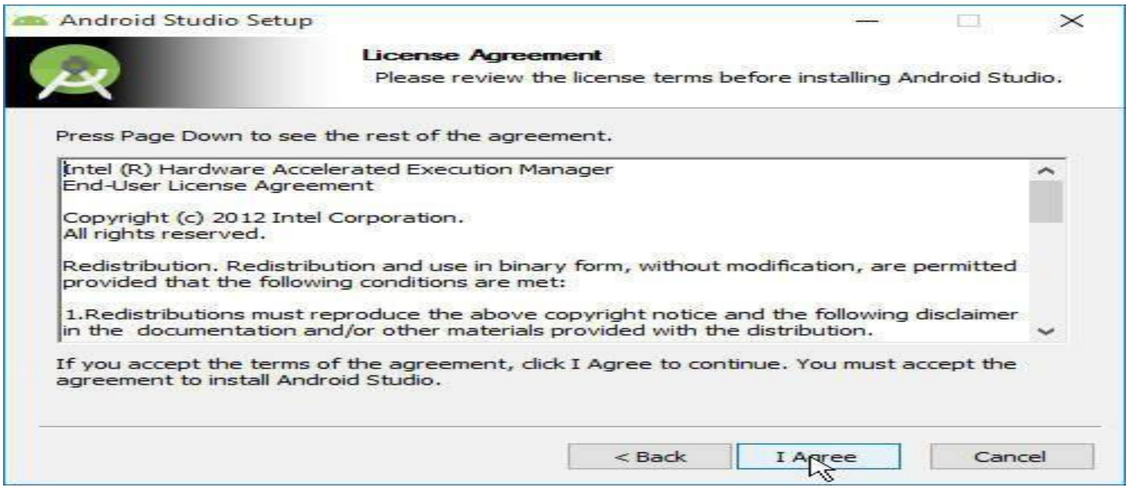
This is the Android SDK License Agreement (the "License Agreement").

1. Introduction

1.1 The Android SDK (referred to in the License Agreement as the "SDK" and specifically including the Android system files, packaged APIs, and SDK library files and tools , if and when they are made available) is licensed to you subject to the terms of the License Agreement. The License Agreement forms a legally binding contract between you and

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Android Studio.

Step5: Go for "I Agree", It's for Haxm installation



License Agreement
Please review the license terms before installing Android Studio.

Press Page Down to see the rest of the agreement.

Intel (R) Hardware Accelerated Execution Manager
End-User License Agreement

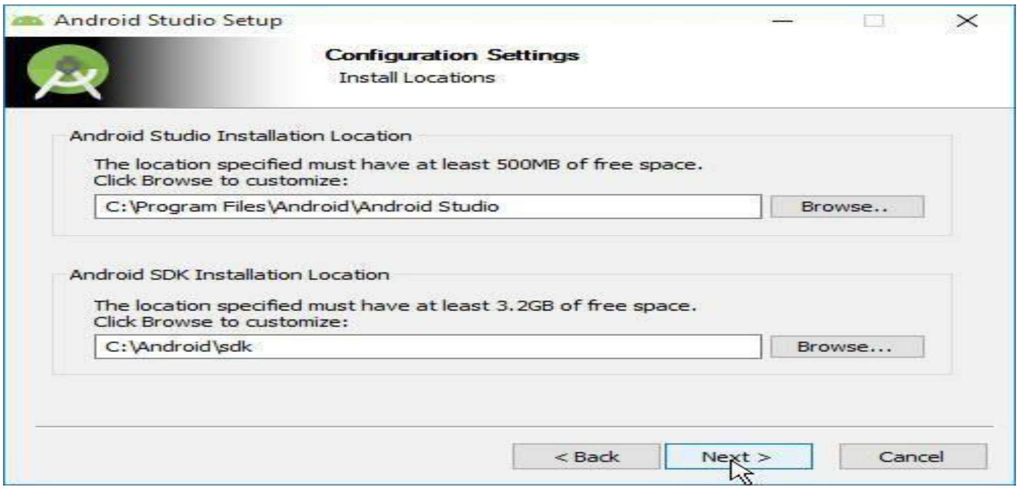
Copyright (c) 2012 Intel Corporation.
All rights reserved.

Redistribution. Redistribution and use in binary form, without modification, are permitted provided that the following conditions are met:

1.Redistributions must reproduce the above copyright notice and the following disclaimer in the documentation and/or other materials provided with the distribution.

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Android Studio.

Step6: Now choose the location for Android Studio and SDK.Go for "Next"



Configuration Settings
Install Locations

Android Studio Installation Location

The location specified must have at least 500MB of free space.
Click Browse to customize:

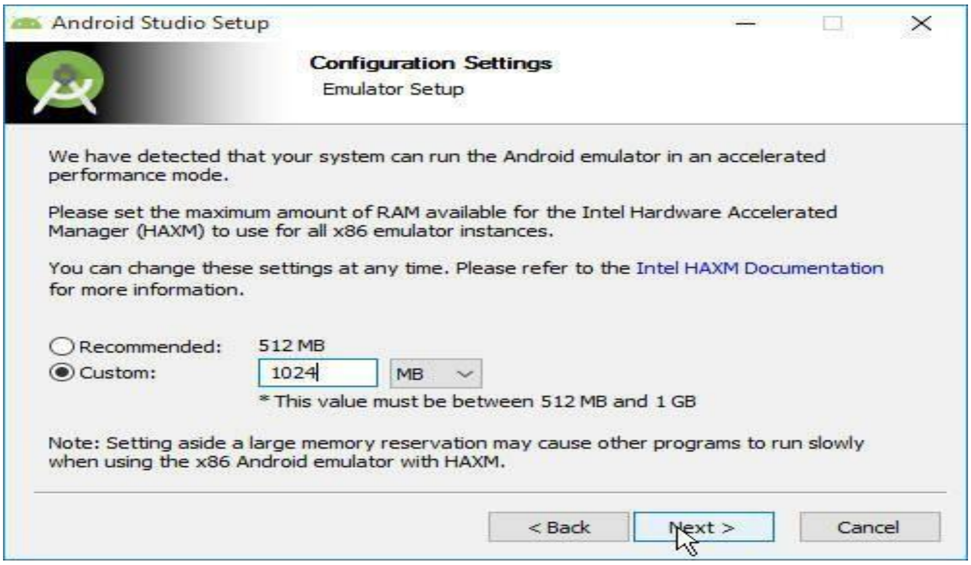
C:\Program Files\Android\Android Studio

Android SDK Installation Location

The location specified must have at least 3.2GB of free space.
Click Browse to customize:

C:\Android\sdk

Step7: Here you need to define the size of android emulator processor. You can define up to 2GB and more than that, depends on your RAM capacity. The size will be determined through Haxm. This will help your android studio to run faster.



Configuration Settings
Emulator Setup

We have detected that your system can run the Android emulator in an accelerated performance mode.

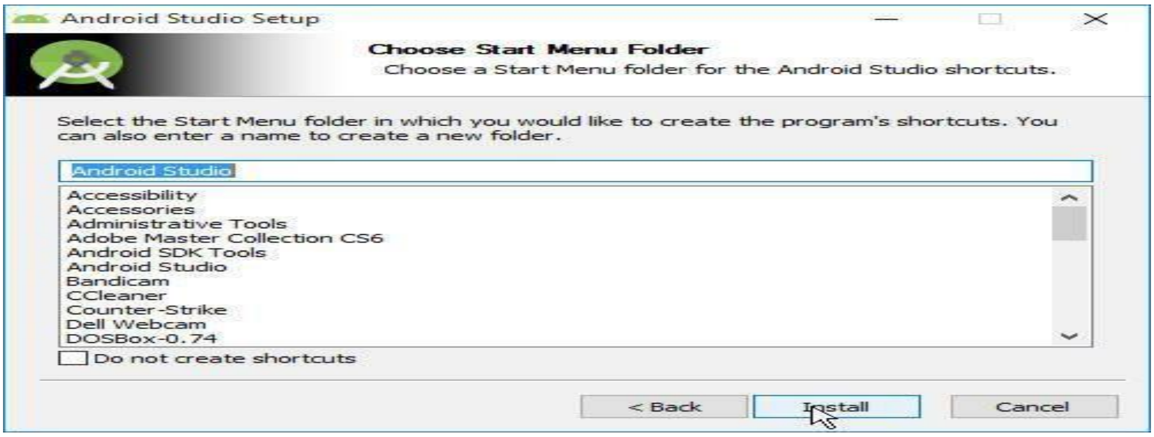
Please set the maximum amount of RAM available for the Intel Hardware Accelerated Manager (HAXM) to use for all x86 emulator instances.

You can change these settings at any time. Please refer to the Intel HAXM Documentation for more information.

○ Recommended: 512 MB
● Custom: 1024 MB
* This value must be between 512 MB and 1 GB

Note: Setting aside a large memory reservation may cause other programs to run slowly when using the x86 Android emulator with HAXM.

Step8: Now choose setting for shortcut of android studio on your computer.


Android Studio Setup

Choose Start Menu Folder

Choose a Start Menu folder for the Android Studio shortcuts.

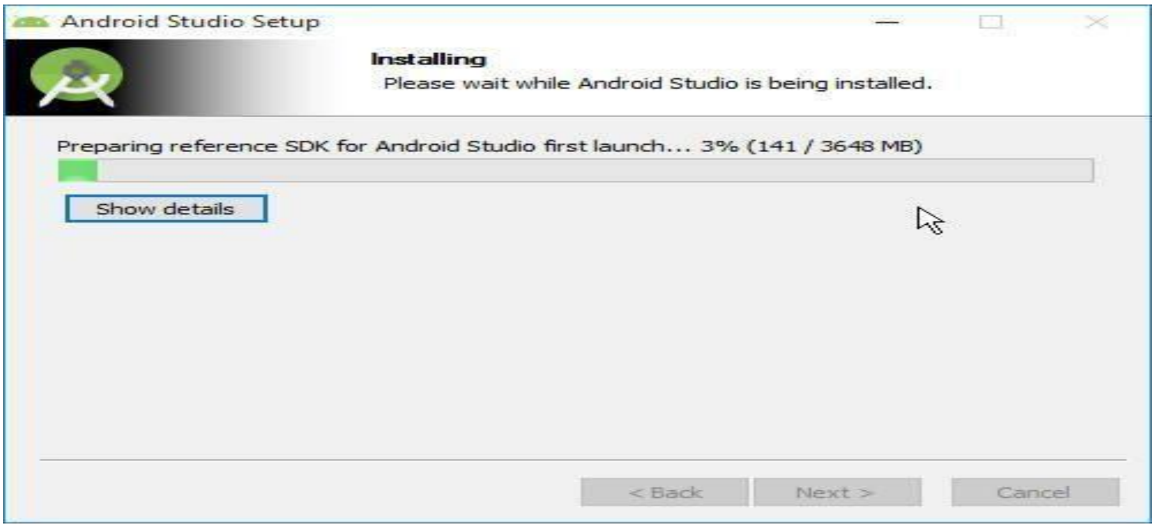
Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.

Android Studio

Accessibility
Accessories
Administrative Tools
Adobe Master Collection CS6
Android SDK Tools
Android Studio
Bandicam
CCleaner
Counter-Strike
Dell Webcam
DOSBox-0.74

□ Do not create shortcuts

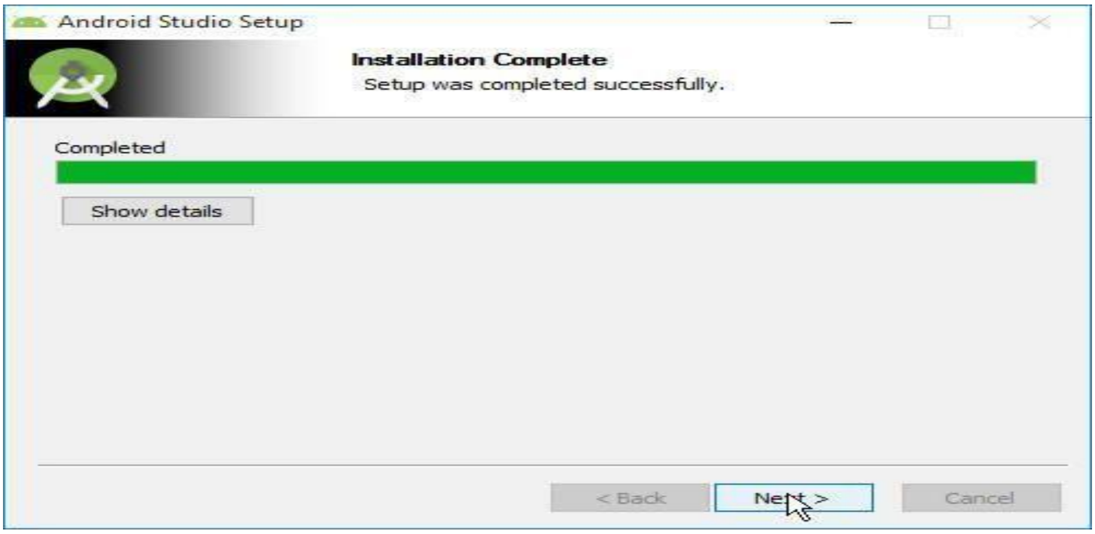
Step9: After that it will take lots of time to copy all the files to the storage location you have selected before


Android Studio Setup

Installing

Please wait while Android Studio is being installed.

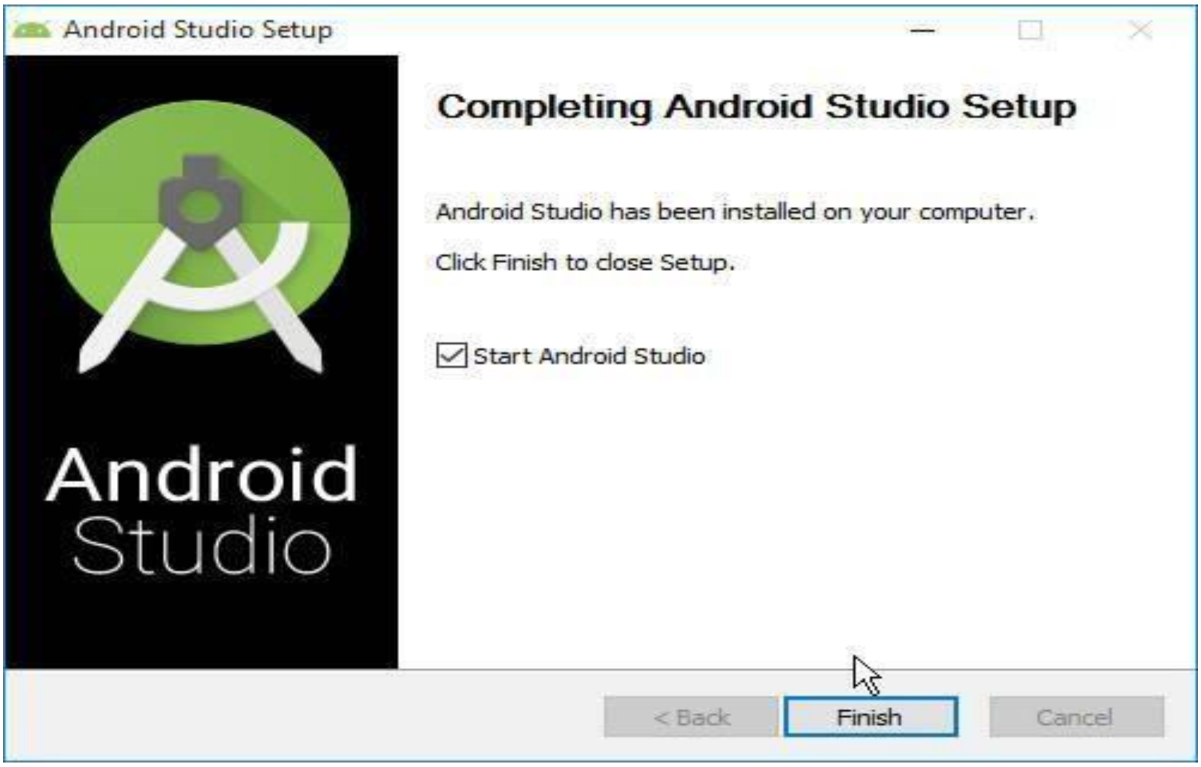
Preparing reference SDK for Android Studio first launch... 3% (141 / 3648 MB)


Android Studio Setup

Installation Complete

Setup was completed successfully.

Completed



Completing Android Studio Setup

Android Studio has been installed on your computer.

Click Finish to close Setup.

☑ Start Android Studio

EXPERIMENT – 2

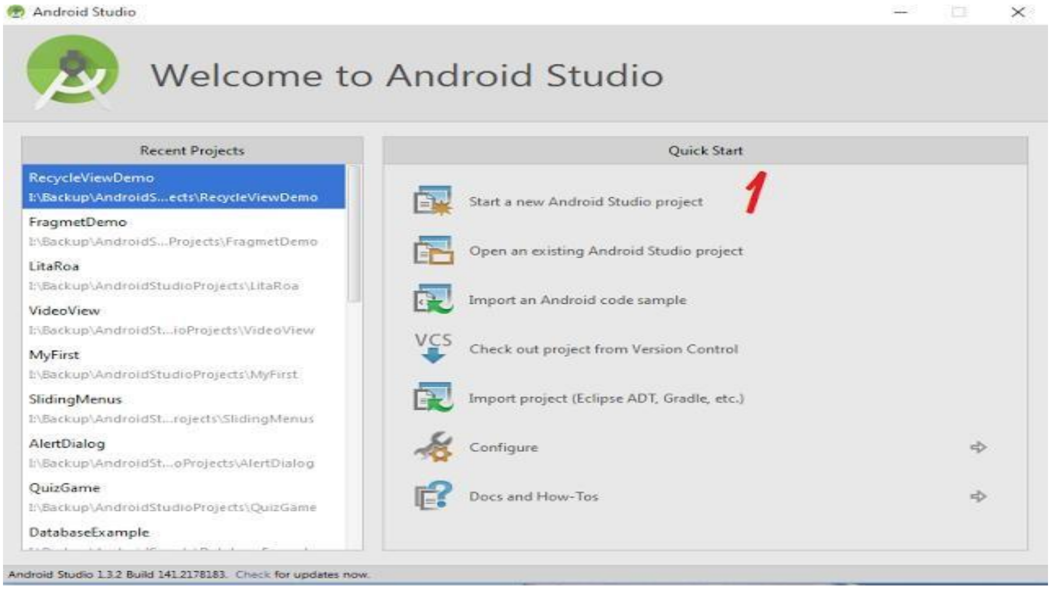
AIM: a) Displaying the welcome message in AVD.

Android Studio is one of the best IDE (Integrated Development Environment) for Android Developers with rich functionality and control mechanism. It was announced on 2013 and today it is one of the most popular IDE in the world for android development purpose.

In this we will learn how to create new project on android studio.

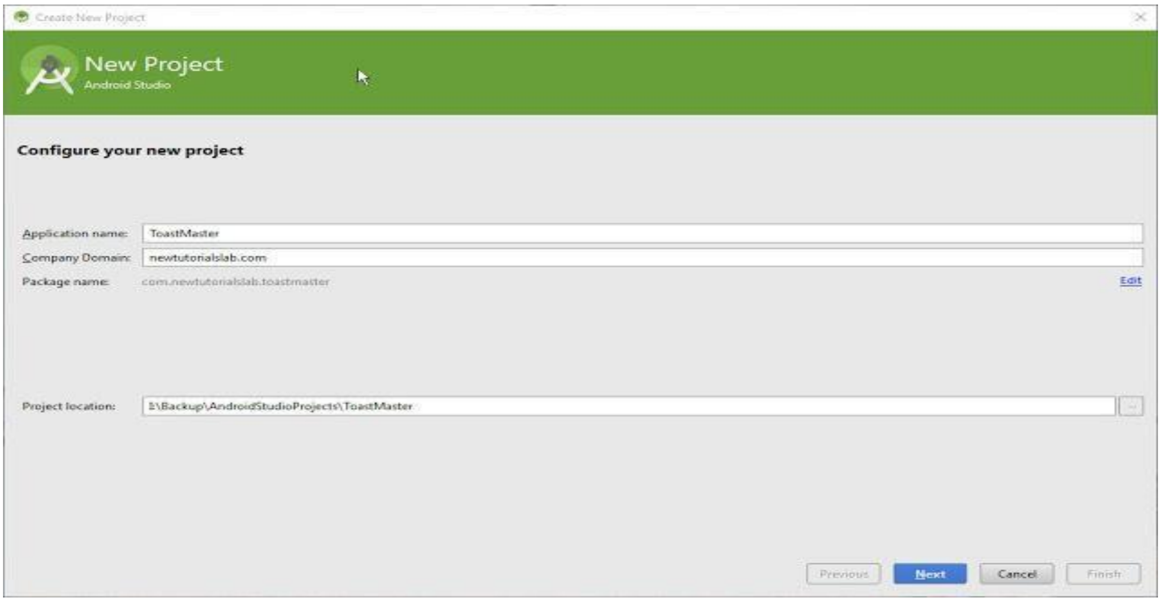
Step1: First of all you need to start your Android Studio. Then it will show you the default start window on your screen, where you can see various options.

In this window you have to click on Start a new Android Studio project, that will let you create new project on android studio.


Android Studio
Welcome to Android Studio
Recent Projects
RecycleViewDemo
I:\Backup\AndroidS...ects\RecycleViewDemo
FragmetDemo
I:\Backup\AndroidS...Projects\FragmetDemo
LitaRoa
I:\Backup\AndroidStudioProjects\LitaRoa
VideoView
I:\Backup\AndroidSt...ioProjects\VideoView
MyFirst
I:\Backup\AndroidStudioProjects\MyFirst
SlidingMenus
I:\Backup\AndroidSt...rojects\SlidingMenus
AlertDialog
I:\Backup\AndroidSt...oProjects\AlertDialog
QuizGame
I:\Backup\AndroidStudioProjects\QuizGame
DatabaseExample
Quick Start
Start a new Android Studio project
1
Open an existing Android Studio project
Import an Android code sample
VCS
Check out project from Version Control
Import project (Eclipse ADT, Gradle, etc.)
Configure
Docs and How-Tos

Android Studio 1.3.2 Build 141.2178183. Check for updates now.

Step2: After clicking on new project will get another window on your screen which will ask for project information or configuration.



Configure your new project

Application name: ToastMaster

Company Domain: newtutorialslab.com

Package name: com.newtutorialslab.toastmaster

Project location: I:\Backup\AndroidStudioProjects\ToastMaster

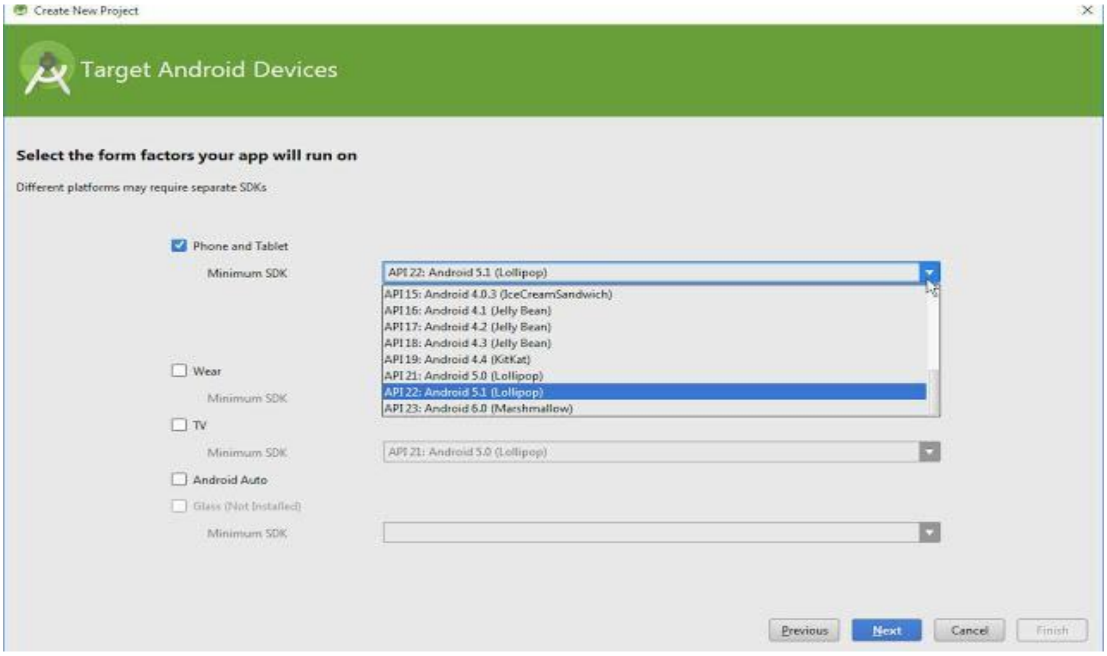
As you can see on the above windows, you need to put Application Name, Application Name: Without Space, Capitalized letter format.

You must write your company name like we have newtutorialslab.com, or ansmachine.net, satyajoshi.com.np etc.

Package Name: Package naming convention is reverse notation as you write company name it will automatically change the package name in the form of reverse notation.

Project Location: You must select appropriate location for your project. That might be default location of C drive or you can select another drive location to save your special project.

Step3: After doing all the above tasks you must have to choose platform where you want to run your android application. Run able factors for your project. You can choose as per your need.



Create New Project

Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

☑ Phone and Tablet

Minimum SDK: API 22: Android 5.1 (Lollipop)

API 15: Android 4.0.3 (IceCreamSandwich)
API 16: Android 4.1 (Jelly Bean)
API 17: Android 4.2 (Jelly Bean)
API 18: Android 4.3 (Jelly Bean)
API 19: Android 4.4 (KitKat)
API 21: Android 5.0 (Lollipop)
API 22: Android 5.1 (Lollipop)
API 23: Android 6.0 (Marshmallow)

□ Wear

Minimum SDK

□ TV

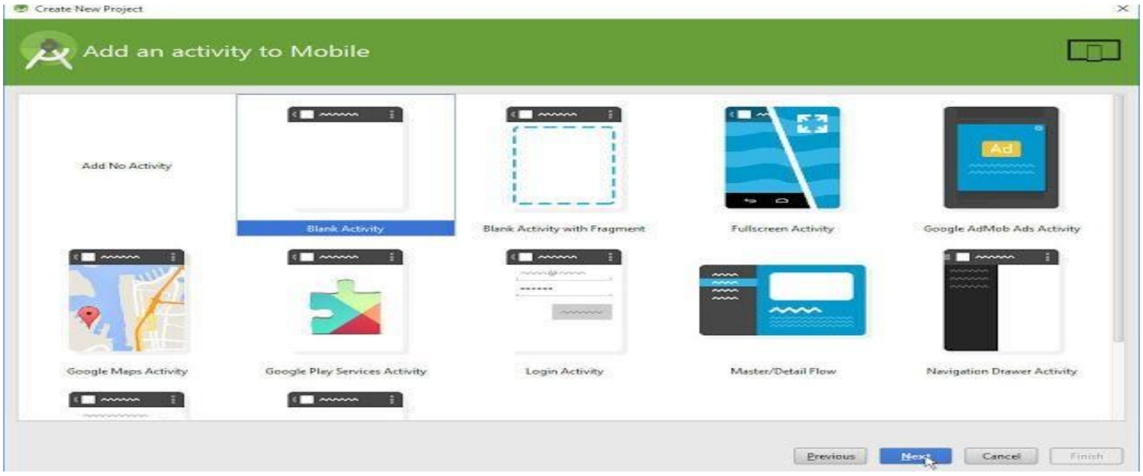
Minimum SDK: API 21: Android 5.0 (Lollipop)

□ Android Auto

□ Glass (Not Installed)

Minimum SDK

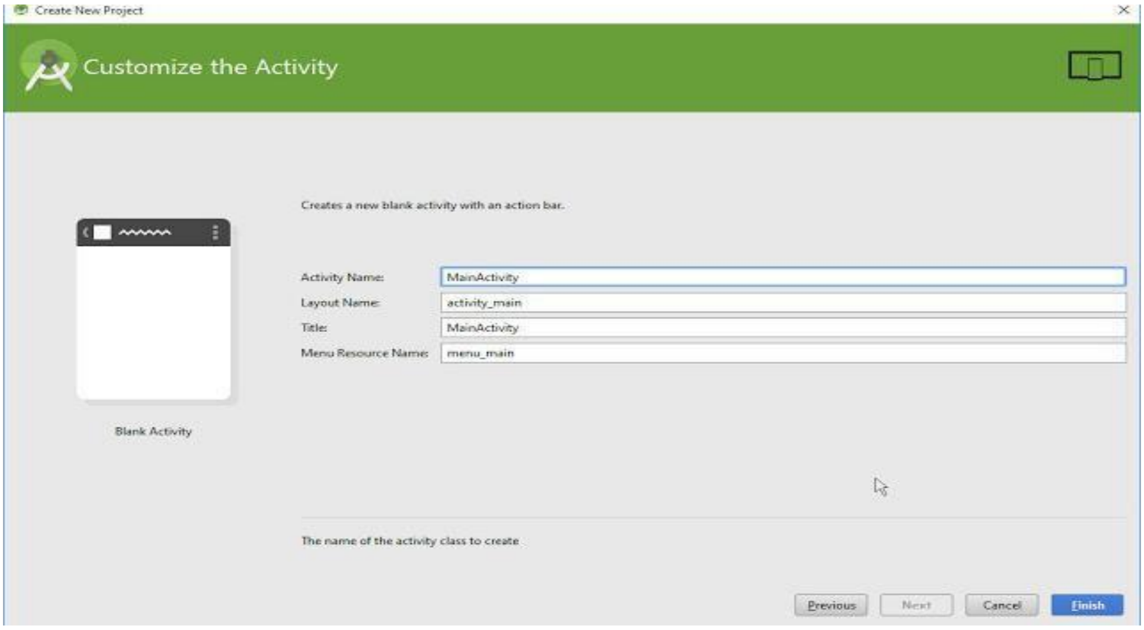
Step4: Now here you have to choose the activity where you will get java classes and xml files for designing and coding purpose. You can even choose no activity format. If you choose that there will be no auto generated codes on your project. If you select Blank Activity, you will get auto generated codes and that will save lots of time.



Create New Project

Add an activity to Mobile

Step5: In this you have to select layout name, activity name etc...



Creates a new blank activity with an action bar.

Activity Name: MainActivity
Layout Name: activity_main
Title: MainActivity
Menu Resource Name: menu_main

The name of the activity class to create

Than after you just have to click on Finish button. Now it will take little time to create your new project on Android Studio

Displaying the welcome message in AVD:

Step1: create a new project

Step 2: open a activity.xml file

XMLfile (activity_main.xml)

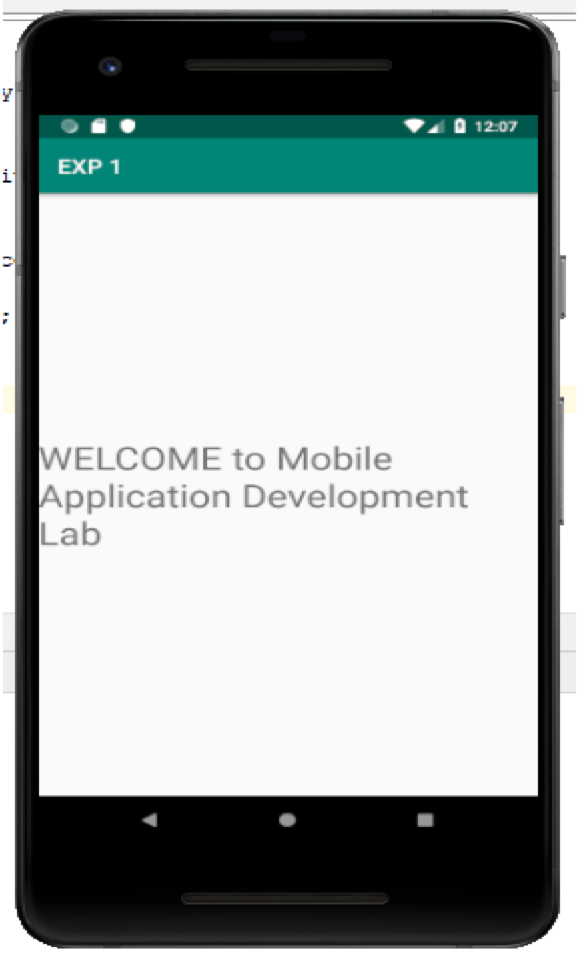
```
<?xml version="1.0" encoding="utf-8"?>

<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:textSize="32dp"
        android:text="WELCOME to Mobile Application Development Lab"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>
```

Step 3: open the java file

MainActivity.java

```
package com.example.exp1;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity {

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

step 4: Debug and run the application.



WELCOME to Mobile Application Development Lab

Exercise Program: b) Design an android application to display name, class, college, address etc. of a student.

EXPERIMENT – 3

AIM: a) How the life cycle of an activity is executed.

Step 1: open the new project

```
Java file(Mainactivity.java)
package com.example.myactivity101;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.app.Activity;
import android.os.Bundle;
import android.util.Log;
public class MainActivity extends AppCompatActivity {
            String tag=”Life cycle”;
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      Log.d(tag,"onCreate() invoked");
   }
     protected void onStart() {
      super.onStart();
      Log.d(tag,"onStart() invoked");
   }
   protected void onResume() {
      super.onResume();
      Log.d(tag,"onResume() invoked");
   }
   protected void onPause() {
      super.onPause();
      Log.d(tag,"onPause() invoked");
   }
   protected void onStop() {
      super.onStop();
      Log.d(tag,"onstop() invoked");
   }
   protected void onRestart() {
      super.onRestart();
      Log.d(tag,"onRestart() invoked");
   }
   protected void onDestroy() {
      super.onDestroy();
      Log.d(tag,"onDestroy() invoked");
   }
}
```

```
(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:orientation=”vertical”
   android:layout_height="match_parent"
   tools:context=".MainActivity">
   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Hello World!"       />
   <Button
      android:id="@+id/button"
      android:layout_width="wrap_content"
      android:layout_height="48dp"
      android:text="Button"      />
</android.support.constraint.ConstraintLayout>
```

Output:

When Activity is loaded , the following is displayed in the LogCat window.

```
D/Lifecycle : onCreate() invoked
D/Lifecycle : onStart() invoked
D/Lifecycle : onResume() invoked
```

If you click back button on the android emulator, the following is printed.

```
D/Lifecycle : onPause() invoked
D/Lifecycle : onStop() invoked
D/Lifecycle : onDestroy() invoked
```

Click home button and hold it there. Click the Activities icon and observe the following.

```
D/Lifecycle : onCreate() invoked
D/Lifecycle : onStart() invoked
D/Lifecycle : onResume() invoked
```

Click the phone button on the android emulator so that the activity is pushed to the background Observe the output in Logcat.

```
D/Lifecycle : onPause() invoked
D/Lifecycle : onStop() invoked
```

When closing the activity it display the following

```
D/Lifecycle : onRestart() invoked
D/Lifecycle : onStart() invoked
D/Lifecycle : onResume() invoked
```

b) Creating a basic Activity and applying a themes, styles to it.

In the manifest file apply attribute named as **android:theme.**

Manifestfile

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="com.example.s1">
   <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"
           android:theme="@style/Theme.AppCompat">
            <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
         </intent-filter>
      </activity>
   </application>
```

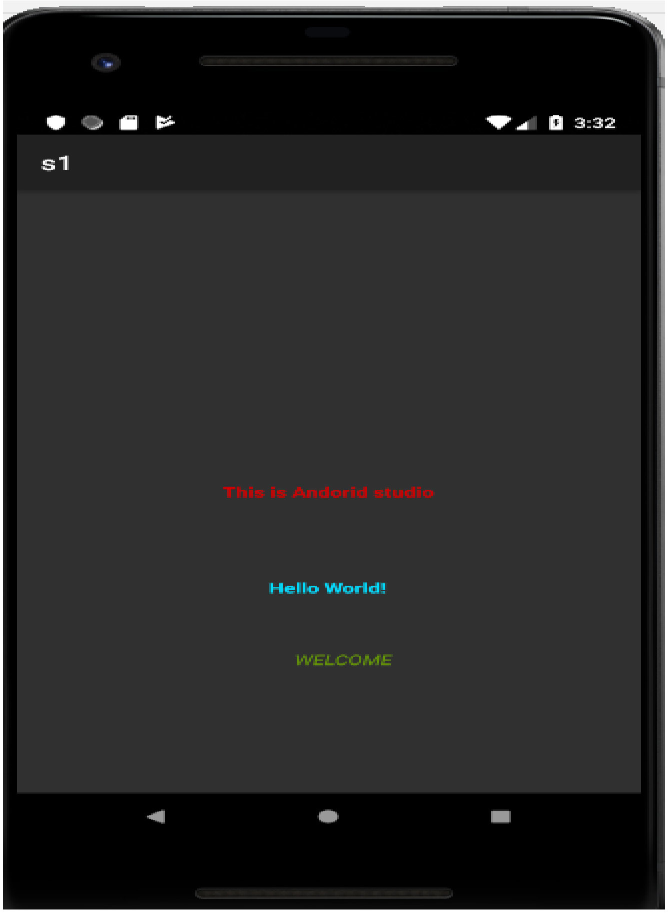
Activity.xml file

```
<?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"
   adroid:orientation="vertical"
   tools:context=".MainActivity" >
   <TextView
      android:id="@+id/textView3"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Hello World!”
      android:textColor="@android:color/holo_blue_bright"
      android:textStyle="bold”        />
   <TextView
      android:id="@+id/textView"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:textColor="@android:color/holo_red_dark"
      android:textStyle="bold"
      android:text="This is Andorid studio"/>
   <TextView
      android:id="@+id/textView2"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="@android:color/holo_green_dark"
        android:textStyle="italic"
        android:text="WELCOME"          />
</LinearLayout >
```

mainActivity.xml

```
package com.example.s1;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```



This is Andorid studio

Hello World!

WELCOME

Exercise Program: c) Create an android application which display three buttons on main activity named RED, GREEN and BLUE. The background colour of the activity should be changed to appropriate colour when the user selects any of these buttons.

EXPERIMENT – 4

AIM:4a) Displaying a dialog types of dialog objects

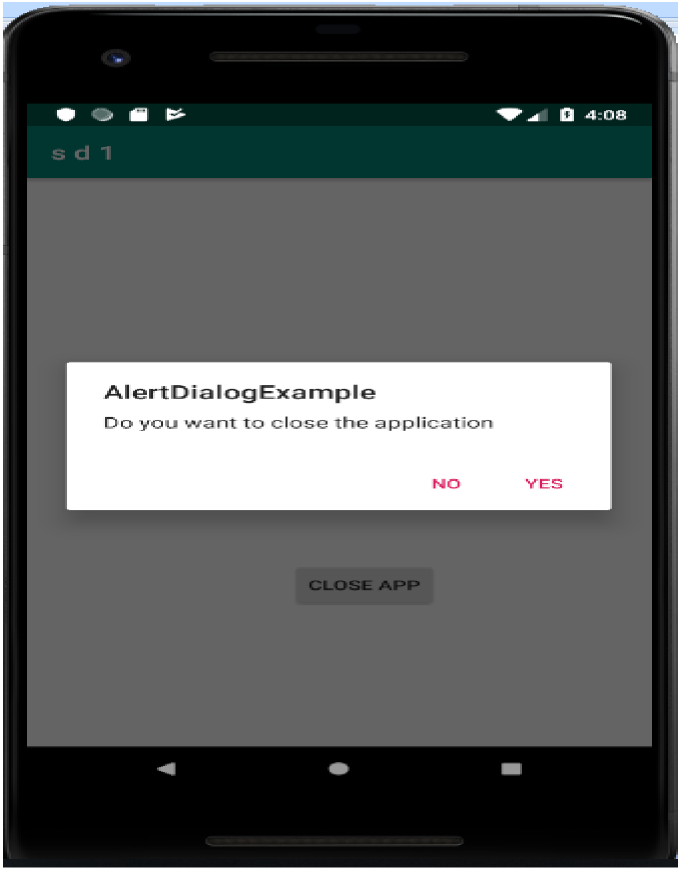
activity.xml file

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

mainActivity.java

```
package com.example.sd1;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
    Button closeButton;
    AlertDialog.Builder builder;
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        closeButton = (Button) findViewById(R.id.button);
        builder = new AlertDialog.Builder(this);
        closeButton.setOnClickListener(new View.OnClickListener() {
               public void onClick(View v) {
                              builder.setMessage("Do you want to close the application");
                              builder.setCancelable(false);
                builder.setPositiveButton("yes", new DialogInterface.OnClickListener() {
                 public void onClick(DialogInterface dialog, int which) {
```

```
                finish();
              Toast.makeText(getApplicationContext(), "",Toast.LENGTH_SHORT).show();
            }
        });

        builder.setNegativeButton("NO", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                dialog.cancel();
                Toast.makeText(getApplicationContext(),"you choose the no action in the
                         alert box",Toast.LENGTH_SHORT).show();
            }
        });
        AlertDialog alert =builder.create();
        alert.setTitle("AlertDialogExample");
        alert.show();
      }
    });
  }
}
```



AlertDialogExample

Do you want to close the application

Exercise Program: b) Develop a simple application showing a dialog box.

EXPERIMENT – 5

AIM: a) Linking of Activities with Intents.

Implicit Intents:

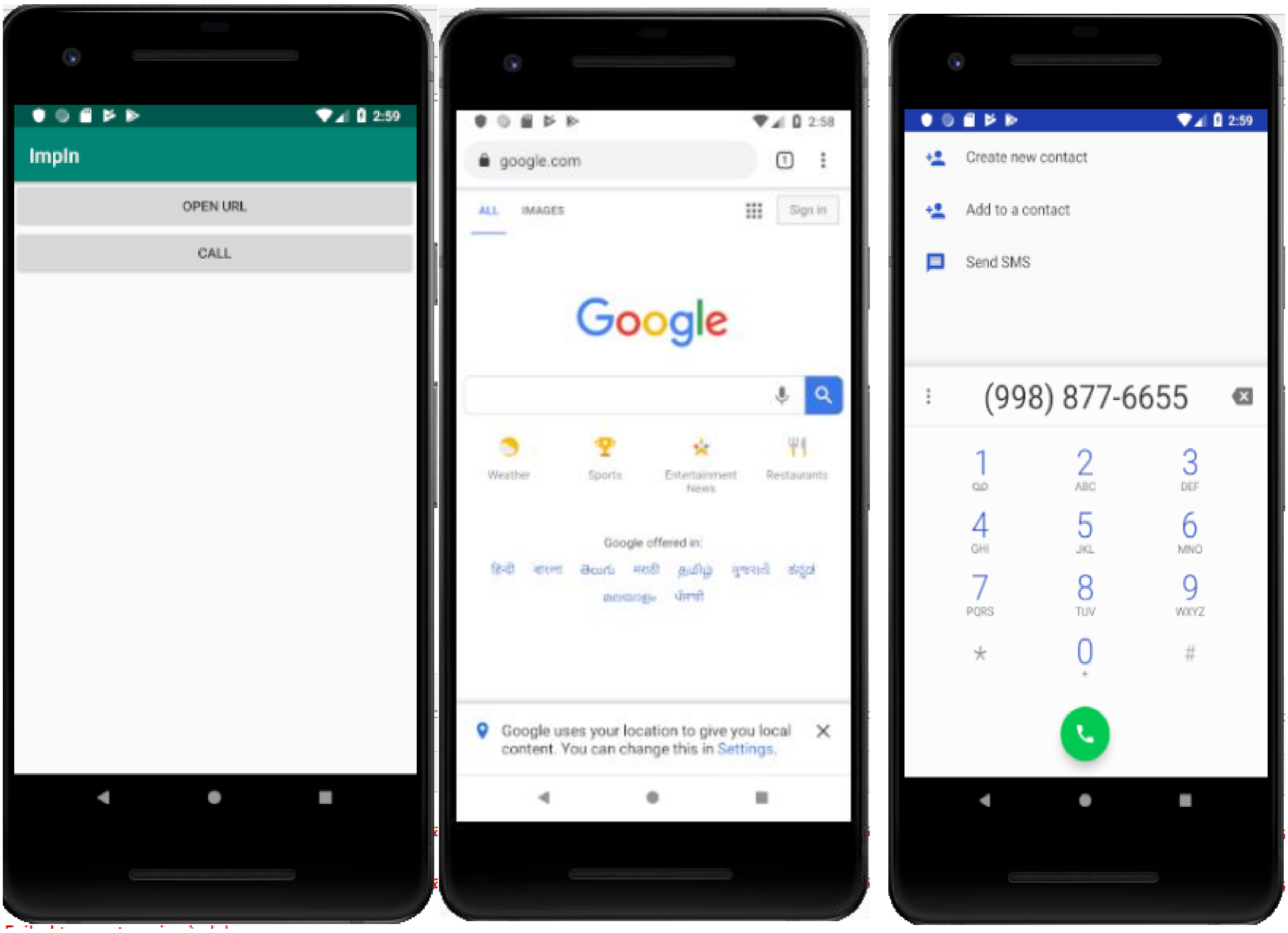
Activity_main.xml

```
<?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:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Open URL"
        android:onClick="openURL"       />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="call"
        android:onClick="makeacall"        />
</LinearLayout>
```

MainActivity.java

```
package com.example.impin;
import android.content.Intent;
import android.net.Uri;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class MainActivity extends AppCompatActivity {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    public void openURL(View view)
    {
        Uri u= Uri.parse("https://www.google.com");
        Intent i=new Intent(Intent.ACTION_VIEW,u);
        startActivity(i);
    }
    public void makeacall(View view)
    {
        Uri u=Uri.parse("tel:9988776655");
        Intent i=new Intent(Intent.ACTION_DIAL,u);
        startActivity(i);
    }
}
```

OUTPUT:



b)Linking of Activities with Explicit Intents:

Activity_main.xml

```
<?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"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
   <Button
       android:id="@+id/btn1"
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Display Second Activity"
        android:onClick="onClick"       />
</LinearLayout>
```

Mainactivity.java

```
package com.example.impin;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
public class MainActivity extends AppCompatActivity {
    Button B;
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        B = (Button) findViewById(R.id.btn1);
    }
    public void onClick(View view) {
        Intent i = new Intent(MainActivity.this, secondA.class);
        startActivity(i);
    }
}
```

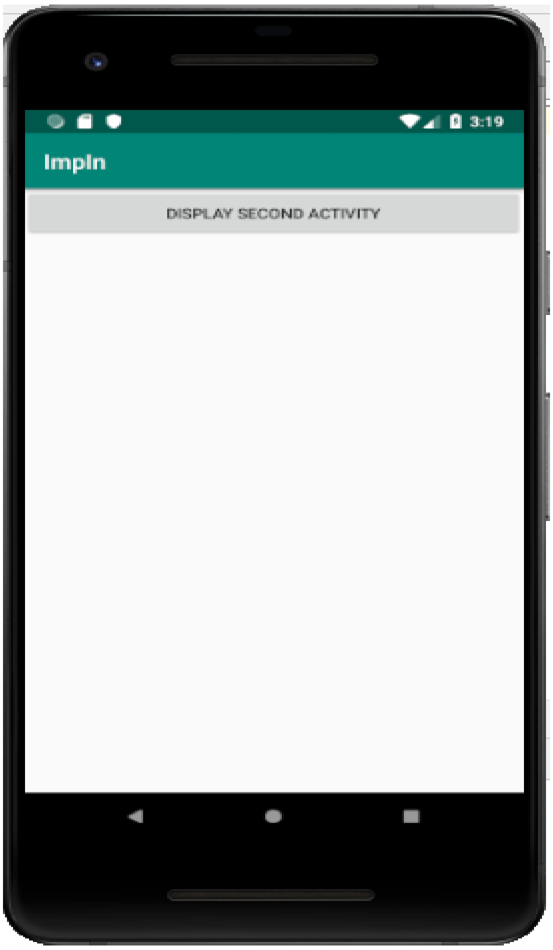
activity_second.xml

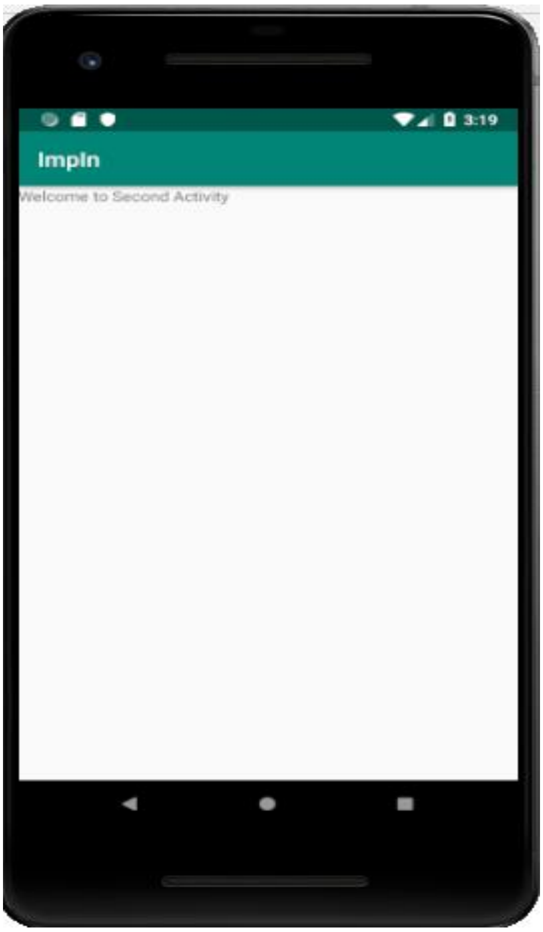
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".secondA">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Welcome to Second Activity"
        />
</LinearLayout>
```

secondA.java

```
package com.example.impin;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
public class secondA extends AppCompatActivity {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);
    }
}
```

output:





Welcome to Second Activity

Exercise Program: c) Design an android application login form whenever user select login in the another displays the message as "**WELCOME TO VIGNAN'S UNIVERSITY**"

EXPERIMENT – 6

AIM: a) Passing data using intent object.

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:orientation="vertical"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".MainActivity">
<Button
   android:layout_width="fill_parent"
   android:layout_height="wrap_content"
   android:text="display secongActivity"
   android:onClick="showSecond"/>
</LinearLayout>
```

MainActivity.java

```
package com.example.passdata;
import android.content.Intent;
import android.support.annotation.Nullable;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
   int request_Value=1;
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
   }
   public void showSecond(View view) {
      Intent i1 = new Intent(this, second.class);
      startActivityForResult(i1, request_Value);
   }
 protected void onActivityResult(int requestCode, int resultCode, Intent data)
{
      super.onActivityResult(requestCode, resultCode, data);
      if (requestCode == requestCode){
         if (resultCode == RESULT_OK) {
            Toast.makeText(this,data.getData().toString(), Toast.LENGTH_SHORT).show();
             }
            }
      }
}
```

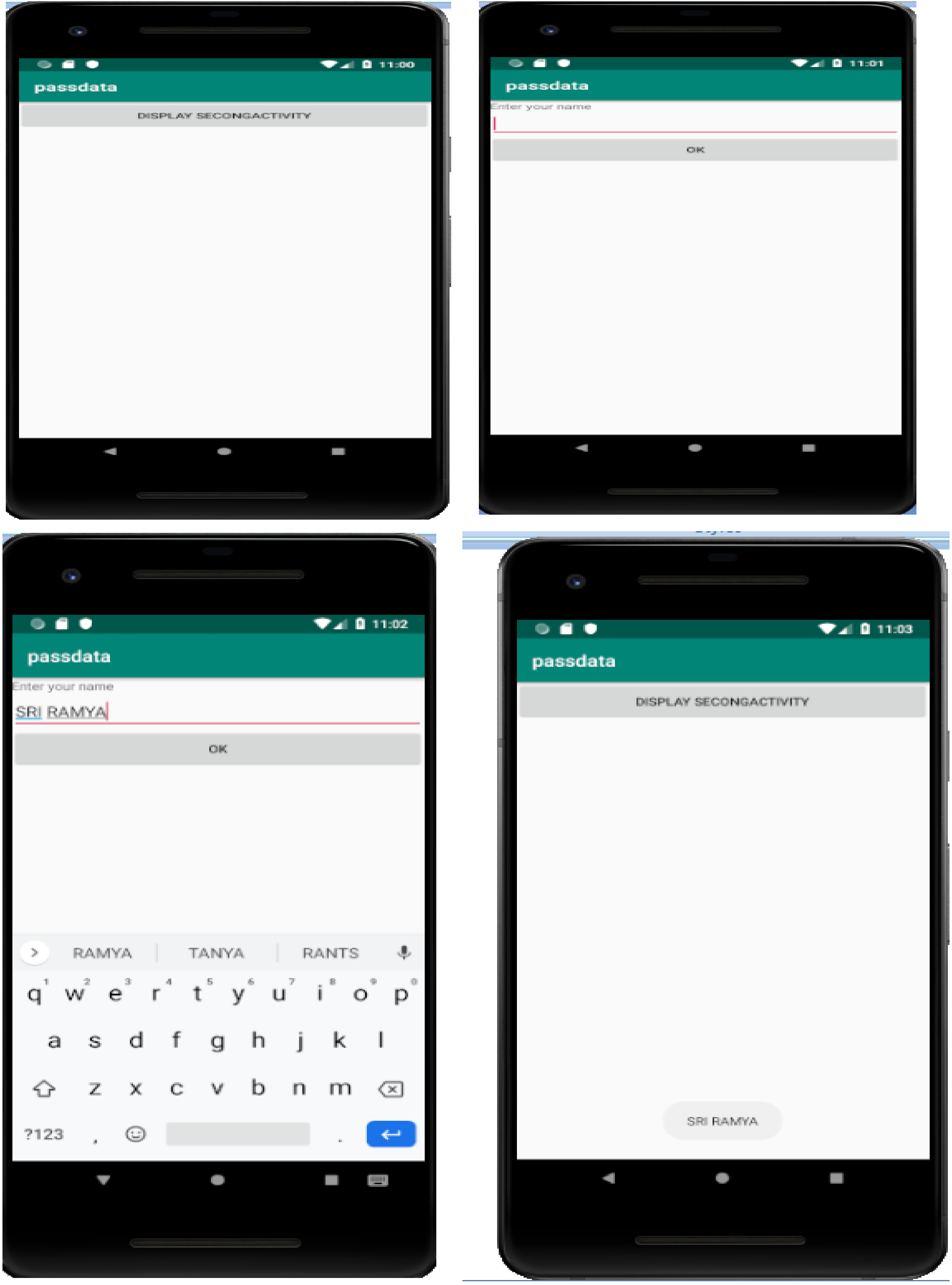
activity_second.xml

```
<?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:orientation="vertical"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".second">
<TextView
   android:id="@+id/t1"
   android:layout_width="fill_parent"
   android:layout_height="wrap_content"
   android:text="Enter your name"/>
   <EditText
      android:id="@+id/e1"
      android:layout_width="fill_parent"
      android:layout_height="wrap_content" />
   <Button
      android:id="@+id/b1"
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:text="ok"
      android:onClick="meth1"/>
</LinearLayout>
```

Second.java

```
package com.example.passdata;
import android.content.Intent;
import android.net.Uri;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
public class second extends AppCompatActivity {
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_second);
   }
   public void meth1(View view){
      Intent data=new Intent();
      EditText ed=(EditText)findViewById(R.id.e1);
      data.setData(Uri.parse(ed.getText().toString()));
      setResult(RESULT_OK,data);
      finish();
   }
}
```

output:



Exercise Program: b) Develop an application of basic calculator app.

EXPERIMENT – 7

AIM: a) Usage of Fragments and adding them Statically to the application.

Fragment1.xml

```
<?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”
        android:background=”#00FF00”  >
        <TextView
                android:layout_width=”fill_parent”
                android:layout_height=”wrap_content”
                android:text=”This is fragment #1”
                android:textColor=”#000000”
                android:textSize=”25sp” />
</LinearLayout>
```

Fragment2.xml

```
<?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”
        android:background=”#FFFE00”>
        <TextView
                android:layout_width=”fill_parent”
                android:layout_height=”wrap_content”
                android:text=”This is fragment #2”
                android:textColor=”#000000”
                android:textSize=”25sp” />
</LinearLayout>
```

Fragment1.java

```
package net.learn2develop.Fragments;
import android.app.Fragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
public class Fragment1 extends Fragment {
        public View onCreateView(LayoutInflater inflater,ViewGroup container, Bundle
        savedInstanceState) {
                    return inflater.inflate(R.layout.fragment1, container, false);
        }
}
```

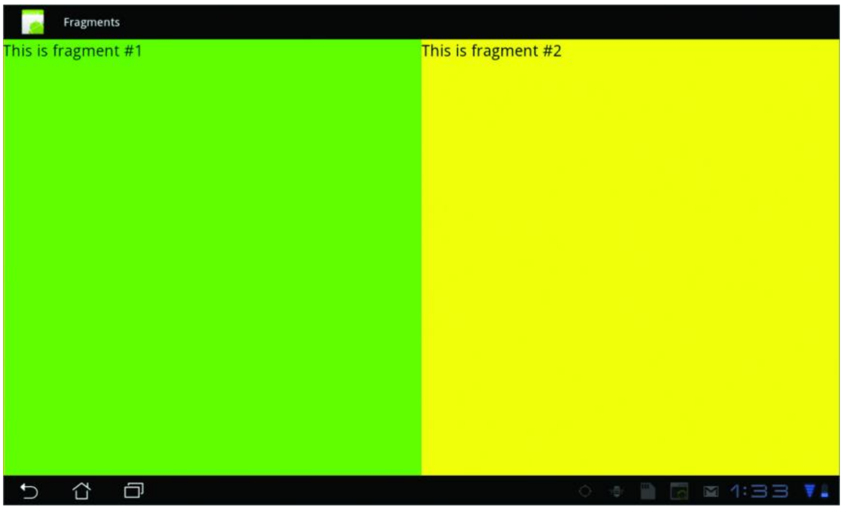
Fragment2.java

```
package net.learn2develop.Fragments;
import android.app.Fragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
public class Fragment2 extends Fragment {
        public View onCreateView(LayoutInflater inflater,ViewGroup container, Bundle
        savedInstanceState) {
                return inflater.inflate(R.layout.fragment2, container, false);
        }
}
```

Main_Activity.xml

```
<?xml version=”1.0” encoding=”utf-8”?>
<LinearLayout
        xmlns:android=”http://schemas.android.com/apk/res/android”
        android:orientation=”horizontal”
        android:layout_width=”fill_parent”
        android:layout_height=”fill_parent”>
        <Fragment
                android:name=”net.learn2develop.Fragmemts.Fragment1”
                android:id=”@+id/Fragment1”
                android:layout_width=”1”
                android:layout_width=”0px”
                android:layout_height=”match_parent”/>
          <Fragment
                android:name=”net.learn2develop.Fragmemts.Fragment2”
                android:id=”@+id/Fragment2”
                android:layout_width=”1”
                android:layout_width=”0px”
                android:layout_height=”match_parent”/>
</LinearLayout>
```

OUTPUT:



AIM: b) Usage of Fragments and adding them dynamically to the application.

Fragment1.xml

```
<?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”
        android:background=”#00FF00”  >
        <TextView
                android:layout_width=”fill_parent”
                android:layout_height=”wrap_content”
                android:text=”This is fragment #1”
                android:textColor=”#000000”
                android:textSize=”25sp” />
</LinearLayout>
```

Fragment2.xml

```
<?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”
        android:background=”#FFFE00”>
        <TextView
                android:layout_width=”fill_parent”
                android:layout_height=”wrap_content”
                android:text=”This is fragment #2”
                android:textColor=”#000000”
                android:textSize=”25sp” />
</LinearLayout>
```

Fragment1.java

```
package net.learn2develop.Fragments;
import android.app.Fragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
public class Fragment1 extends Fragment {
        public View onCreateView(LayoutInflater inflater,ViewGroup container, Bundle
        savedInstanceState) {
                        return inflater.inflate(R.layout.fragment1, container, false);
        }
}
```

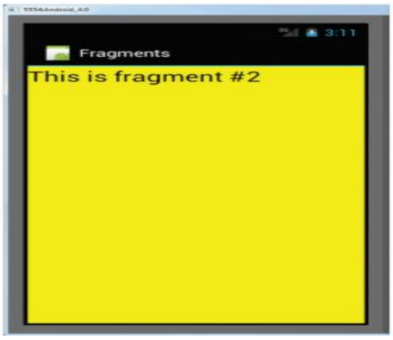
Fragment2.java

```
package net.learn2develop.Fragments;
import android.app.Fragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
public class Fragment2 extends Fragment {
        public View onCreateView(LayoutInflater inflater,ViewGroup container, Bundle
        savedInstanceState) {
                    return inflater.inflate(R.layout.fragment2, container, false);
        }
}
```

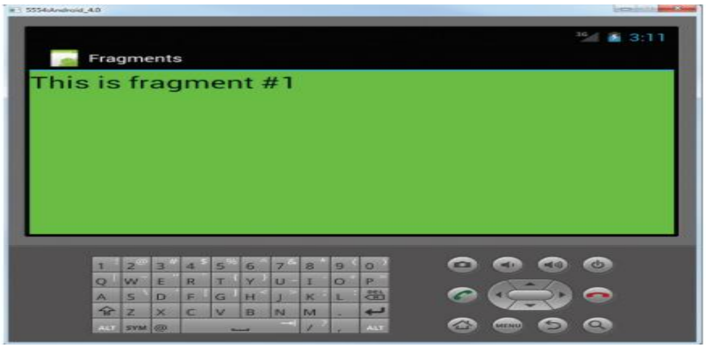
Main_Activity.java

```
package net.learn2develop.Fragments;
import android.app.Activity;
import android.app.FragmentManager;
import android.app.FragmentTransaction;
import android.os.Bundle;
import android.view.Display;
import android.view.WindowManager;
public class FragmentsActivity extends Activity {
        public void onCreate(Bundle savedInstanceState)
        {
                super.onCreate(savedInstanceState);
                FragmentManager fm = getFragmentManager();
                FragmentTransaction ft =fragmentManager.beginTransaction();
                WindowManager wm = getWindowManager();
                Display d = wm.getDefaultDisplay();
                if (d.getWidth() > d.getHeight())
                {
                        Fragment1 fragment1 = new Fragment1();
                        ft.replace(android.R.id.content, fragment1);
                }
                else
                {
                        Fragment2 fragment2 = new Fragment2();
                        ft.replace(android.R.id.content, fragment2);
                }
                ft.commit();
          }
}
```

OUTPUT:



3:11
Fragments
This is fragment #2



3:11
Fragments
This is fragment #1

EXPERIMENT – 8

AIM: Communication between fragments

Fragment1.xml

```
<?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”
        android:background=”#00FF00” >
        <TextView
                android:id=”@+id/lblFragment1”
                android:layout_width=”fill_parent”
                android:layout_height=”wrap_content”
                android:text=”This is fragment #1”
                android:textColor=”#000000”
                android:textSize=”25sp” />
</LinearLayout>
```

Fragments2.xml

```
<?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”
        android:background=”#FFFE00” >
        <TextView
                android:layout_width=”fill_parent”
                android:layout_height=”wrap_content”
                android:text=”This is fragment #2”
                android:textColor=”#000000”
                android:textSize=”25sp” />
        <Button
                android:id=”@+id/btnGetText”
                android:layout_width=”wrap_content”
                android:layout_height=”wrap_content”
                android:text=”Get text in Fragment #1”
                android:textColor=”#000000”
                android:onClick=”onClick” />
</LinearLayout>
```

Main.xml

```
<?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=”horizontal” >
          <fragment
                    android:name=”net.learn2develop.Fragments.Fragment1”
                    android:id=”@+id/fragment1”
                    android:layout_weight=”1”
                    android:layout_width=”0px”
                    android:layout_height=”match_parent” />
          <fragment
                    android:name=”net.learn2develop.Fragments.Fragment2”
                    android:id=”@+id/fragment2”
                    android:layout_weight=”1”
                    android:layout_width=”0px”
                    android:layout_height=”match_parent” />
</LinearLayout>
```

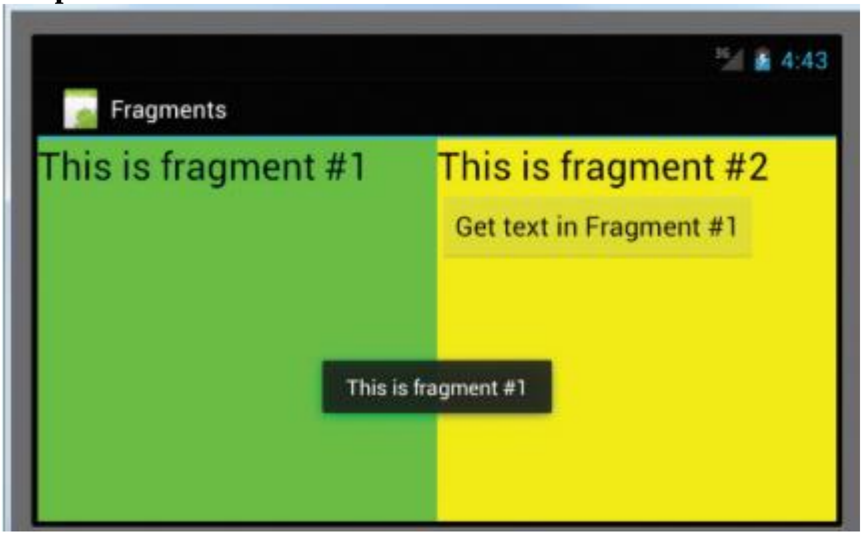
Mainactivity.java

```
public class FragmentsActivity extends Activity {
        public void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.main);
        }
}
```

Fragment2.java

```
package net.learn2develop.Fragments;
import android.app.Fragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
public class Fragment2 extends Fragment {
        public View onCreateView(LayoutInflater inflater,ViewGroup container, Bundle
        savedInstanceState) {
                 return inflater.inflate(R.layout.fragment2, container, false);
        }
        public void onStart() {
                super.onStart();
                Button btnGetText = (Button)getActivity().findViewById(R.id.btnGetText);
                btnGetText.setOnClickListener(new View.OnClickListener() {
                    public void onClick(View v) {
                        TextView lbl = (TextView)getActivity().findViewById (R.id.lblFragment1);
                    Toast.makeText(getActivity(), lbl.getText(),Toast.LENGTH_SHORT). show();
                    }
                });
        }
}
```

Output:



This is fragment #1

This is fragment #2

This is fragment #1

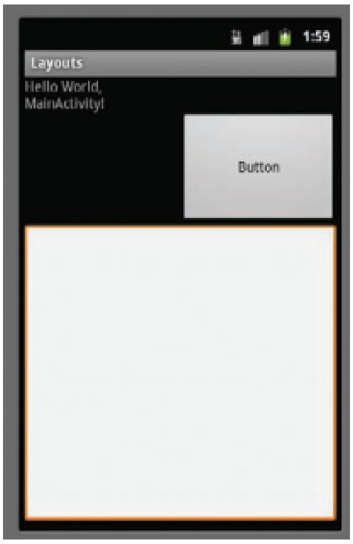
Exercise Program: b) Develop an application of BMI calculation.

EXPERIMENT – 9

AIM: Creating various layouts.

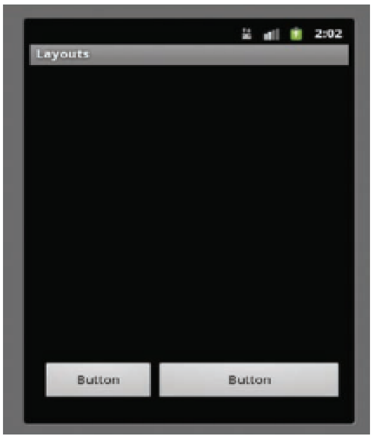
a) Linear Layout

```
<?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”>
        <TextView
                android:layout_width=”105dp”
                android:layout_height=”wrap_content”
                android:text=”@string/hello” />
        <Button
                android:layout_width=”160dp”
                android:layout_height=”wrap_content”
                android:text=”Button”
                android:layout_gravity=”right”
                android:layout_weight=”0.2” />
        <EditText
                android:layout_width=”fill_parent”
                android:layout_height=”wrap_content”
                android:textSize=”18sp”
                android:layout_weight=”0.8” />
</LinearLayout>
```



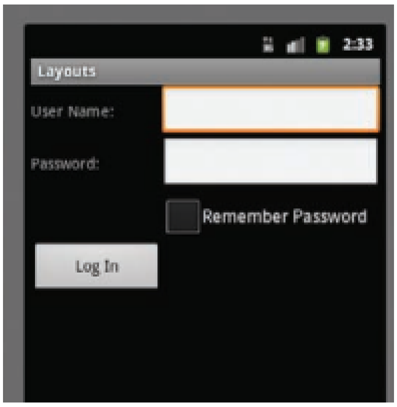
b) Absolute Layout

```
<?xml version=”1.0” encoding=”utf-8”?>
<AbsoluteLayout
        android:layout_width=”fill_parent”
        android:layout_height=”fill_parent”
        xmlns:android=”http://schemas.android.com/apk/res/android”>
        <Button
                android:layout_width=”188dp”
                android:layout_height=”wrap_content”
                android:text=”Button”
                android:layout_x=”126px”
                android:layout_y=”361px”/>
        <Button
                android:layout_width=”113dp”
                android:layout_height=”wrap_content”
                android:text=”Button”
                android:layout_x=”12px”
                android:layout_y=”361px”/>
</AbsoluteLayout>
```

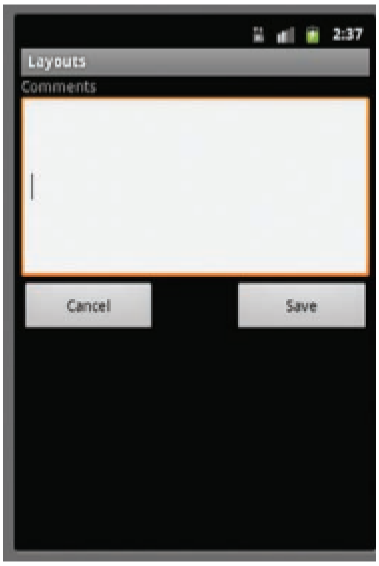


c) Table Layout

```
<?xml version=”1.0” encoding=”utf-8”?>
<TableLayout
        xmlns:android=”http://schemas.android.com/apk/res/android”
        android:layout_height=”fill_parent”
        android:layout_width=”fill_parent”>
        <TableRow>
                <TextView
                android:text=”User Name:”
                android:width =”120px”/>
                <EditText
                android:id=”@+id/txtUserName”
                android:width=”200px” />
        </TableRow>
        <TableRow>
                <TextView
                android:text=”Password:”/>
                <EditText
                android:id=”@+id/txtPassword”
                android:password=”true”/>
        </TableRow>
        <TableRow>
                <TextView />
                <CheckBox
                android:id=”@+id/chkRememberPassword”
                android:layout_width=”fill_parent”
                android:layout_height=”wrap_content”
                android:text=”Remember Password”/>
        </TableRow>
        <TableRow>
                <Button
                android:id=”@+id/buttonSignIn”
                android:text=”Log In” />
        </TableRow>
</TableLayout>
```



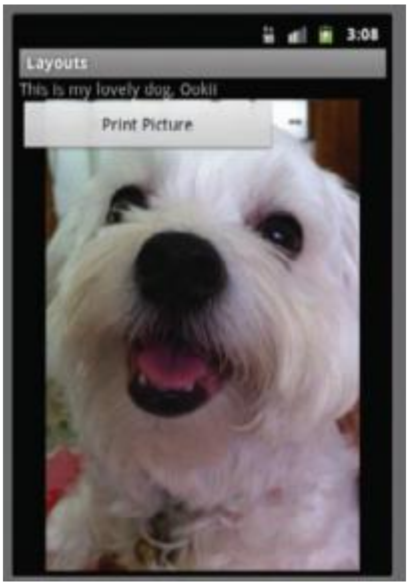
```
<?xml version=”1.0” encoding=”utf-8”?>
<RelativeLayout
        android:id=”@+id/RLayout”
        android:layout_width=”fill_parent”
        android:layout_height=”fill_parent”
        xmlns:android=”http://schemas.android.com/apk/res/android”>
        <TextView
                android:id=”@+id/lblComments”
                android:layout_width=”wrap_content”
                android:layout_height=”wrap_content”
                android:text=”Comments”
                android:layout_alignParentTop=”true”
```



```
                android:layout_alignParentLeft=”true”/>
        <EditText
                android:id=”@+id/txtComments”
                android:layout_width=”fill_parent”
                android:layout_height=”170px”
                android:textSize=”18sp”
                android:layout_alignLeft=”@+id/lblComments”
                android:layout_below=”@+id/lblComments”
                android:layout_centerHorizontal=”true”/>
        <Button
                android:id=”@+id/btnSave”
                android:layout_width=”125px”
                android:layout_height=”wrap_content”
                android:text=”Save”
                android:layout_below=”@+id/txtComments”
                android:layout_alignRight=”@+id/txtComments”/>
        <Button
                android:id=”@+id/btnCancel”
                android:layout_width=”124px”
                android:layout_height=”wrap_content”
                android:text=”Cancel”
                android:layout_below=”@+id/txtComments”
                android:layout_alignLeft=”@+id/txtComments”/>
</RelativeLayout>
```

d) Frame Layout

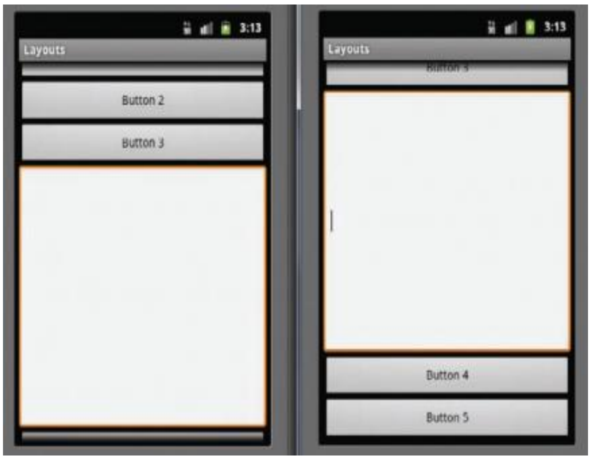
```
<?xmlversion=”1.0”encoding=”utf-8”?>
<RelativeLayout
        android:id=”@+id/Rlayout”
        android:layout_width=”fill_parent”
        android:layout_height=”fill_parent”
        xmlns:android=”http://schemas.android.com/apk/res/android”>
        <TextView
                android:id=”@+id/lblcomments”
                android:layout_width=”wrap_content”
                android:layout_height=”wrap_content”
                android:text=”Hello, Android”
                android:layout_alignParentTop=”true”
                android:layout_alignParentLeft=”true”/>
        <FrameLayout
        android:layout_width=”wrap_content”
        android:layout_height=”wrap_content”
        android:layout_alignLeft=”@+id/lblComments”
        android:layout_below=”@+id/lblComments”
        android:layout_centerHorizontal=”true”>
        <ImageView
                android:src = “@drawable/ookii”
                android:layout_width=”wrap_content”
                android:layout_height=”wrap_content”/>
        <Button
```



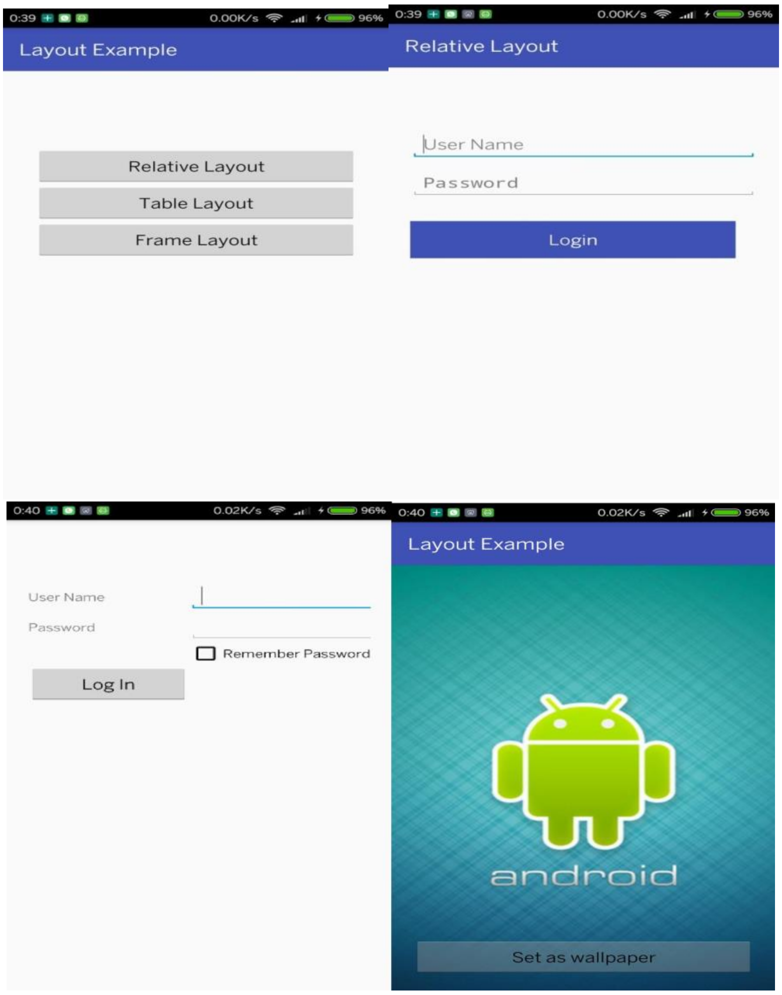
```
        android:layout_width=”124dp”
        android:layout_height=”wrap_content”
        android:text=”Print Picture” />
</FrameLayout>
```

e) Scroll view

```
<?xmlversion=”1.0”encoding=”utf-8”?>
<ScrollView
    android:layout_width=”fill_parent”
    android:layout_height=”fill_parent”
    xmlns:android=”http://schemas.android.com/apk/res/android”>
    <LinearLayout
        android:layout_width=”fill_parent”
        android:layout_height=”wrap_content”
        android:orientation=”vertical”>
    <Button
    android:id=”@+id/button1”
    android:layout_width=”fill_parent”
    android:layout_height=”wrap_content”
    android:text=”Button 1”/>
    <Button
    android:id=”@+id/button2”
    android:layout_width=”fill_parent”
    android:layout_height=”wrap_content”
    android:text=”Button 2”/>
    <Button
        android:id=”@+id/button3”
        android:layout_width=”fill_parent”
        android:layout_height=”wrap_content”
        android:text=”Button 3”/>
        <EditText
            android:id=”@+id/txt”
            android:layout_width=”fill_parent”
            android:layout_height=”300px”/>
        <Button
            android:id=”@+id/button4”
            android:layout_width=”fill_parent”
            android:layout_height=”wrap_content”
            android:text=”Button 4”/>
        <Button
            android:id=”@+id/button5”
            android:layout_width=”fill_parent”
            android:layout_height=”wrap_content”
            android:text=”Button 5”/>
    </LinearLayout>
</ScrollView>
```



Exercise Program: f) Design an application which contains some buttons named as relative layout, Linear layout, Table Layout…etc. Whenever the user selects the corresponding button it will opens a new activity which are designed according to the name displayed in the button.



EXPERIMENT – 10

AIM: Displaying Action bar.

Step1: Create a new project and name as exp1 and then select the Empty Activity, and click finish.

Step 2: open the xml file and write the code.

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="wrap_content"
   android:layout_height="wrap_content"
   android:text="Welcome to Mobile Application Development lab"/>

</LinearLayout>
```

Step3: In java create a method **onCreateOptionsMenu().** For this one goto res→ and create a folder menu, in that new→Vector Asset→name that as menu1.

Open that menu 1 and add the items and subitems in this file.

And to select the items implement **onOptionsItemSelected()** by using the switch loop.And to know which item is selected implement Toast.makeText.show().

Main.java

```
package com.example.exp1;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

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

```
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater i=getMenuInflater();
        i.inflate(R.menu.menu1,menu);
        return true;
    }
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.item1:
                Toast.makeText(this," The selected Item is 1", Toast.LENGTH_SHORT).show();
                return true;
            case R.id.item2:
                Toast.makeText(this," The selected Item is 2", Toast.LENGTH_SHORT).show();
                return true;
            case R.id.item3:
                Toast.makeText(this," The selected Item is 3", Toast.LENGTH_SHORT).show();
                return true;
            case R.id.item4:
                Toast.makeText(this," The selected Item is 4", Toast.LENGTH_SHORT).show();
                return true;
            case R.id.item7:
                Toast.makeText(this," The Item is alaram", Toast.LENGTH_SHORT).show();
                return true;
            case R.id.sitem1:
                Toast.makeText(this,” selected sub Item is 1", Toast.LENGTH_SHORT).show();
                return true;
            case R.id.sitem2:
                Toast.makeText(this," selected sub item is 2", Toast.LENGTH_SHORT).show();
                return true;
                default:return super.onOptionsItemSelected(item);
        }
    }
}
```

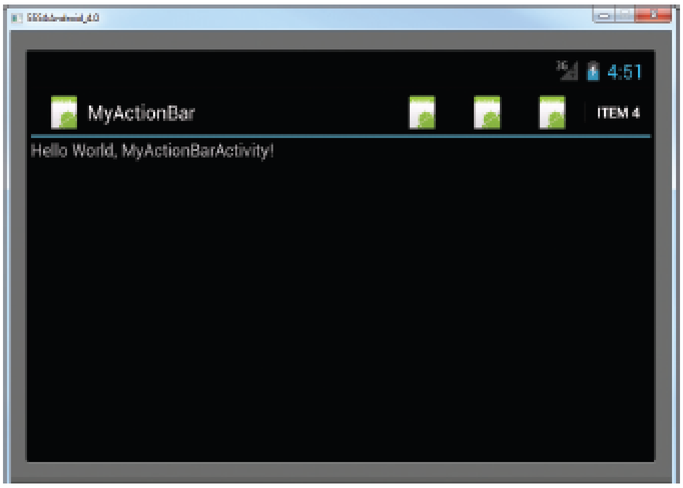
menu1.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <item
        android:id="@+id/item7"
        android:icon="@drawable/ic_alarm_add_black_24dp"
        android:title="alaram"
        app:showAsAction="ifRoom"
        />
    <item
        android:id="@+id/item1"
        android:icon="@drawable/ic_launcher_background"
        android:title="Item 1"
        app:showAsAction="never"
        />
```

```
  <item
     android:id="@+id/item2"
     android:icon="@color/colorAccent"
     android:title="Item 2"
     app:showAsAction="never" >

     <menu>
           <item
              android:id="@+id/sitem1"
              android:title="sub Item 1"
              app:showAsAction="ifRoom"
              />
           <item
              android:id="@+id/sitem2"
              android:title="sub Item 2"
              app:showAsAction="ifRoom"
              />
     </menu>
  </item>

  <item
     android:id="@+id/item3"
     android:title="Item 3"
     app:showAsAction="never"
     />
  <item
     android:id="@+id/item4"
     android:title="Item 4"
     app:showAsAction="never"
     />
</menu>
```

Step 4: Run the application the output is displayed.

OUTPUT:



```
Hello World, MyActionBarActivity!
```

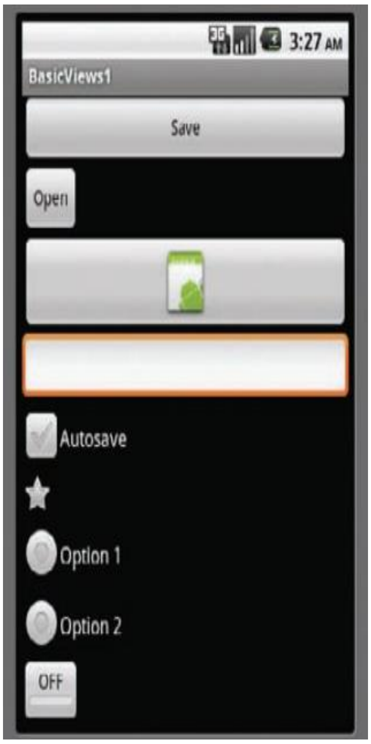
Exercise Program: b) Design an Action bar which is similar to WHATS APP.

EXPERIMENT – 11

AIM: a) Handling view events

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:orientation="vertical"
             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="hello"          />
             <Button
                          android:id= "@id/btnsave"
                          android:layout_width="match_parent"
                          android:layout_height="wrap_content"
                          android:text="Save"           />
             <Button
                         android:id= "@id/btnopen"
                         android:layout_width="wrap_content"
                         android:layout_height="wrap_content"
                         android:text="open"            />

             <ImageButton
                         android:id= "@id/btnsave"

                          android:layout_width="match_parent"
                         android:layout_height="wrap_content"
                         android:src="@drawable/icon"/>
             <EditText
                         android:id= "@id/txtname"
                         android:layout_width="match_parent"
                         android:layout_height="wrap_content"
/>
             <Checkbox
                         android:id= "@id/chkautosave"
                         android:layout_width="match_parent"
                         android:layout_height="wrap_content"
                         android:text="AutoSave"     />
             <Checkbox
                         android:id= "@id/star"
                         style=”?android:att/startStyle”
                         android:layout_width="wrap_content"
                         android:layout_height="wrap_content"    />
```



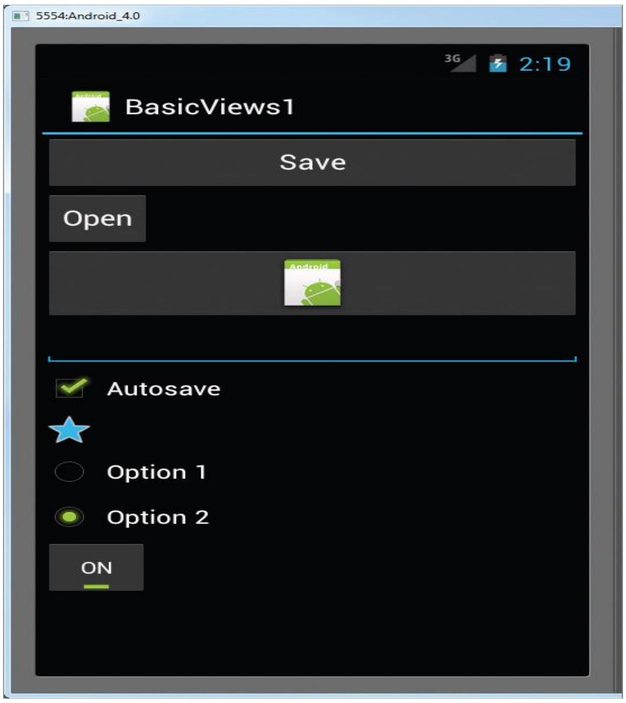
```
        <RadioGroup
                android:id= “@id/rdbgp1”
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="vertical"          >
                <RadioButton
                        android:id= “@id/rdb1”
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:text=”option1"      />
                <RadioButton
                        android:id= “@id/rdb2”
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:text=”option2"      />
        </RadioGroup>
        <ToggleButton
                android:id= “@id/toggle1”
                android:layout_width="wrap_content"
                android:layout_height="wrap_content”    />
</LinearLayout>
```

Main.java

```
package net.learn2develop.BasicViews1;
 import android.app.Activity;
 import android.os.Bundle;
 import android.view.View;
 import android.widget.Button;
 import android.widget.CheckBox;
 import android.widget.RadioButton;
import android.widget.RadioGroup;
 import android.widget.Toast;
 import android.widget.ToggleButton;
import android.widget.RadioGroup.OnCheckedChangeListener;
public class MainActivity extends Activity {
        public void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.main);
                Button btnOpen = (Button) findViewById(R.id.btnOpen);
                btnOpen.setOnClickListener(new View.OnClickListener() {
                        public void onClick(View v) {
                                DisplayToast(“You have clicked the Open button”);
                        }
                 });
        Button btnSave = (Button) findViewById(R.id.btnSave);
        btnSave.setOnClickListener(new View.OnClickListener()
        {
                public void onClick(View v) {
                        DisplayToast(“You have clicked the Save button”);
```

```
			}
		});
		CheckBox  checkBox = (CheckBox) findViewById(R.id.chkAutosave);
		checkBox.setOnClickListener(new View.OnClickListener()
		{
			public void onClick(View v) {
			 if (((CheckBox)v).isChecked())
				DisplayToast(“CheckBox is checked”);
			else
				DisplayToast(“CheckBox is unchecked”);
			 }
		 });
		RadioGroup radioGroup = (RadioGroup) findViewById(R.id.rdbGp1);
		radioGroup.setOnCheckedChangeListener(new OnCheckedChangeListener()
		{
			 public void onCheckedChanged(RadioGroup group, int checkedId) {
				RadioButton rb1 = (RadioButton) findViewById(R.id.rdb1);
				if (rb1.isChecked()) {
					DisplayToast(“Option 1 checked!”);
				}
				else
				{
					DisplayToast(“Option 2 checked!”);
				}
			}
		});
		ToggleButton toggleButton = (ToggleButton) findViewById(R.id.toggle1);
		toggleButton.setOnClickListener(new View.OnClickListener()
		{
			public void onClick(View v) {
			 if (((ToggleButton)v).isChecked())
				DisplayToast(“Toggle button is On”);
			else
				DisplayToast(“Toggle button is Off”);
			}
			});
		}
		private void DisplayToast(String msg)
		{
			Toast.makeText(getBaseContext(), msg, Toast.LENGTH_SHORT).show();
		}

}
```

OUTPUT:



Exercise b): Create an application of scoreboard app

12.**Exercise:** Create an application by following all the conditions

- Registration form by including all basic views,
- Display the dialog box as “Please verify all the details once again”
- Successfully register, Open the activity is designed with fragments
- Navigate to another activity present in different application.
- Develop the action bar for that activity.