Week 10: Implement an application that writes data to the SD card.

Android Application that writes data to the SD Card

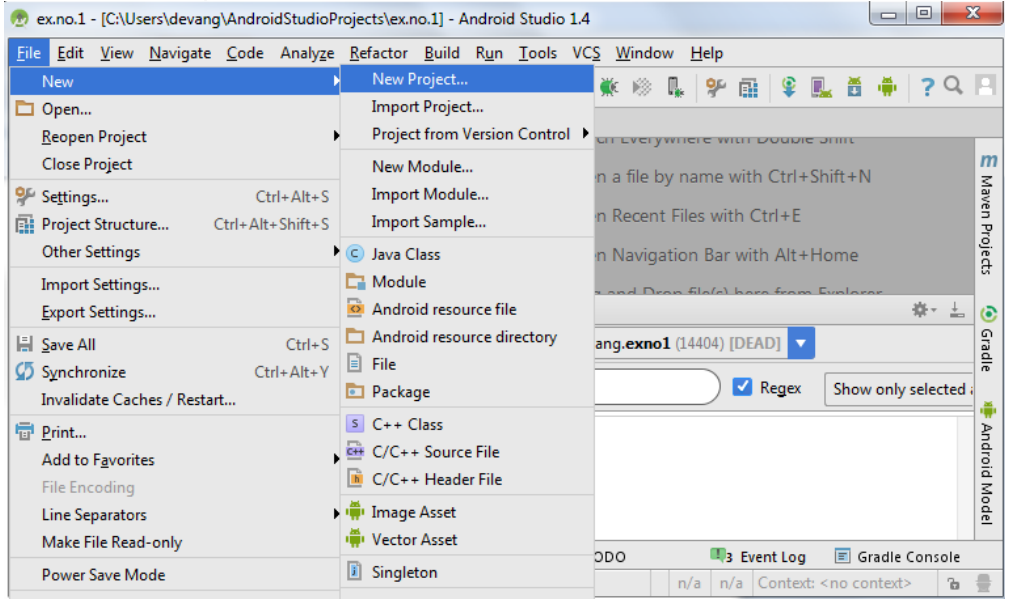
Aim:

To develop a Android Application that writes data to the SD Card.

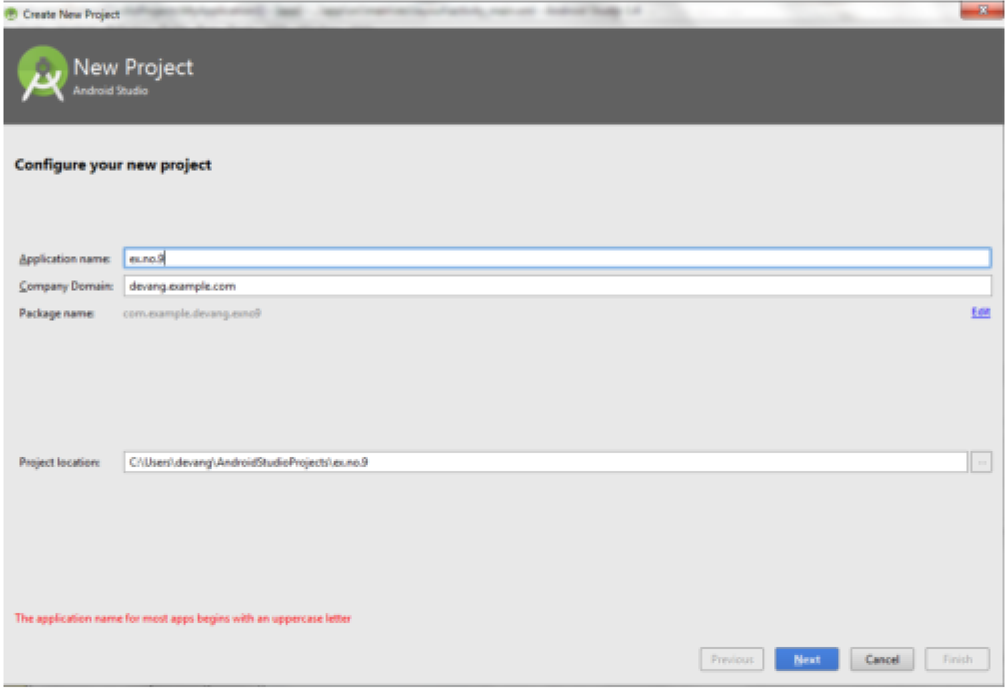
Procedure:

Creating a New project:

- Open Android Studio and then click on File -> New -> New project.



- Then type the Application name as "ex.no.10" and click Next.



Configure your new project

Application name: ex.no.9

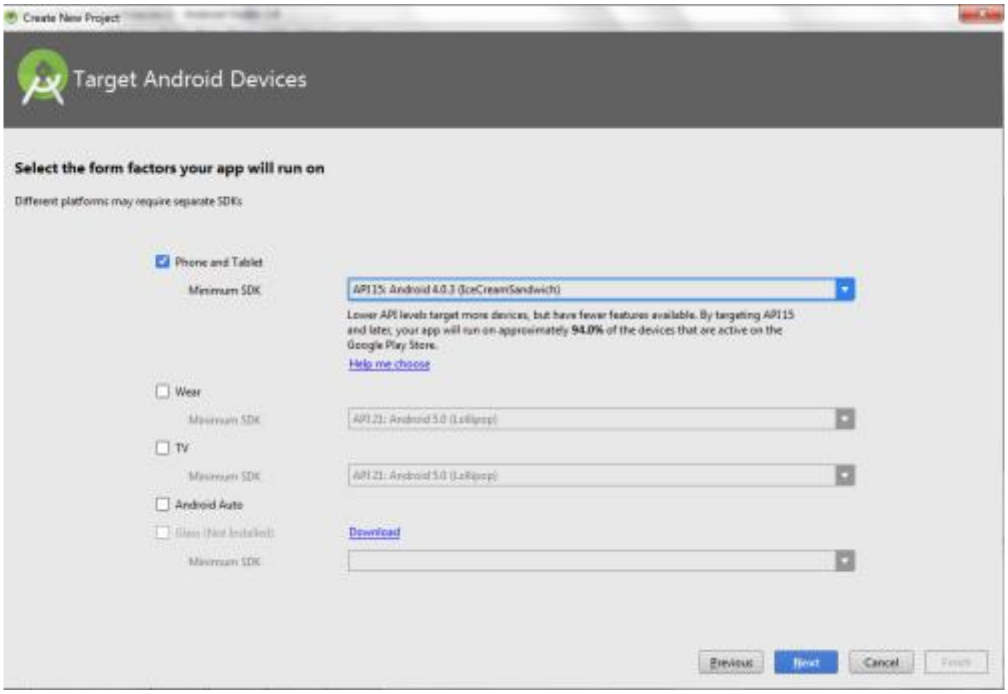
Company Domain: devang.example.com

Package name: com.example.devang.exno9

Project location: C:\Users\devang\AndroidStudioProjects\ex.no.9

The application name for most apps begins with an uppercase letter

- Then select the Minimum SDK as shown below and click Next.


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: API15: Android 4.0.3 (IceCreamSandwich)

Lower API levels target more devices, but have fewer features available. By targeting API15 and later, your app will run on approximately **94.0%** of the devices that are active on the Google Play Store.

Help me choose

☐ Wear

Minimum SDK: API 21: Android 5.0 (Lollipop)

☐ TV

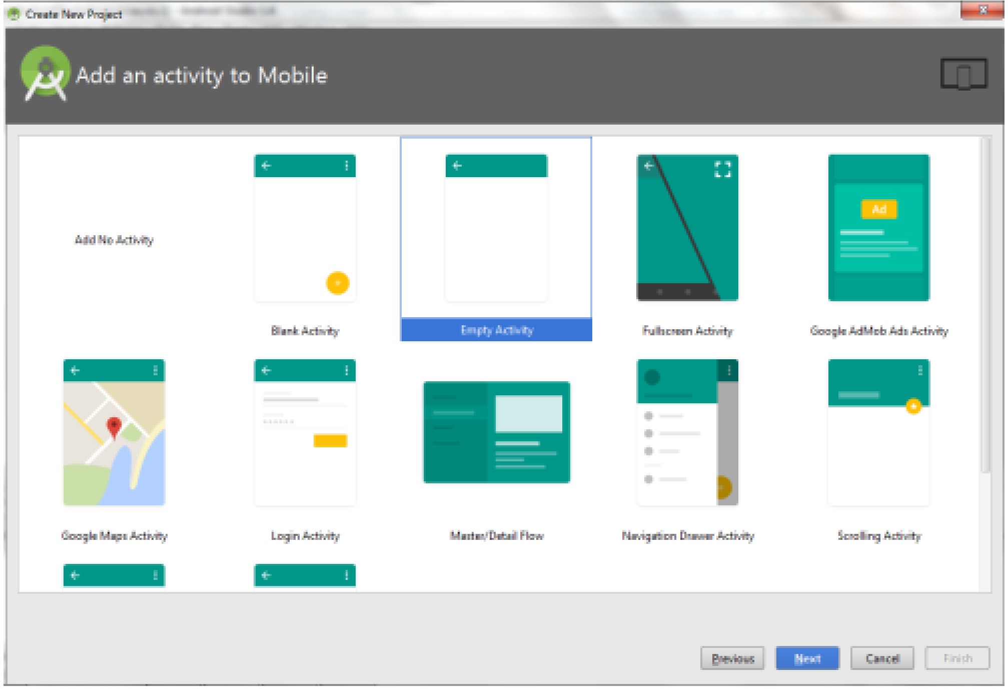
Minimum SDK: API 21: Android 5.0 (Lollipop)

☐ Android Auto

☐ Glass (Not Installed) Download

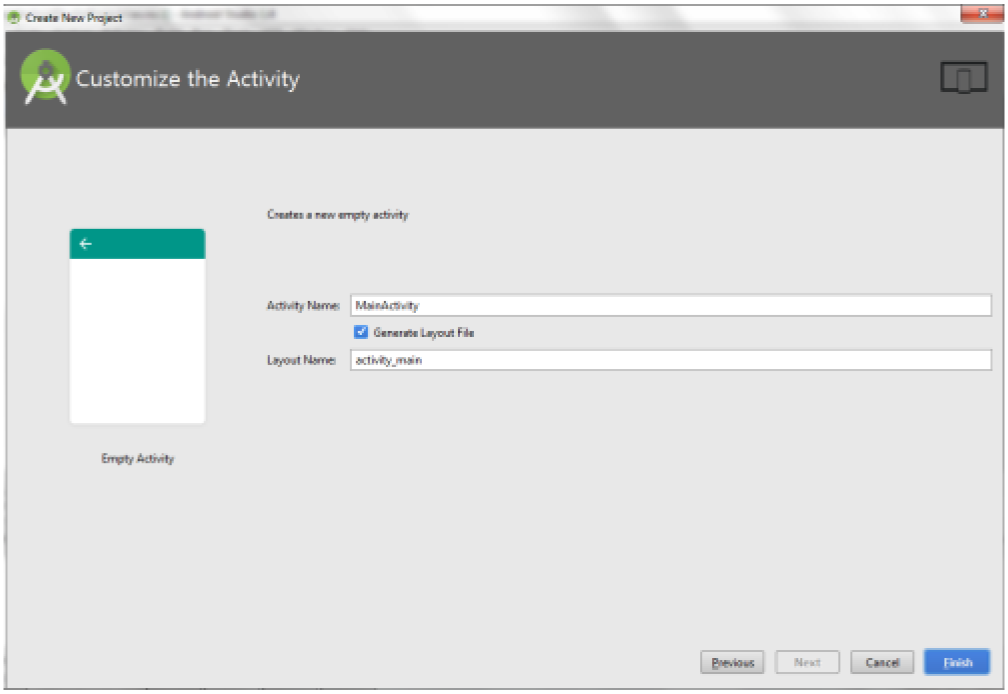
Minimum SDK:

- Then select the Empty Activity and click Next.


Create New Project

Add an activity to Mobile

- Finally click Finish.



Customize the Activity

Creates a new empty activity

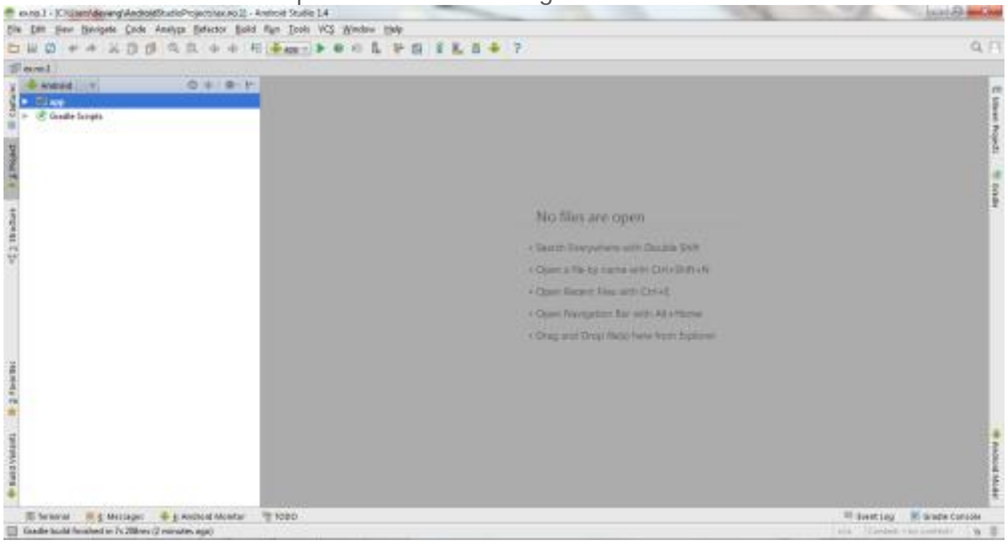
Activity Name: MainActivity

☑ Generate Layout File

Layout Name: activity_main

Empty Activity

- It will take some time to build and load the project.
- After completion it will look as given below.



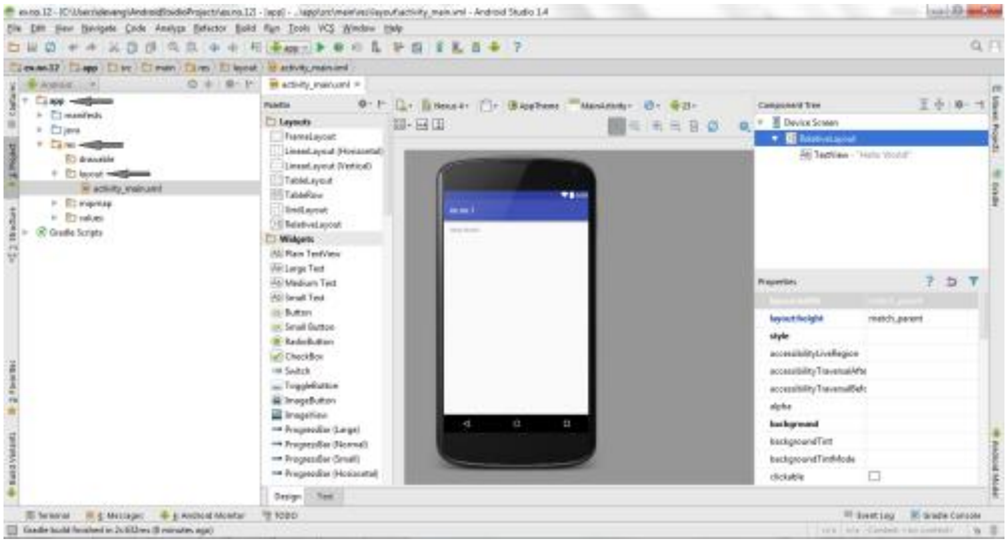
No files are open

- Search Everywhere with Double Shift
- Open a file by name with Ctrl+Shift+N
- Open Recent Files with Ctrl+E
- Open Navigation Bar with Alt+Home
- Drag and Drop file(s) here from Explorer

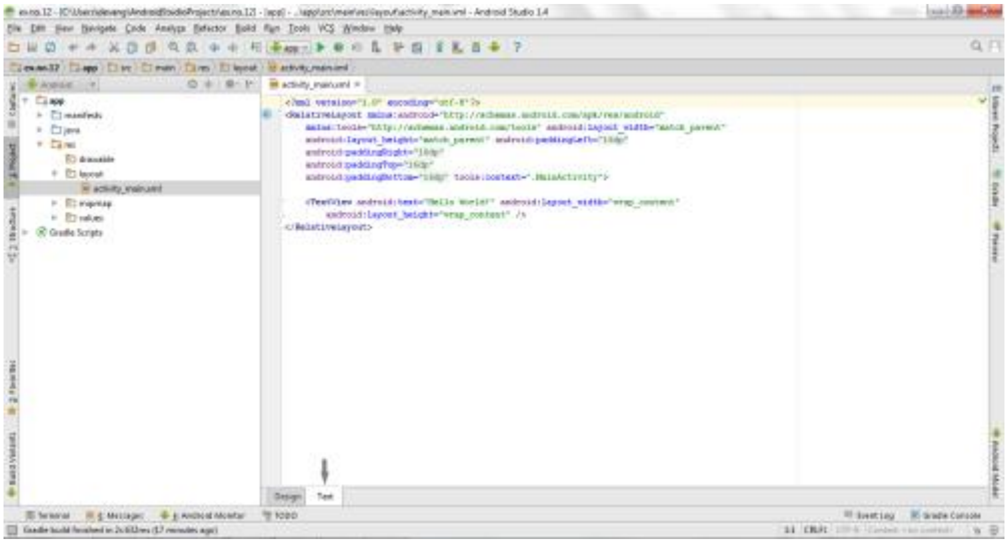
Gradle build finished in 7s 28ms (2 minutes ago)

Designing layout for the Android Application:

- Click on app -> res -> layout -> activity_main.xml.



- Now click on Text as shown below.



```
<?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:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp" tools:context=".MainActivity">

    <TextView android:text="Hello World!" android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</RelativeLayout>
```

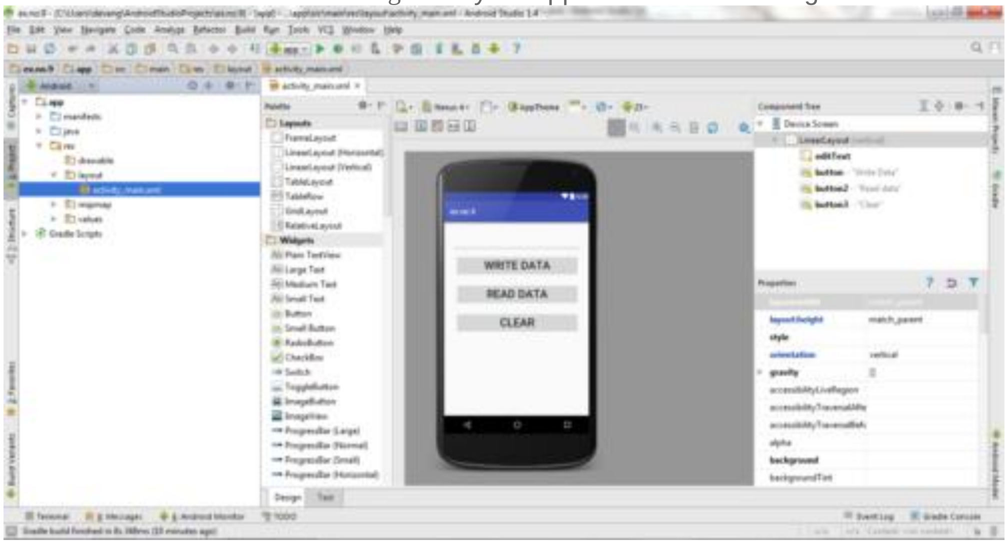
- Then delete the code which is there and type the code as given below.

Code for 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="20dp"
    android:orientation="vertical">

    <EditText
        android:id="@+id/editText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:singleLine="true"
        android:textSize="30dp" />

    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:text="Write Data"
        android:textSize="30dp" />

    <Button
        android:id="@+id/button2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:text="Read data"
        android:textSize="30dp" />

    <Button
        android:id="@+id/button3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:text="Clear"
        android:textSize="30dp" />

</LinearLayout>
```

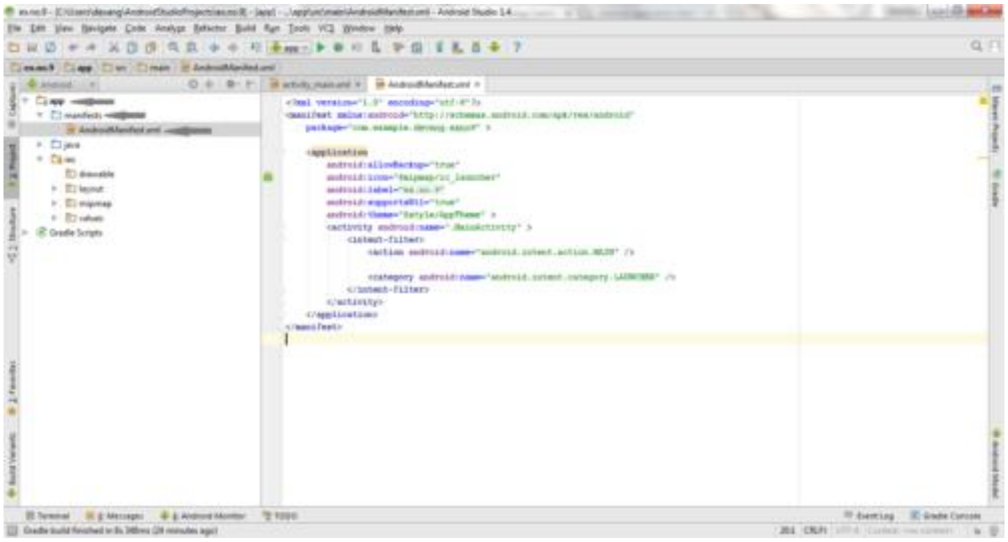
- Now click on Design and your application will look as given below.



- So now the designing part is completed.

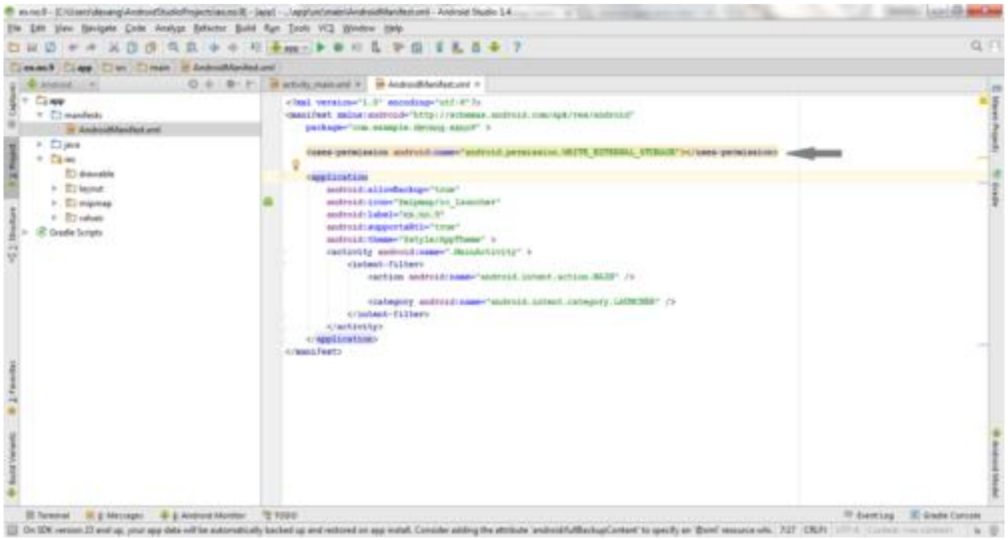
Adding permissions in Manifest for the Android Application:

- Click on app -> manifests -> AndroidManifest.xml



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.devang.exno9" >

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="ex.no.9"
        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>
```

- Now include the WRITE_EXTERNAL_STORAGE permissions in the AndroidManifest.xml file as shown below



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.devang.exno9" >

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="ex.no.9"
        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>
```

Code for AndroidManifest.xml:

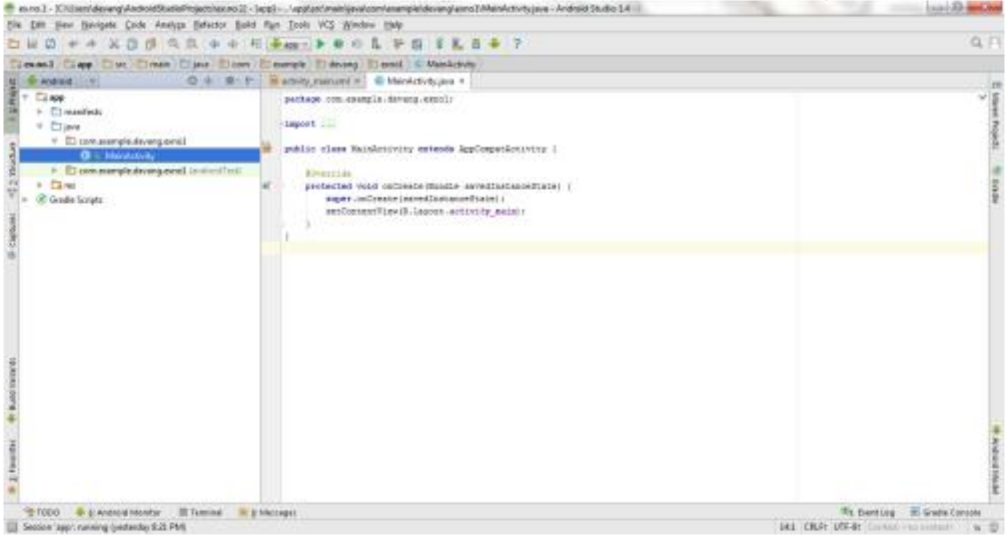
?

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.exno9" >

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

    <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>
```

- So now the Permissions are added in the Manifest.

Java Coding for the Android Application:

- Click on app -> java -> com.example.exno9 -> MainActivity.



```
package com.example.devang.exno1;

import ...

public class MainActivity extends AppCompatActivity {

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

- Then delete the code which is there and type the code as given below.

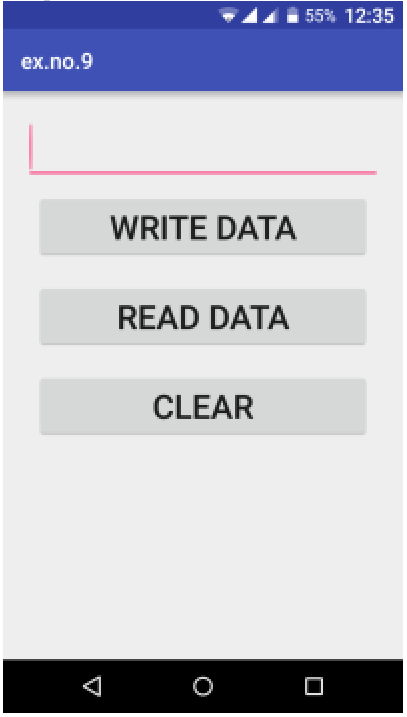
Code for MainActivity.java:

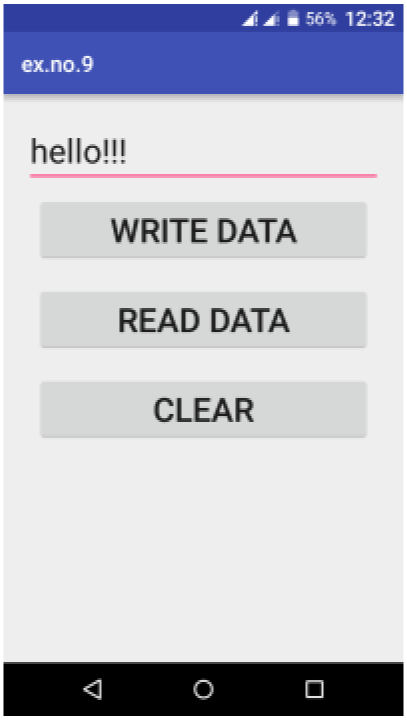
?

```
package com.example.exno9;

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

import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStreamReader;

public class MainActivity extends AppCompatActivity
{
    EditText e1;
    Button write,read,clear;
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        e1= (EditText) findViewById(R.id.editText);
        write= (Button) findViewById(R.id.button);
        read= (Button) findViewById(R.id.button2);
        clear= (Button) findViewById(R.id.button3);

        write.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                String message=e1.getText().toString();
                try
                {
                    File f=new File("/sdcard/myfile.txt");
                    f.createNewFile();
                    FileOutputStream fout=new FileOutputStream(f);
                    fout.write(message.getBytes());
                    fout.close();
                    Toast.makeText(getBaseContext(),"Data Written in
SDCARD",Toast.LENGTH_LONG).show();
                }
                catch (Exception e)
                {
                    Toast.makeText(getBaseContext(),e.getMessage(),Toast.LENGTH_
LONG).show();
                }
            }
        });

        read.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                String message;
                String buf = "";
                try
                {
                    File f = new File("/sdcard/myfile.txt");
                    FileInputStream fin = new FileInputStream(f);
                    BufferedReader br = new BufferedReader(new
InputStreamReader(fin));
                    while ((message = br.readLine()) != null)
                    {
                        buf += message;
                    }
                    e1.setText(buf);
                    br.close();
                    fin.close();
                    Toast.makeText(getBaseContext(),"Data Recived from
SDCARD",Toast.LENGTH_LONG).show();
                }
                catch (Exception e)
                {
                    Toast.makeText(getBaseContext(), e.getMessage(),
Toast.LENGTH_LONG).show();
                }
            }
        });

        clear.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                e1.setText("");
            }
        });
    }
}
```

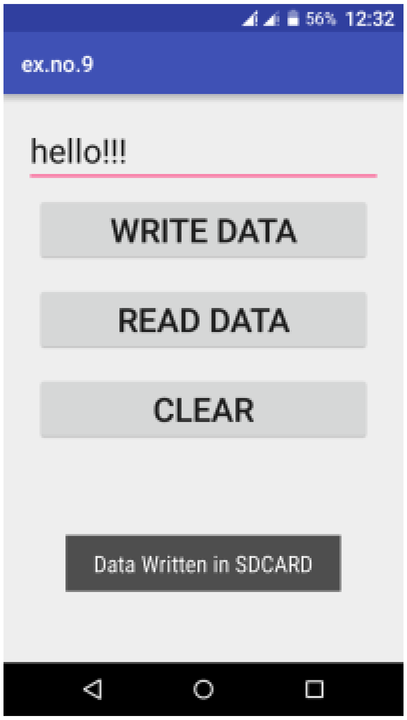
- So now the Coding part is also completed.
- Now run the application to see the output.

Output:



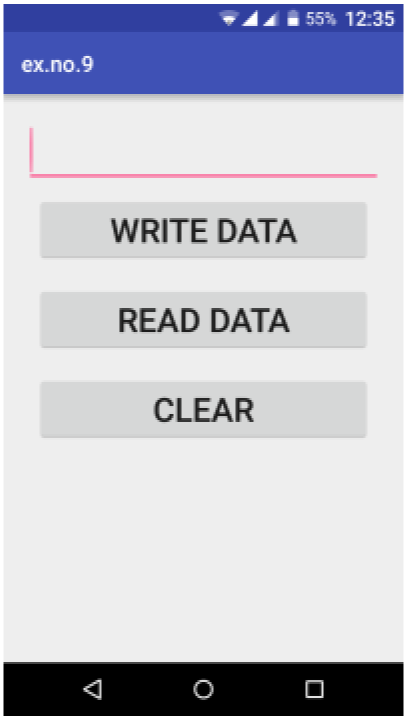


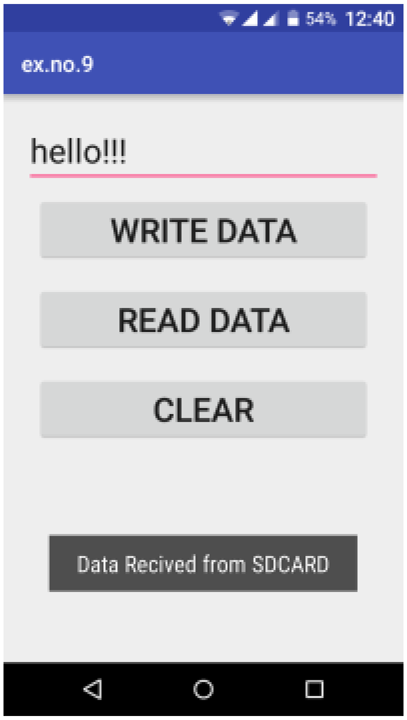
hello!!!



hello!!!

Data Written in SDCARD





hello!!!

Data Recived from SDCARD

Result:

Thus Android Application that writes data to the SD Card is developed and executed successfully.