Android Application that writes data to the SD Card

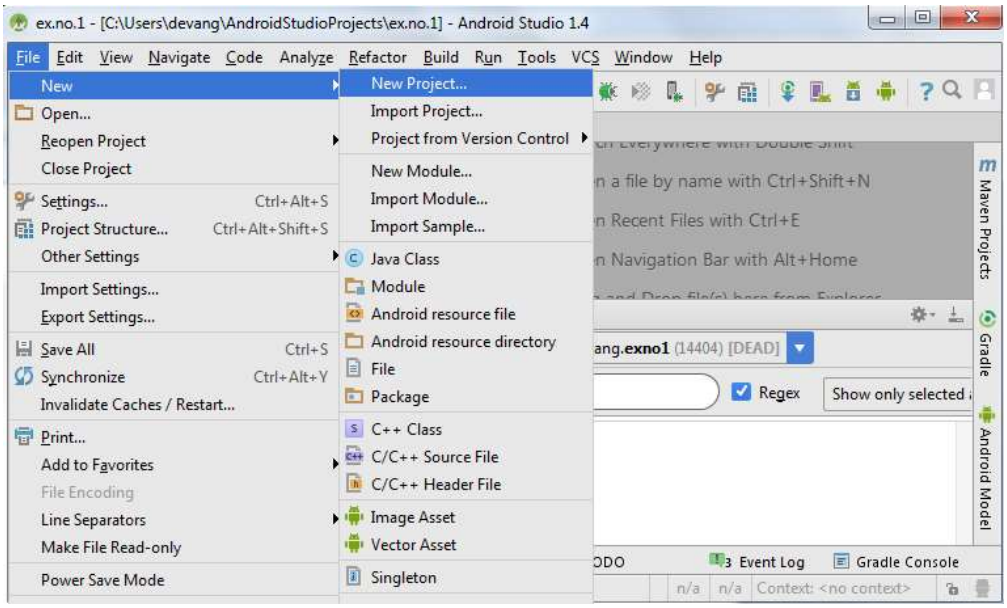
Aim:

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

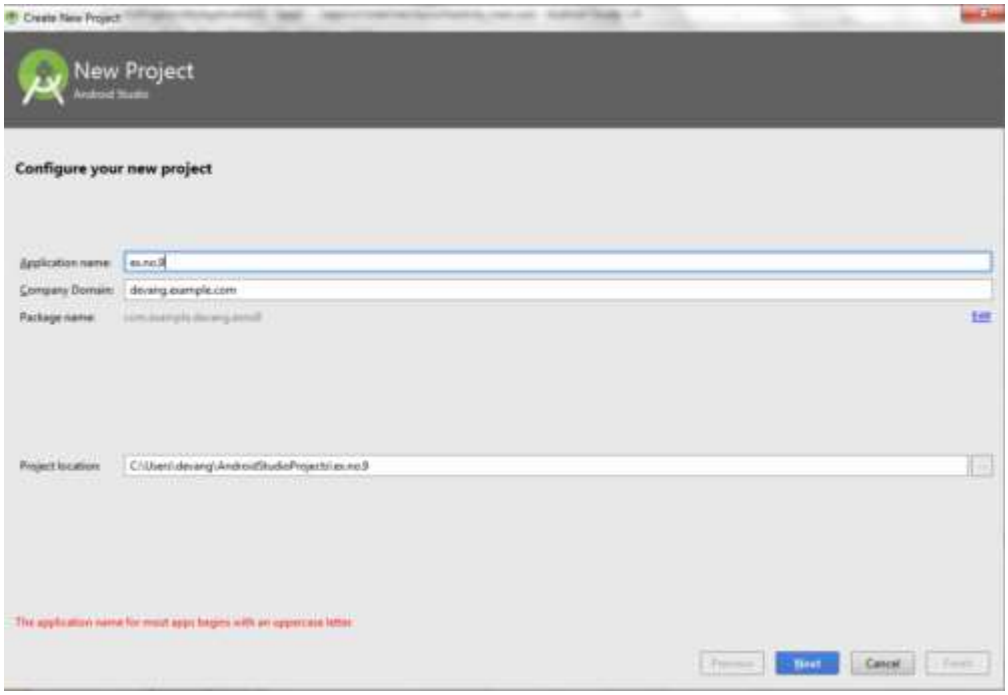
Procedure:

Creating a New project:

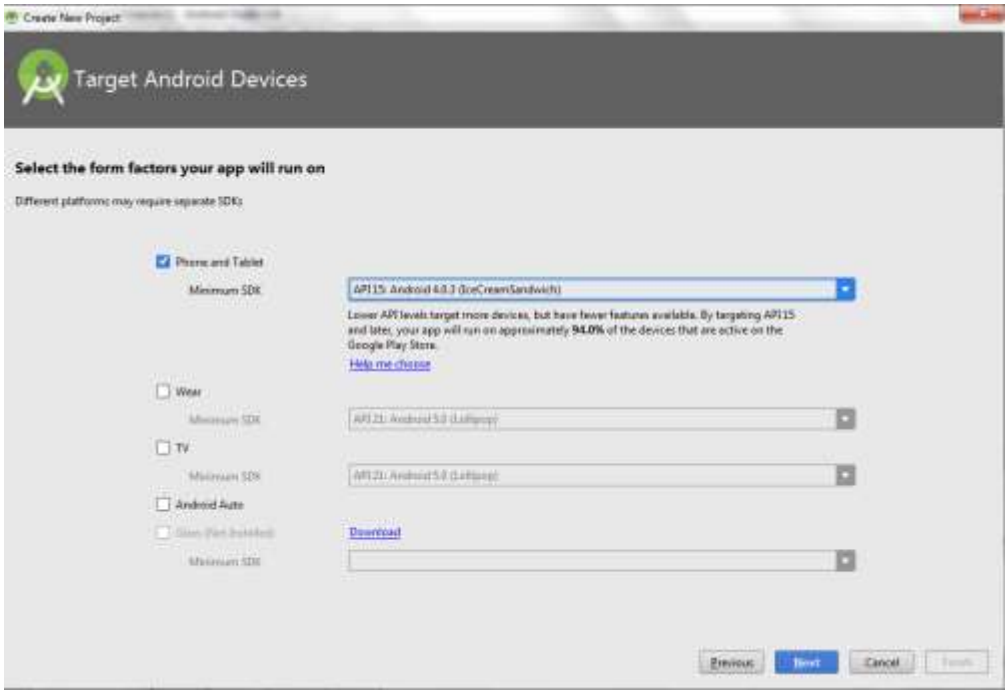
- Open Android Studio and then click on **File -> New -> New project.**



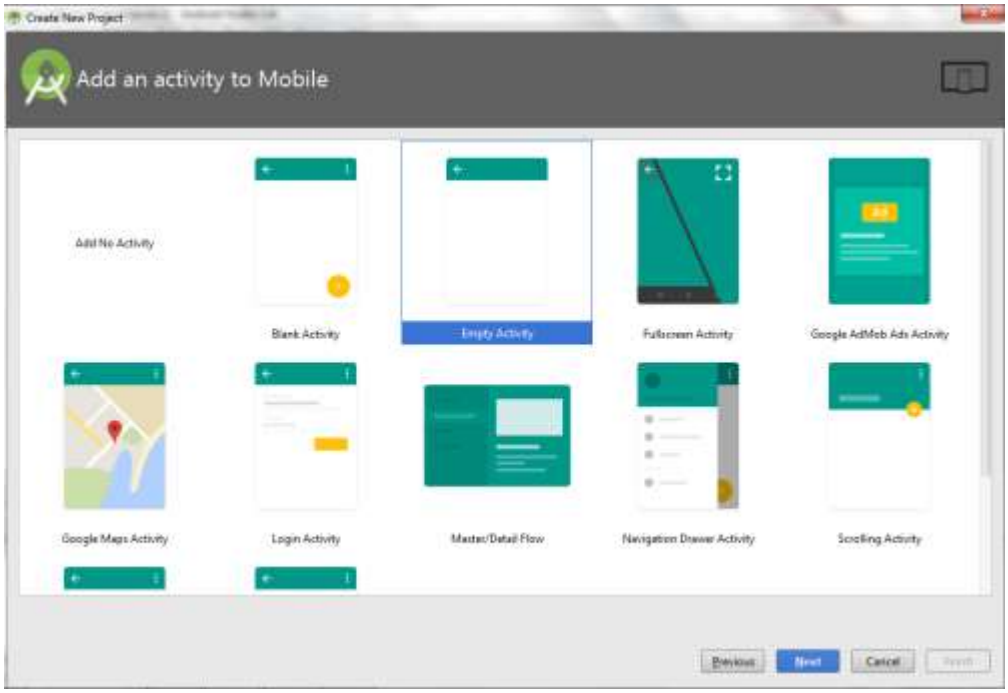
- Then type the Application name as "**ex.no.9"** and click **Next.**



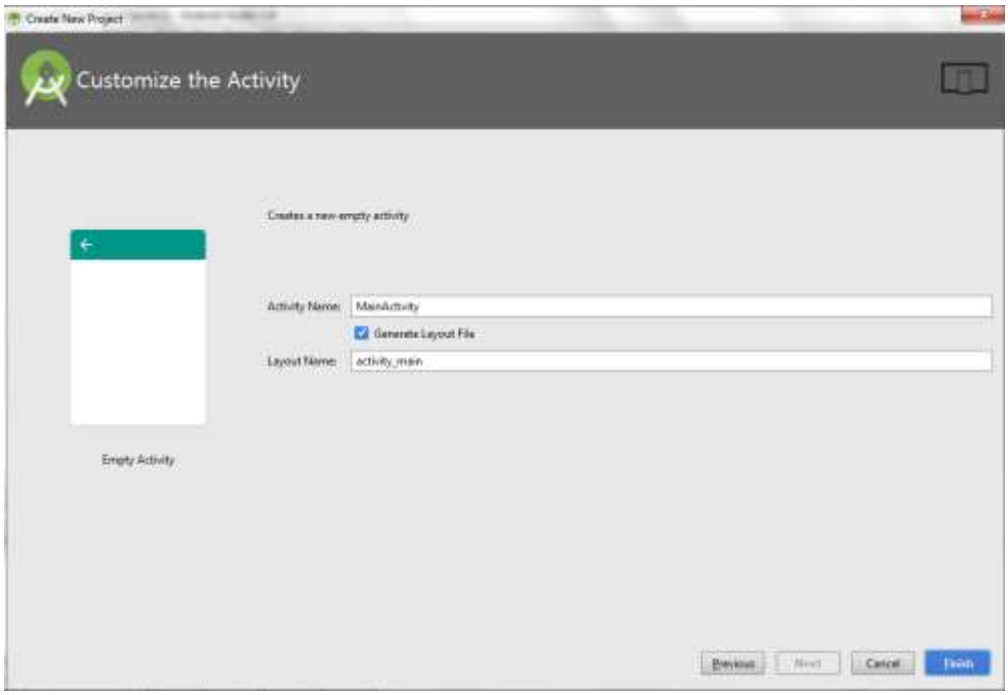
- Then select the **Minimum SDK** as shown below and click **Next**.



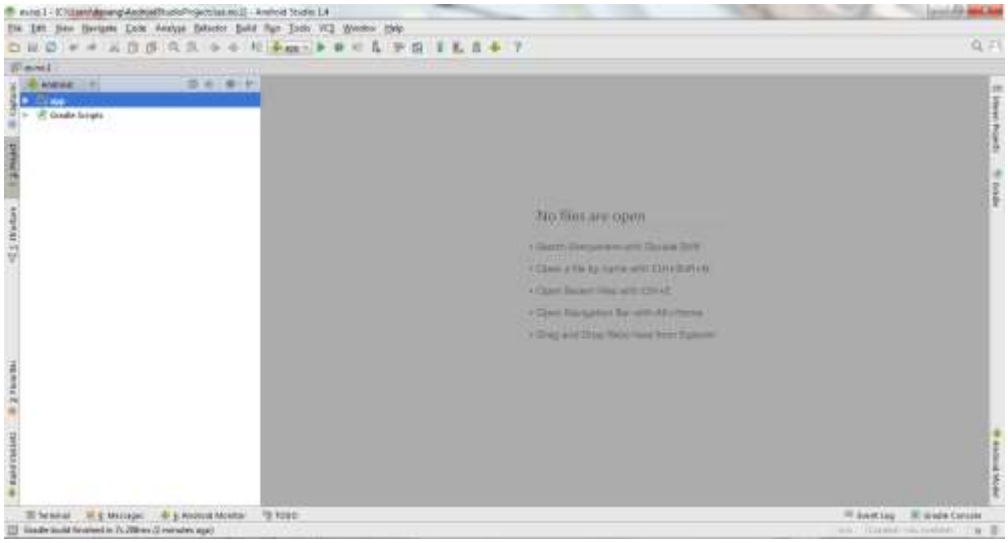
- Then select the **Empty Activity** and click **Next.**



- Finally click F**inish**.

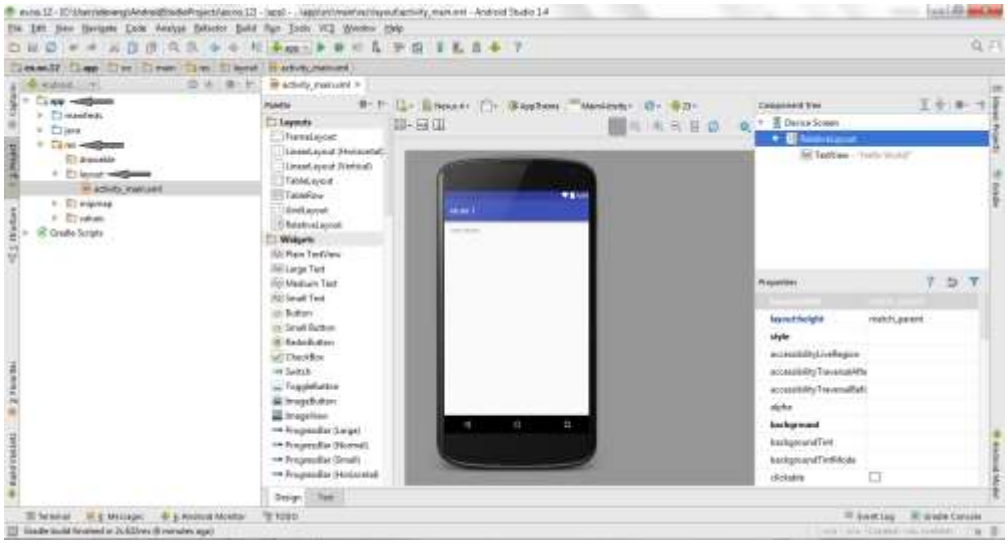


- It will take some time to build and load the project.
- After completion it will look as given below.

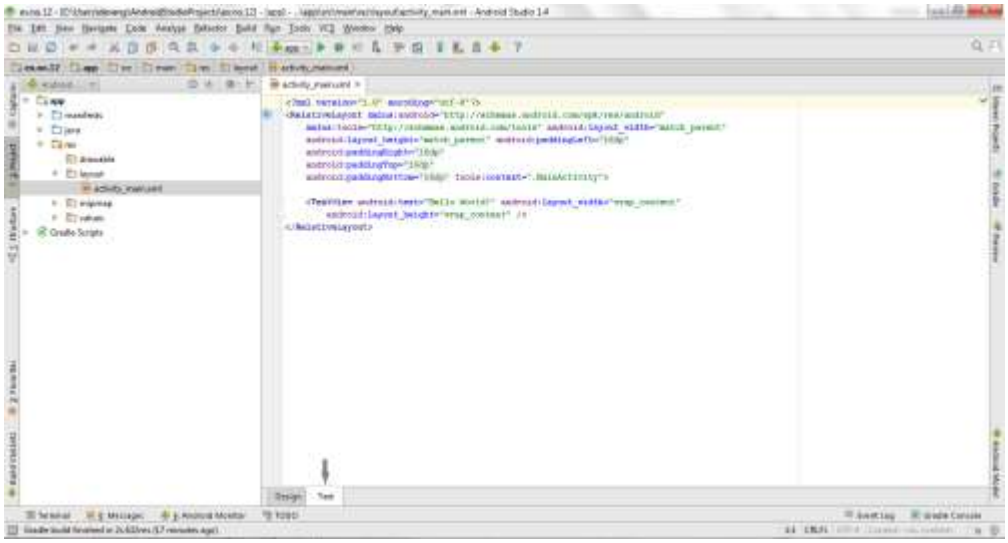


Designing layout for the Android Application:

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



- Now click on **Text** as shown below.



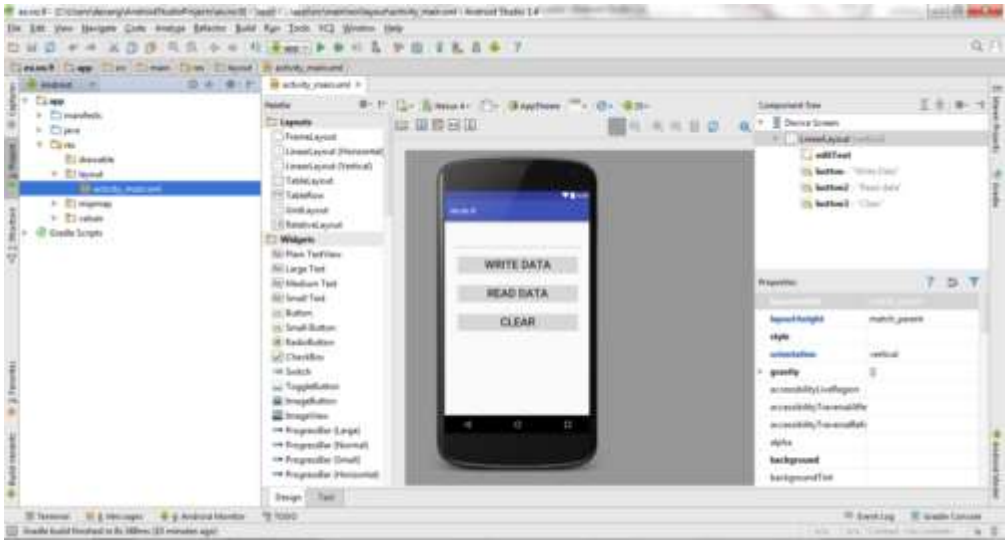
- Then delete the code which is there and type the code as given below.

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"
```

```
30          android:text="Clear"
31          android:textSize="30dp" />
32
33  </LinearLayout>
34
35
36
37
38
39
```

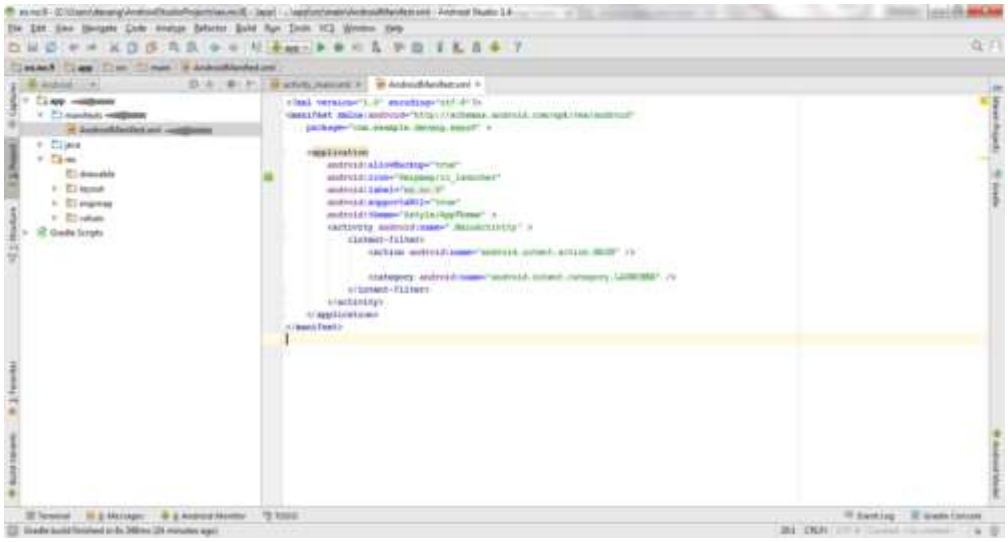
- Now click on **Design** and your application will look as given below.



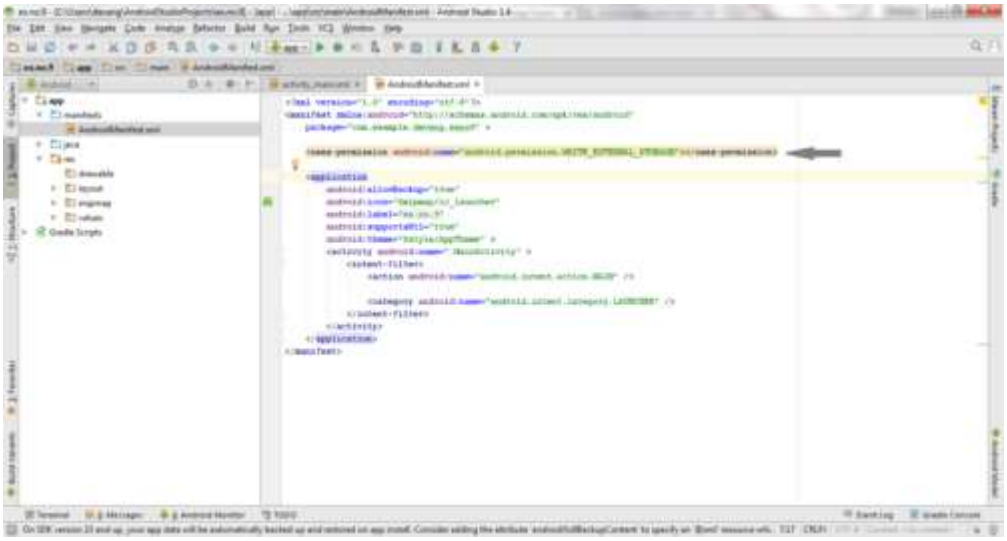
- So now the designing part is completed.

Adding permissions in Manifest for the Android Application:

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



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



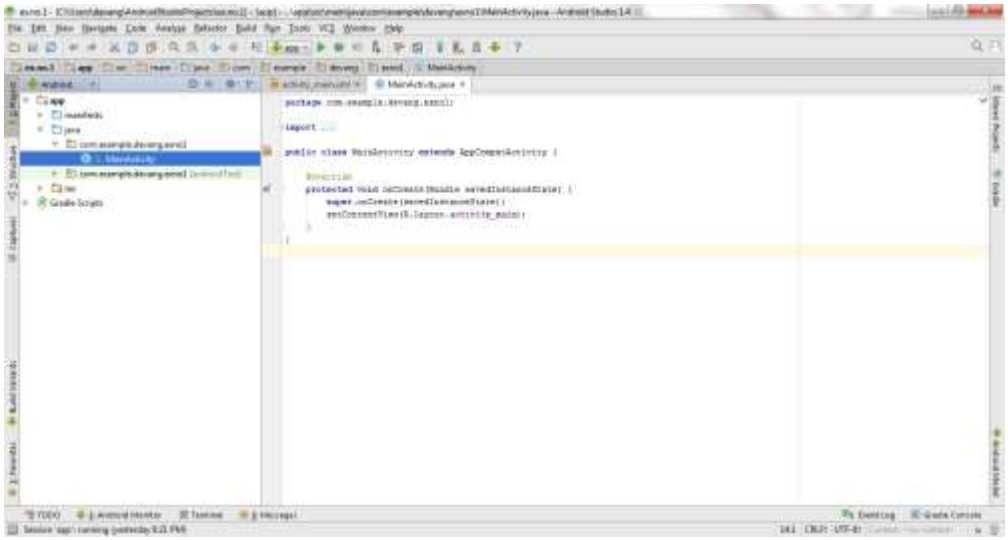
Code for AndroidManifest.xml:

```
?
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="com.example.exno9" >
4
5      <uses-permission
6  android:name="android.permission.WRITE_EXTERNAL_STORAGE"></uses-
7  permission>
8
9      <application
10         android:allowBackup="true"
11         android:icon="@mipmap/ic_launcher"
12         android:label="@string/app_name"
13         android:supportsRtl="true"
14         android:theme="@style/AppTheme" >
15         <activity android:name=".MainActivity" >
16             <intent-filter>
17                 <action android:name="android.intent.action.MAIN" />
18
19                 <category android:name="android.intent.category.LAUNCHER"
20 />
21             </intent-filter>
22         </activity>
23     </application>
24 </manifest>
```

- So now the Permissions are added in the Manifest.

Java Coding for the Android Application:

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



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

Code for MainActivity.java:

```
?
1  package com.example.exno9;
2
3  import android.os.Bundle;
4  import android.support.v7.app.AppCompatActivity;
5  import android.view.View;
6  import android.widget.Button;
7  import android.widget.EditText;
8  import android.widget.Toast;
9
10 import java.io.BufferedReader;
11 import java.io.File;
12 import java.io.FileInputStream;
13 import java.io.FileOutputStream;
14 import java.io.InputStreamReader;
15
16 public class MainActivity extends AppCompatActivity
17 {
18     EditText e1;
19     Button write,read,clear;
20     @Override
21     protected void onCreate(Bundle savedInstanceState)
22     {
23         super.onCreate(savedInstanceState);
24         setContentView(R.layout.activity_main);
25
26         e1= (EditText) findViewById(R.id.editText);
27         write= (Button) findViewById(R.id.button);
28         read= (Button) findViewById(R.id.button2);
29         clear= (Button) findViewById(R.id.button3);
30
31         write.setOnClickListener(new View.OnClickListener()
32         {
33             @Override
34             public void onClick(View v)
35             {
```

```
                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("");
            }
        });
    }
}
```

4
5
4
6
4
7
4
8
4
9
5
0
5
1
5
2
5
3
5
4
5
5
5
6
5
7
5
8
5
9
6
0
6
1
6
2
6
3
6
4
6
5
6
6
6
7
6
8
6
9

7
0
7
1
7
2
7
3
7
4
7
5
7
6
7
7
7
8
7
9
8
0
8
1
8
2
8
3
8
4
8
5
8
6
8
7
8
8
8
9
9
0

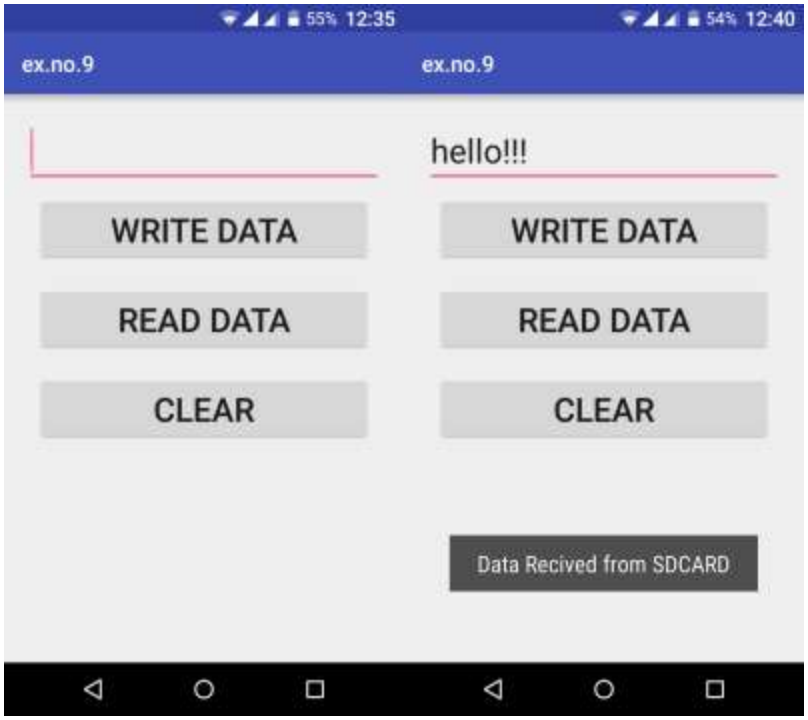
- So now the Coding part is also completed.
- Now run the application to see the output.

Output:


55% 12:35
ex.no.9
WRITE DATA
READ DATA
CLEAR


56% 12:32
ex.no.9
hello!!!
WRITE DATA
READ DATA
CLEAR


56% 12:32
ex.no.9
hello!!!
WRITE DATA
READ DATA
CLEAR
Data Written in SDCARD



Result:

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