LAB MANUAL

Lab Name : Mobile Application Development Lab

Lab Code : 6CS4-24

Branch : Computer Science and Engineering

Year : 3rd Year/6th Semester

Jaipur Engineering College and Research Center, Jaipur

Department of Computer Science and Engineering

(Rajasthan Technical University, KOTA)

INDEX

1. VISION & MISSION

VISION: To become renowned Centre of excellence in computer science and engineering and make competent engineers & professionals with high ethical values prepared for lifelong learning.

MISSION:

M1: To impart outcome based education for emerging technologies in the field of computer science and engineering.
M2: To provide opportunities for interaction between academia and industry.
M3: To provide platform for lifelong learning by accepting the change in technologies.
M4: To develop aptitude of fulfilling social responsibilities.

2. PEO

1. To provide students with the fundamentals of Engineering Sciences with more emphasis in Computer Science & Engineering by way of analyzing and exploiting engineering challenges.

2. To train students with good scientific and engineering knowledge so as to comprehend, analyze, design, and create novel products and solutions for the real life problems.

3. To inculcate professional and ethical attitude, effective communication skills, teamwork skills, multidisciplinary approach, entrepreneurial thinking and an ability to relate engineering issues with social issues.

4. To provide students with an academic environment aware of excellence, leadership, written ethical codes and guidelines, and the self-motivated life-long learning needed for a successful professional career in Computer Science & Engineering.

5. To prepare students to excel in Industry and Higher education by Educating Students along With High moral values and Knowledge in Computer Science & Engineering.

3. PROGRAM OUTCOMES

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and Computer Science & Engineering specialization to the solution of complex Computer Science & Engineering problems.
2. Problem analysis: Identify, formulate, research literature, and analyze complex Computer Science and Engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. Design/development of solutions: Design solutions for complex Computer Science and Engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of Computer Science and Engineering experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex Computer Science Engineering activities with an understanding of the limitations.
6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional Computer Science and Engineering practice.
7. Environment and sustainability: Understand the impact of the professional Computer Science and Engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the Computer Science and Engineering practice.
9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings in Computer Science and Engineering.
10. Communication: Communicate effectively on complex Computer Science and Engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. Project management and finance: Demonstrate knowledge and understanding of the Computer Science and Engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. Life-long learning: Recognize the need for, and have the preparation andability to engage in independent and life-long learning in the broadest contextof technological changein Computer Science and Engineering.

4. COURSE OUTCOMES

Students will be able to:

CO1: Ability to apply general programming knowledge in the field of developing mobile applications.
CO2: Understanding of the specific requirements, possibilities and challenges when developing for a mobile context.

5. MAPPING OF CO & PO

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
1	3	2	3	2	2	1	1	1	2	2	3	2
2	3	2	2	2	2	1	1	1	2	2	3	2

[L=1, M=2, H=3]

6. SYLLABUS

RAJASTHAN TECHNICAL UNIVERSITY, KOTA

Syllabus

III Year-VI Semester: B.Tech. Computer Science and Engineering

6CS4-24: Mobile Application Development Lab

Credit: 1.5 **Max. Marks: 75(IA:45, ETE:30)**

0L+0T+3P **End Term Exam: 2 Hours**

SN	List of Experiments
1	To study Android Studio and android studio installation. Create "Hello World" application.
2	To understand Activity, Intent, Create sample application with login module.(Check username and password).
3	Design simple GUI application with activity and intents e.g. calculator.
4	Develop an application that makes use of RSS Feed.
5	Write an application that draws basic graphical primitives on the screen
6	Create an android app for database creation using SQLite Database.
7	Develop a native application that uses GPS location information
8	Implement an application that writes data to the SD card.
9	Design a gaming application
10	Create an application to handle images and videos according to size.

INSTRUCTIONAL METHODS:-

Direct Instructions:

- Black board presentation

Interactive Instruction:

- Coding

Indirect Instructions:

- Problem solving

LEARNING MATERIALS:-

Text/Lab Manual

ASSESSMENT OF OUTCOMES:-

- End term Practical exam (Conducted by RTU, KOTA)
- Daily Lab interaction.

OUTCOMES WILL BE ACHIEVED THROUGH FOLLOWING:-

1. Lab Teaching (through chalk and board).
2. Discussion on website work

INSTRUCTIONS OF LAB

DO's

- Please switch off the Mobile/Cell phone before entering Lab.
- Enter the Lab with complete source code and data.
- Check whether all peripheral are available at your desktop before proceeding for program.
- Intimate the Lab in Charge whenever you are incompatible in using the system or in case software get corrupted/ infected by virus.
- Arrange all the peripheral and seats before leaving the lab.
- Properly shutdown the system before leaving the lab.

- Keep the bag outside the laboratory.
- Enter the lab on time and leave at proper time.
- Maintain the decorum of the lab.
- Utilize lab hours in the corresponding experiment.

DON'TS

- Don't mishandle the system.
- Don't leave the system on standing for long time.
- Don't make noise in the lab.
- Don't bring the mobile in the lab. If extremely necessary, then keep ringers off.
- Don't enter in the lab without permission of lab in charge.
- Don't litter in the lab.
- Don't delete or make any modification in system files.
- Don't carry any lab equipment outside the lab.

INSTRUCTIONS FOR STUDENT

BEFORE ENTERING IN THE LAB

- All the students are supposed to prepare the theory regarding the next program.
- Students are supposed to bring the practical file and required notes.
- Previous programs should be written in the practical file.
- Any student not following these instructions will be denied entry in the lab.

EXPERIMENT-1

Create “Hello World” application.

AIM: Designan application of ―Hello World‖

PROCEDURE:

1. Open Eclipse IDE.
2. Create the project Ex_No_1.
3. Go to package explorer in the left hand side. Select the project Ex_No_1.
4. Go to res folder and select layout. Double click the activity_main.xml file.
5. Now you can see the Graphical layout window.
6. Drag and drop the Text view.
7. Again go to package explorer in the left hand side. Select the project Ex_No_1.
8. Go to src folder. Double click the MainActivity.java file.
9. In java file write the activities done by the application such as, actions of button.
10. Finally run the Android application

PROGRAM:

Acitivity_main.xml:

```
?xml version="1.0"encoding="utf-
8"?><androidx.constraintlayout.widget.ConstraintLayout
        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"
        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"/>

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

MainActivity.java

```
package com.example.myapplication;

import androidx.appcompat.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);
```

VIVA QUESTIONS

1. **What is Android?**
   Android is the open source, Linux-based operating system used in mobiles, tablets, televisions, etc.
2. **Who is the founder of Android?** Andy Rubin
3. **What are the code names of android?**
   - Aestro
   - Blender
   - Cupcake
   - Donut
   - Éclair
   - Froyo
   - Gingerbread
   - Honeycomb
   - Ice cream sandwich
   - Jelly Bean
   - Kitkat
   - Lollipop
   - Marshmallow
   - Nougat
   - Oreo
4. **What is the Google Android SDK?**
   The Google Android SDK is a toolset that developers need in order to write apps on Android enabled devices. It contains a graphical interface that emulates an Android driven handheld environment, allowing them to est and debug their codes.
5. **Does android support other language than java?**
   Yes, an android app can be developed in C/C++ also using android NDK(Native Development Kit). It makes the performance faster. It should be used with Android SDK.

EXPERIMENT-2

To understand Activity, Intent, Create sample application with login module.(Check username and password).

AIM: Design To understand Activity, Intent, Create sample application with login module.(Check usernameand password).

PROCEDURE:

Open Eclipse IDE.
2. Create the project Ex_No_2.
3. Go to package explorer in the left hand side. Select the project Ex_No_3.
4. Go to res folder and select layout. Double click the activity_main.xml file.
5. Now you can see the Graphical layout window.
6. Drag and drop the following components:
    a. Two EditTexts with hints. Enter the first number and enter the second number
    b. Four Buttons with labeled as ADD, SUB, MUL and DIV
7. Again go to package explorer in the left hand side. Select the project Ex_No_3.
8. Go to src folder. Double click the MainActivity.java file.
9. In java file write the activities done by the application such as, actions of button.
10. Finally run the Android application

PROGRAM

Activity_main.xml

```
<?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"
    tools:context=".MainActivity" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="56dp"
        android:text="@string/Username"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <EditText
        android:id="@+id/editText1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBottom="@+id/textView1"
        android:layout_alignParentRight="true"
```

```
        android:ems="10"
        android:inputType="text" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignRight="@+id/textView1"
        android:layout_below="@+id/textView1"
        android:layout_marginTop="48dp"
```

```
        android:text="@string/Password"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <EditText
        android:id="@+id/editText2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/textView2"
        android:layout_alignBottom="@+id/textView2"
        android:layout_alignLeft="@+id/editText1"
        android:ems="10"
        android:inputType="textPassword" >

        <requestFocus />
    </EditText>

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/editText2"
        android:layout_marginTop="68dp"
        android:layout_toLeftOf="@+id/editText2"
        android:text="@string/Login" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/button1"
        android:layout_alignBottom="@+id/button1"
        android:layout_alignLeft="@+id/editText2"
        android:layout_marginLeft="42dp"
        android:text="@string/Cancel"/>
```

Activity_Second.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:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:text="@string/Secondform"
        tools:context=".Second" />
```

```
    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/textView1"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="22dp"
        android:text="@string/Logout" />

</RelativeLayout>
```

MainActivity.java

```
package com.example.loginform;

import android.os.Bundle;
```

```
import android.annotation.SuppressLint;
import android.app.Activity;
import android.content.Intent;
import android.view.*;
import android.view.View.OnClickListener;
import android.widget.*;

public class MainActivity extends Activity implements OnClickListener{

    EditText name;
    EditText pass;
    Button login;
    Button cancel;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        name=(EditText )findViewById(R.id.editText1);
        pass=(EditText )findViewById(R.id.editText2);
        login=(Button )findViewById(R.id.button1);
        cancel=(Button )findViewById(R.id.button2);
        login.setOnClickListener(this);
        cancel.setOnClickListener(this);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }

    @SuppressLint("ShowToast") @Override
    public void onClick(View v) {
        String na=name.getText().toString();
        String pa=pass.getText().toString();
        switch(v.getId())
        {
            case R.id.button1:
                if(na.equals("chaitu") && pa.equals("root") || pa.equals("admin")){ Intent i
                    = new Intent(this,Second.class);
                    i.putExtra("c1", na);
                    startActivity(i);
                }
                else
                {
                        Toast.makeText(this,"check #username or #Password", 3600).show();
                }
```

```
                break;

            case R.id.button2:
                name.setText("");
                pass.setText("");
                break;

            default:

                break;
        }

    }

}
```

Second.java

```
package com.example.loginform;
import com.example.loginform.R;
```

```
import android.os.Bundle;
import android.app.Activity;
import android.view.*;
import android.view.View.OnClickListener;
import android.widget.*;
import android.content.*;

public class Second extends Activity implements OnClickListener{

    Button Logout;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second); Intent
        i=getIntent();
        String str = i.getStringExtra("c1");
        TextView t1 = (TextView )findViewById(R.id.textView1);
        t1.setText("Logged In Successfully ! "+str); Logout=(Button
        )findViewById(R.id.button1);
        Logout.setOnClickListener(this);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_second, menu);
        return true;
    }

    @Override
    public void onClick(View v1) {
        switch(v1.getId())
        {
            case R.id.button1:
                Intent i = new Intent(this,MainActivity.class);
                startActivity(i);
                finish();
                System.exit(0);
                break;

            default:
                break;
        }

    }
```

VIVA QUESTIONS

1. **What is the Android Architecture?**
   Android Architecture is made up of 4 key components:
   - Linux Kernel
   - Libraries
   - Android Framework
   - Android Applications
2. **What is Activity in Android?**
   Activity is like a frame or window in java that represents GUI. It represents one screen of android.
3. **What are the life cycle methods of android activity?** There are 7 life-cycle methods of activity. They are as follows:
   1. onCreate()
   2. onStart()
   3. onResume()
   4. onPause()
   5. onStop()
   6. onRestart()
   7. onDestroy()
4. **What is intent?**
   It is a kind of message or information that is passed to the components. It is used to launch an activity, display a web page, send SMS, send email, etc. There are two types of intent in android:
   1. Implicit Intent
   2. Explicit Intent
5. **How is view elements identified in the android program?**
   View elements can be identified using the keyword findViewById.

EXPERIMENT-3

Design simple GUI application with activity and intents e.g. calculator

AIM: Design simple GUI application with activity and intents e.g. calculator

PROCEDURE:

1. Open Eclipse IDE.
2. Create the project Ex_No_3.
3. Go to package explorer in the left hand side. Select the project Ex_No_3.
4. Go to res folder and select layout. Double click the activity_main.xml file.
5. Now you can see the Graphical layout window.
6. Drag and drop the following components:
    a. Two EditTexts with hints. Enter the first number and enter the second number
    b. Four Buttons with labeled as ADD, SUB, MUL and DIV
7. Again go to package explorer in the left hand side. Select the project Ex_No_3.
8. Go to src folder. Double click the MainActivity.java file.
9. In java file write the activities done by the application such as, actions of button.
10. Finally run the Android application

PROGRAM:

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"
        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.ex_no_3.MainActivity">
<EditText
        android:id="@+id/editText1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:ems="10"
        android:hint="Enter the first number"
        tools:ignore="TextFields,HardcodedText">
<requestFocus />
</EditText>
<EditText
        android:id="@+id/editText2"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_below="@+id/editText1"
        android:ems="10"
        android:hint="Enter the second number"
        tools:ignore="TextFields,HardcodedText" />
<Button
        android:id="@+id/button4"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_below="@+id/button3"
        android:text="DIV"
        tools:ignore="HardcodedText" />
<Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_below="@+id/editText2"
        android:text="ADD"
        tools:ignore="HardcodedText" />
<Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_below="@+id/button1"
        android:text="SUB"
        tools:ignore="HardcodedText" />
<Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_below="@+id/button2"
        android:text="MUL"
        tools:ignore="HardcodedText" />
<TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/button4"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="22dp"
        android:text=""
        android:textAppearance="?android:attr/textAppearanceLarge" />
</RelativeLayout>
```

MainActivity.java:

```
package com.example.ex_no_3;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle; import
android.view.View;

import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
public class MainActivity extends ActionBarActivity {
        int n1,n2;
        float num1,num2;
        @Override
        protected void onCreate(Bundle savedInstanceState) {
```

```
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
final EditText e1=(EditText)findViewById(R.id.editText1);
final EditText
e2=(EditText)findViewById(R.id.editText2);Button
b1=(Button)findViewById(R.id.button1); Button
b2=(Button)findViewById(R.id.button2);
Button b3=(Button)findViewById(R.id.button3);
Button b4=(Button)findViewById(R.id.button4);
final TextView
t=(TextView)findViewById(R.id.textView1);b1.setOnClick
Listener(
new OnClickListener()
{
@Override
public void onClick(View arg0) {
        // TODO Auto-generated method stub
        n1=Integer.parseInt(e1.getText().toString());
        n2=Integer.parseInt(e2.getText().toString());
        t.setText(e1.getText().toString()+"+"+e2.getText().toString()+" = "+(n1+n2)); }
});
b2.setOnClickListener(
        new OnClickListener()
        {
                @Override
                public void onClick(View arg0) {
                        // TODO Auto-generated method stub
                        n1=Integer.parseInt(e1.getText().toString());
                        n2=Integer.parseInt(e2.getText().toString());
                        t.setText(e1.getText().toString()+"-"+e2.getText().toString()+"= "+(n1-
                        n2));
                }
        });
b3.setOnClickListener(
        new OnClickListener()
        {
                @Override
                public void onClick(View arg0) {
                        //   TODO Auto-generated method
                stubn1=Integer.parseInt(e1.getText().toString());
                n2=Integer.parseInt(e2.getText().toString());
                t.setText(e1.getText().toString()+"*"+e2.getText().toString()+" =
                "+(n1*n2));
                }
        });
b4.setOnClickListener(
        new OnClickListener()
        {
                @Override
```

```
                    public void onClick(View arg0) {
                        // TODO Auto-generated method stub
                        num1=Float.parseFloat(e1.getText().toString());
                        num2=Float.parseFloat(e2.getText().toString());
                        t.setText(e1.getText().toString()+"/"+e2.getText().toString()+" =
                        "+(num1/num2));
                    }
                });
            }
        }
```

OUTPUT


10:13
Ex_No_3
7
5
ADD
SUB
MUL
DIV
7 + 5 = 12
10:13
Ex_No_3
7
5
ADD
SUB
MUL
DIV
7 - 5 = 2
10:14
Ex_No_3
7
5
ADD
SUB
MUL
DIV
7 * 5 = 35
10:14
Ex_No_3
7
5
ADD
SUB
MUL
DIV
7 / 5 = 1.4

VIVA QUESTIONS

1. **Define Android toast.**
   An android toast provides feedback to the users about the operation being performed by them. It displays the message regarding the status of operation initiated by the user.
2. **Explain the use of „bundle‟ in android?**
   We use bundles to pass the required data to various subfolders.
3. **What is an application resource file?**
   The files which can be injected for the building up of a process are called as application resource file.
4. **How are layouts placed in Android?**
   Layouts in Android are placed as XML
   files.
5. **What is implicit intent in Android?**
   The Implicit intent is used to invoke the system components.

EXPERIMENT-4

Develop an application that makes use of RSS Feed.

AIM: Develop an application that makes use of RSS Feed.

PROCEDURE:

1. Open Eclipse IDE.
2. Create the project Ex_No_4.
3. Go to package explorer in the left hand side. Select the project Ex_No_4.
4. Go to res folder and select layout. Double click the activity_main.xml file.
5. Now you can see the Graphical layout window.
6. Create the FrameLayout.
7. Create a new layout named as fragment_layout.xml which has following components:
    a. ListView
    b. ProgressBar
8. Create another one layout named as rss_item.xml which has only one TextView.
9. Again go to package explorer in the left hand side. Select the project Ex_No_4.
10. Go to src folder. Double click the MainActivity.java file.
11. In java file write the activities done by the application.
12. Create the following additional classes for this application:
    a. Constants.java
    b. PcWorldRssParser.java
    c. RssAdapter.java
    d. RssFragement.java
    e. RssItem.java
    f. RssService.java
13. Write appropriate actions for the created additional classes.
14. Get the following permission in AndroidManifest.xml file:

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

15. Finally run the android application.

PROGRAM:

activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:id="@+id/fragment_container"
    android:layout_height="fill_parent" />
```

fragement_layout.xml:

```
<?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"
    android:orientation="vertical">
```

```
<ListView
        android:id="@+id/listView"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent">
```

```
</ListView>
<ProgressBar
        android:id="@+id/progressBar"
        style="?android:attr/progressBarStyleLarge"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true" />
</RelativeLayout>
```

rss_item.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        android:id="@+id/itemTitle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="18dp"
        tools:ignore="SpUsage" />
```

MainActivity.java:

```
package com.example.ex_no_4;
import android.os.Bundle;
import android.support.v4.app.FragmentActivity;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentTransaction;
public class MainActivity extends FragmentActivity {
        @Override
        public void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);setConte
                ntView(R.layout.activity_main);
                if (savedInstanceState == null) {
                        addRssFragment();
                }
        }
        private void addRssFragment() {
                FragmentManager manager = getSupportFragmentManager();
                FragmentTransaction transaction = manager.beginTransaction();
                RssFragment fragment = new RssFragment();
                transaction.add(R.id.fragment_container, fragment);
                transaction.commit();
        }
        @Override
        protected void onSaveInstanceState(Bundle outState) {
                super.onSaveInstanceState(outState);outState.put
                Boolean("fragment_added", true);
        }
```

```
}
```

Constants.java

```
package com.example.ex_no_4;
public class Constants {
        public static final String TAG="RssApp";
}
```

PcWorldRssParser.java

```
package com.example.ex_no_4;
import java.io.IOException;
```

```
import java.io.InputStream;
import java.util.ArrayList;
import java.util.List;
import org.xmlpull.v1.XmlPullParser;
import org.xmlpull.v1.XmlPullParserException;
import android.util.Xml;
public class PcWorldRssParser {
	//   We don't use namespaces
	private final Stringns= null;
	public List<RssItem> parse(InputStream inputStream) throws XmlPullParserException,IOException
	{
		try {
			XmlPullParser parser = Xml.newPullParser();
			parser.setFeature(XmlPullParser.FEATURE_PROCESS_NAMESPACES, false);
			parser.setInput(inputStream, null);
			parser.nextTag(); return
			readFeed(parser);
		} finally {
			inputStream.close();
		}
	}
	private List<RssItem> readFeed(XmlPullParser parser) throws
	XmlPullParserException,IOException {
		parser.require(XmlPullParser.START_TAG, null, "rss");
		String title = null;
		String link = null;
		List<RssItem> items = new ArrayList<RssItem>();
		while (parser.next() != XmlPullParser.END_DOCUMENT) {
			if (parser.getEventType() != XmlPullParser.START_TAG) {
				continue;
			}
			String name = parser.getName();
			if (name.equals("title")) {
				title = readTitle(parser);
			    } else if (name.equals("link")) {
					link = readLink(parser);
			}
			if (title != null && link != null) {
				RssItem item = new RssItem(title, link);
				items.add(item);
				title = null;
				link = null;
			}
		}
		return items;
	}
```

```
private String readLink(XmlPullParser parser) throws XmlPullParserException,IOException
        {parser.require(XmlPullParser.START_TAG, ns, "link");
        String link = readText(parser);
        parser.require(XmlPullParser.END_TAG, ns, "link");
        return link;
}
private String readTitle(XmlPullParser parser) throws XmlPullParserException,IOException
        {parser.require(XmlPullParser.START_TAG, ns, "title");
        String title = readText(parser);
        parser.require(XmlPullParser.END_TAG, ns, "title");
        return title;
}/
/ For the tags title and link, extract their text values.
private String readText(XmlPullParser parser) throws IOException, XmlPullParserException {
```

```
                String result = "";
                if (parser.next() == XmlPullParser.TEXT) {
                        result = parser.getText();
                        parser.nextTag();
                }
                return result;
        }
}
```

RssAdapter.java

```
package com.example.ex_no_4;

import java.util.List;
import android.content.Context;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.TextView;
public class RssAdapter extends BaseAdapter {
        private final List<RssItem>items;
        private final Contextcontext;
        public RssAdapter(Context context, List<RssItem> items) {
                this.items= items;
                this.context= context;
        }
        @Override
        public int getCount() {
                return items.size();
        }
        @Override
        public Object getItem(int position) {
                return items.get(position);
        }
        @Override
        public long getItemId(int id) {
                return id;
        }
        @Override
        public View getView(int position, View convertView, ViewGroup parent)
                {ViewHolder holder;
                if (convertView == null) {
                        convertView = View.inflate(context, R.layout.rss_item, null);
                        holder = new ViewHolder();
                        holder.itemTitle = (TextView) convertView.findViewById(R.id.itemTitle);
                        convertView.setTag(holder);
                } else {
                        holder = (ViewHolder) convertView.getTag();
                }
```

```
                holder.itemTitle.setText(items.get(position).getTitle()); return
                convertView;
        }
        static class ViewHolder {
                TextView itemTitle;
        }
}
```

RssFragement.java

```
package com.example.ex_no_4;
```

```
import java.util.List;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.os.Handler;
import android.os.ResultReceiver;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ListView; import
android.widget.ProgressBar;
import android.widget.Toast;
public class RssFragment extends Fragment implements OnItemClickListener
        { private ProgressBarprogressBar;
        private ListViewlistView;
        private Viewview;
        @Override
        public void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);setRetain
                Instance(true);
        }
        @Override
        public View onCreateView(LayoutInflater inflater, ViewGroup container,
        BundlesavedInstanceState) {
                if (view== null) {
                        view = inflater.inflate(R.layout.fragment_layout, container, false);
                        progressBar = (ProgressBar) view.findViewById(R.id.progressBar);
                        listView = (ListView) view.findViewById(R.id.listView);
                        listView.setOnItemClickListener(this);
                        startService();
                } else {
                        ViewGroup parent = (ViewGroup) view.getParent();
                        parent.removeView(view);
                }
                return view;
        }
        private void startService() {
                Intent intent = new Intent(getActivity(), RssService.class);
                intent.putExtra(RssService.RECEIVER, resultReceiver);
                getActivity().startService(intent);
        }
        private final ResultReceiverresultReceiver= new ResultReceiver(new Handler())
                {@SuppressWarnings("unchecked")
                @Override
                protected void onReceiveResult(int resultCode, Bundle resultData)
                        {progressBar.setVisibility(View.GONE);
                        List<RssItem> items = (List<RssItem>)
```

```
resultData.getSerializable(RssService.ITEMS);
if (items != null) {
        RssAdapter adapter = new RssAdapter(getActivity(), items);
        listView.setAdapter(adapter);
}
else {
        Toast.makeText(getActivity(), "An error occured while downloading the
        rss feed.",
        Toast.LENGTH_LONG).show();
}
```

```
                };
        };
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int position, long id)
                {RssAdapter adapter = (RssAdapter) parent.getAdapter();
                RssItem item = (RssItem) adapter.getItem(position); Uri
                uri = Uri.parse(item.getLink());
                Intent intent = new Intent(Intent.ACTION_VIEW, uri);
                startActivity(intent);
        }
}
```

RssItem.java

```
package com.example.ex_no_4;

public class RssItem {
        private final Stringtitle;
        private final Stringlink;
        public RssItem(String title, String link) {
                this.title= title;
                this.link= link;
        }
        public String getTitle() {
                return title;
        }
        public String getLink() {
                return link;
        }
}
```

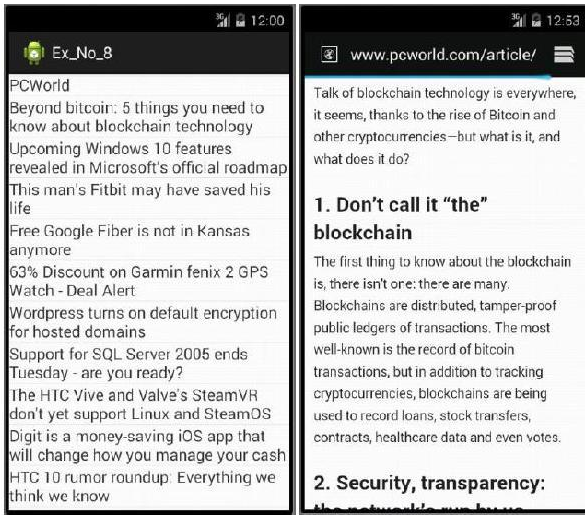
RssService.java

```
package com.example.ex_no_4;
import java.io.IOException;
import java.io.InputStream;
import java.io.Serializable;
import java.net.URL;
import java.util.List;
import org.xmlpull.v1.XmlPullParserException;
import android.app.IntentService;
import android.content.Intent;
import android.os.Bundle;
import android.os.ResultReceiver;
import android.util.Log;
public class RssService extends IntentService {
        private static final String RSS_LINK="http://www.pcworld.com/index.rss";
        public static final String ITEMS="items";
        public static final String RECEIVER="receiver";
        public RssService() {
                super("RssService");
```

```
}
@Override
protected void onHandleIntent(Intent intent)
        {Log.d(Constants.TAG, "Service started");
        List<RssItem> rssItems = null;
        try {
                PcWorldRssParser parser = new PcWorldRssParser();
                rssItems = parser.parse(getInputStream(RSS_LINK));
        }
```

```
            catch (XmlPullParserException e) {
                Log.w(e.getMessage(), e);
            }
            catch (IOException e) {
                Log.w(e.getMessage(), e);
            }
            Bundle bundle = new Bundle();
            bundle.putSerializable(ITEMS, (Serializable) rssItems); ResultReceiver
            receiver = intent.getParcelableExtra(RECEIVER); receiver.send(0,
            bundle);
        }
        public InputStream getInputStream(String link) {
            try {
                URL url = new URL(link);
                return url.openConnection().getInputStream();
            }
            catch (IOException e) {
                Log.w(Constants.TAG, "Exception while retrieving the input stream", e); return
                null;
            }
        }
}
```

OUTPUT:



VIVA QUESTIONS

1. **What is the Adapter in Android?**
   An adapter is used to create a child view to present the parent view items.
2. **What is a content provider in android?**
   A content provider component supplies data from one application to others on request. Such requests are handled by the methods of the ContentResolver class. A content provider can use different ways to store its data and the data can be stored in a database, in files, or even over a network.
3. **How do you pass the data to sub-activities android?**
   Using with Bundle, we can pass the data to sub activities.
   Bundle bun = new Bundle();

```
bun.putString("EMAIL",
"abc@xyz.com");
```

4. **Name some exceptions in Android?**
   - Inflate Exception
   - Surface.OutOfResourceException
   - SurfaceHolder.BadSurfaceTypeExceptio

   n o WindowManager.BadTokenException

5. **What is sleep mode in Android?**
   In sleep mode, CPU is slept and doesn't accept any commands from android device except Radio interface layer and alarm.

EXPERIMENT-5

Write an application that draws basic graphical primitives on the screen

AIM: Write an application that draws basic graphical primitives on the screen

PROCEDURE:

1. Open Eclipse IDE.
2. Create the project Ex_no_5.
3. Go to package explorer in the left hand side. Select the project Ex_no_5.
4. Go to res folder and select layout. Double click the activity_main.xml file.
5. Now you can see the Graphical layout window.
6. Drag and drop only one ImageView
7. Again go to package explorer in the left hand side. Select the project Ex_No_5.
8. Go to src folder. Double click the MainActivity.java file.
9. In java file write the activities done by the application such as drawing the graphical primitives.
10. Finally run the android application.

PROGRAMS:

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"
        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.ex_no_5.MainActivity">
<ImageView
        android:id="@+id/imageView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:src="@drawable/ic_launcher"
        tools:ignore="ContentDescription" />
</RelativeLayout>
```

MainActivity.java:

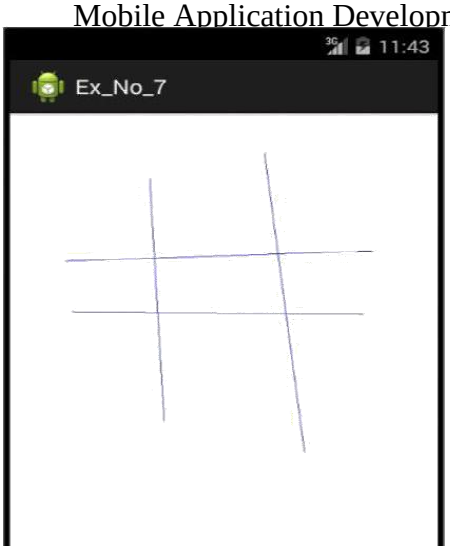
```
package com.example.ex_no_5;
```

```
import android.support.v7.app.ActionBarActivity;
import android.annotation.SuppressLint;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.os.Bundle;
import android.view.Display;
```

```
import android.view.MotionEvent;
import android.view.View;
import android.view.View.OnTouchListener;
import android.widget.ImageView;
@SuppressLint("ClickableViewAccessibility")
public class MainActivity extends ActionBarActivity implements OnTouchListener
	{ImageView iv;
	Bitmap b;
	Canvas c;
	Paint p;
	float dx=0,dy=0,ux=0,uy=0;
	@SuppressWarnings("deprecation")
	@Override
	protected void onCreate(Bundle savedInstanceState) {
		super.onCreate(savedInstanceState);
		setContentView(R.layout.activity_main);
		iv=(ImageView)this.findViewById(R.id.imageView1);
		Display d = getWindowManager().getDefaultDisplay();
		float dw = d.getWidth();
		float dh = d.getHeight();
		b  = Bitmap.createBitmap((int) dw, (int) dh,Bitmap.Config.ARGB_8888);
		c  = new Canvas(b);
		p  = new Paint();
		p.setColor(Color.BLUE);
		iv.setImageBitmap(b);
		iv.setOnTouchListener(this);
	}
	@Override
	public boolean onTouch(View v, MotionEvent event) {
		//   TODO Auto-generated method
		stub int action = event.getAction();
		switch (action)
		{
			case MotionEvent.ACTION_DOWN:
				dx   =   event.getX();
				dy   =   event.getY();
				break;
			case
				MotionEvent.ACTION_M
				OVE: break;
			case
				MotionEvent.ACTION_
				UP:ux = event.getX();
				uy = event.getY();
				c.drawLine(dx, dy, ux, uy, p);
				iv.invalidate();
```

```
                    break;
                case
                    MotionEvent.ACTION_CAN
                    CEL: break;
                default:
                    break
                    ;
            }
            return true;
        }
}
```

OUTPUT:


11:43
Ex_No_7

VIVA QUESTIONS

1. **What is an Action?**
   Description of the intent. For instance, ACTION_CALL — used to perform calls.
2. **What are fragments?**
   Fragment is a UI entity attached to Activity. Fragments can be reused by attaching in different activities. Activity can have multiple fragments attached to it. Fragment must be attached to an activity and its lifecycle will depend on its host activity.
3. **When should you use a fragment rather than an activity?**
   - When there are UI components that are going to be used across multiple activities.
   - When there are multiple views that can be displayed side by side (viewPager tabs)
   - When you have data that needs to be persisted across Activity restarts (such as retained fragments)
4. **What is the difference between Dialog & DialogFragment?**
   A fragment that displays a dialog window, floating on top of its activity‘s window. This fragment contains a Dialog object, which it displays as appropriate, based on the fragment‘s state. Dialogs are entirely dependent on Activities. If the screen is rotated, the dialog is dismissed. Dialog fragments take care of orientation, configuration changes as well.
5. **Difference between margin and padding.**
   Padding will be space added inside the container, for instance, if it is a button, padding will be added inside the button. Margin will be space added outside the container.

EXPERIMENT-6

Create an android app for database creation using SQLite Database

AIM: Create an android app for database creation using SQLite Database

PROCEDURE:

1. Open Eclipse IDE.
2. Create the project Ex_No_6.
3. Go to package explorer in the left hand side. Select the project Ex_No_6.
4. Go to res folder and select layout. Double click the activity_main.xml file.
5. Now you can see the Graphical layout window.
6. Drag and drop the following components:

a. Three TextViews with texts as Reg.No., Name and
Marks b. Three EditTexts
c. Five Buttons with labeled as ADD, VIEW, VIEW ALL, UPDATE and DELETE

7. Again go to package explorer in the left hand side. Select the project Ex_No_6.
8. Go to src folder. Double click the MainActivity.java file.
9. In java file write the activities done by the application such as, actions of button.
10. Finally run the android application.

PROGRAMS:

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"
        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.ex_no_6.MainActivity">
<TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:text="Reg. No."
        android:textAppearance="?android:attr/textAppearanceMedium"
        tools:ignore="HardcodedText" />
<EditText
        android:id="@+id/editText1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
        android:layout_alignTop="@+id/textView1"
        android:layout_toRightOf="@+id/textView1"
        android:ems="10"
        android:inputType="number">
<requestFocus />
</EditText>
<TextView
        android:id="@+id/textView2"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/textView1"
        android:layout_below="@+id/editText1"
        android:layout_marginTop="20dp"
        android:text="Name"
        android:textAppearance="?android:attr/textAppearanceMedium"
        tools:ignore="HardcodedText" />
<TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/textView2"
        android:layout_below="@+id/editText2"
        android:layout_marginTop="26dp"
        android:text="Marks"
        android:textAppearance="?android:attr/textAppearanceMedium"
        tools:ignore="HardcodedText" />
<EditText
        android:id="@+id/editText3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBottom="@+id/textView3"
        android:layout_alignLeft="@+id/editText2"
        android:ems="10"
        android:inputType="number" />
<EditText
        android:id="@+id/editText2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/textView2"
        android:layout_alignBottom="@+id/textView2"
        android:layout_alignLeft="@+id/editText1"
        android:ems="10"
        tools:ignore="TextFields" />
<Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/textView3"
        android:layout_marginTop="32dp"
        android:text="ADD"
        tools:ignore="HardcodedText" />
<Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/button2"
        android:layout_alignBottom="@+id/button2"
```

```
        android:layout_alignParentRight="true"
        android:text="VIEW ALL"
        tools:ignore="HardcodedText" />
<Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/button1"
        android:layout_alignBottom="@+id/button1"
        android:layout_alignLeft="@+id/editText3"
```

```
        android:layout_marginLeft="24dp"
        android:text="VIEW"
        tools:ignore="HardcodedText" />
<Button
        android:id="@+id/button4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/button1"
        android:layout_below="@+id/button1"
        android:layout_marginLeft="27dp"
        android:layout_marginTop="18dp"
        android:text="UPDATE"
        tools:ignore="HardcodedText" />
<Button
        android:id="@+id/button5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/button4"
        android:layout_alignBottom="@+id/button4"
        android:layout_marginLeft="20dp"
        android:layout_toRightOf="@+id/button4"
        android:text="DELETE"
        tools:ignore="HardcodedText" />

</RelativeLayout>
```

MainActivity.java:

```
package com.example.ex_no_6;

import android.support.v7.app.ActionBarActivity;
import android.app.AlertDialog.Builder; import
android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
public class MainActivity extends ActionBarActivity
        {EditText name,regno,mark;
        Button btnAdd,btnDelete,btnUpdate,btnView,btnViewAll;
        SQLiteDatabase db;
        @Override
        protected void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.activity_main);
                regno= (EditText)findViewById(R.id.editText1);
                name= (EditText)findViewById(R.id.editText2);
```

```
mark=(EditText)findViewById(R.id.editText3);
btnAdd=(Button)findViewById(R.id.button1);
btnView=(Button)findViewById(R.id.button2);
btnViewAll=(Button)findViewById(R.id.button3);
btnUpdate=(Button)findViewById(R.id.button4);
btnDelete=(Button)findViewById(R.id.button5);
db=openOrCreateDatabase("Students", Context.MODE_PRIVATE, null);
db.execSQL("CREATE TABLE IF NOT EXISTS student(regno VARCHAR,name
VARCHR,mark VARCHAR);");
btnAdd.setOnClickListener(new OnClickListener()
```

```
        {
                @Override
                public void onClick(View arg0) {
                        // TODO Auto-generated method stub
                        if(regno.getText().toString().trim().length()==0||name.getText().toString().trim().
                        length()==0||mark.getText().toString().trim().length()==0) {

                                showMessage("Error", "Please enter all values");
                                return;
                        }
                        db.execSQL("INSERT INTO student
        VALUES('"+regno.getText()+"','"+name.getText()+"','"+mark.getText()+"');");
                        showMessage("Success", "Record added");
                        clearText();
        }
});
btnDelete.setOnClickListener(new OnClickListener()
{
        @Override
        public void onClick(View v) {
        // TODO Auto-generated method stub
        if(regno.getText().toString().trim().length()==0)
        {
                showMessage("Error", "Please enter Reg. No.");
                return;
        }
        Cursor c=db.rawQuery("SELECT * FROM student
        WHERE regno='"+regno.getText()+"'", null);
        if(c.moveToFirst())
        {
                db.execSQL("DELETE FROM student WHERE
                regno='"+regno.getText()+"'"); showMessage("Success", "Record Deleted");
        }
        else {
                showMessage("Error", "Invalid Reg. No.");
        }
        clearText();
        }
});
btnUpdate.setOnClickListener(new OnClickListener()
{
        @Override
        public void onClick(View v) {
                // TODO Auto-generated method stub
                if(regno.getText().toString().trim().length()==0)
                {
                        showMessage("Error", "Please enter Reg. No.");
                        return;
                }
```

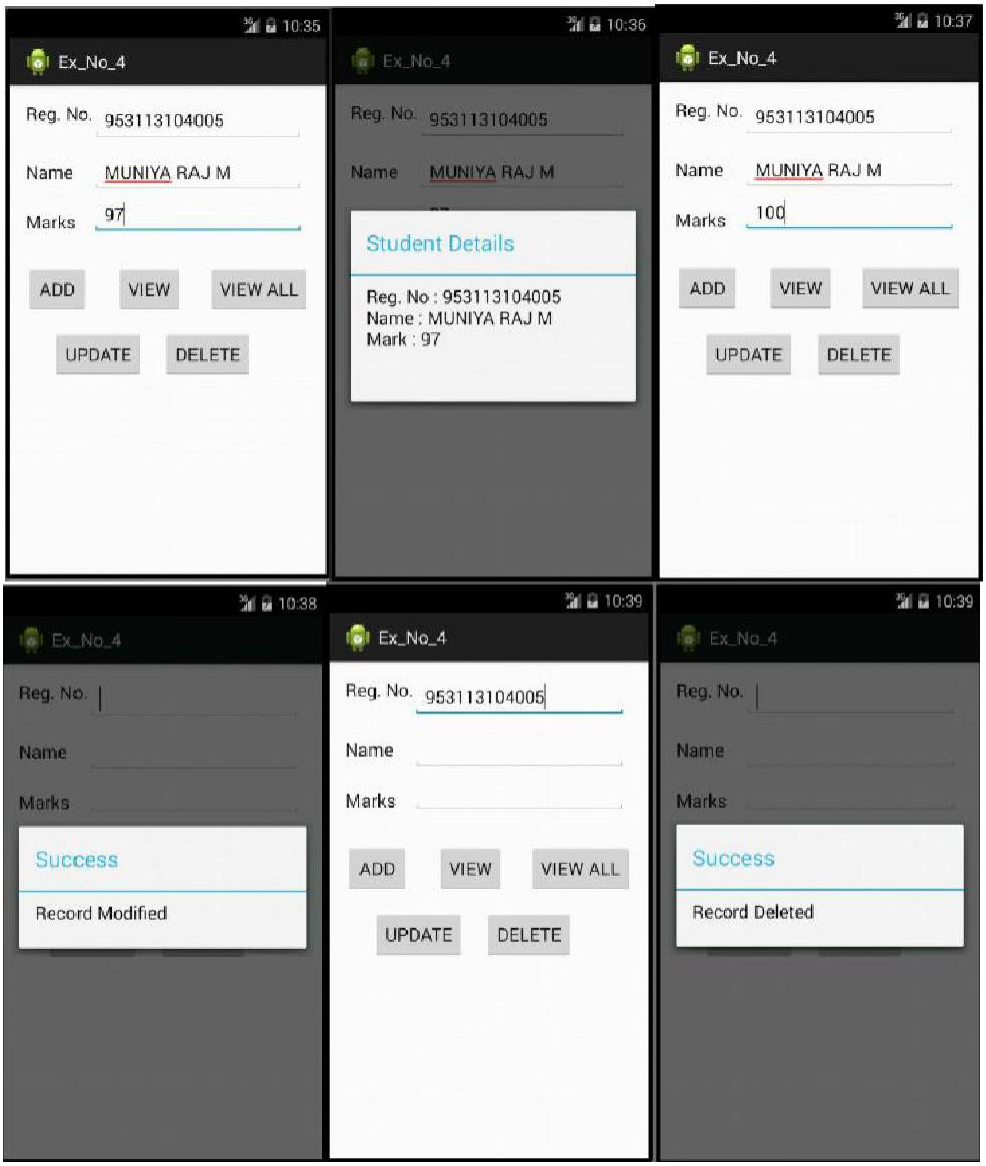
```
Cursor c=db.rawQuery("SELECT * FROM student
WHERE regno='"+regno.getText()+"'", null);
if(c.moveToFirst())
{
        db.execSQL("UPDATE student SET
        name='"+name.getText()+"',mark='"+mark.getText()+"'
        WHERE regno='"+regno.getText()+"'");
        showMessage("Success", "Record Modified");
} else
{
        showMessage("Error", "Invalid Reg. No.");
}
```

```
                clearText();
        }
});
btnView.setOnClickListener(new OnClickListener()
{
        @Override
        public void onClick(View v) {
                // TODO Auto-generated method stub
                if(regno.getText().toString().trim().length()==0)
                {
                        showMessage("Error", "Please enter Reg. No.");
                        return;
                }
                Cursor c=db.rawQuery("SELECT * FROM student
                WHERE regno='"+regno.getText()+"'", null);
                if(c.moveToFirst())
                {
                        name.setText(c.getString(1));
                        mark.setText(c.getString(2));
                } else
                {
                        showMessage("Error", "Invalid Reg. No.");
                        clearText();
                }
        }
});
btnViewAll.setOnClickListener(new OnClickListener()
{
        @Override
        public void onClick(View v) {
        // TODO Auto-generated method stub
        Cursor c=db.rawQuery("SELECT * FROM student",
        null); if(c.getCount()==0)
        {
                showMessage("Error", "No records found");
                return;
        }
        StringBuffer buffer=new StringBuffer();
        while(c.moveToNext())
        {
                buffer.append("Reg. No : "+c.getString(0)+"\n");
                buffer.append("Name : "+c.getString(1)+"\n");
                buffer.append("Mark : "+c.getString(2)+"\n\n");
        }
        showMessage("Student Details", buffer.toString());
        }
});
}
public void showMessage(String title,String message)
{
```

```
        Builder builder=new Builder(this);
        builder.setCancelable(true);
        builder.setTitle(title);
        builder.setMessage(message);
        builder.show();
}
public void clearText()
{
        regno.setText("");
        name.setText("");
        mark.setText("");
```

```
                regno.requestFocus();
        }
}
```

OUTPUT



EXPERIMENT-7

Develop a native application that uses GPS location information

AIM: Develop a native application that uses GPS location information

PROCEDURE:

1. Open Eclipse IDE.
2. Create the project Ex_No_7.
3. Go to package explorer in the left hand side. Select the project Ex_No_7.
4. Go to res folder and select layout. Double click the activity_main.xml file.
5. Now you can see the Graphical layout window.
6. Drag and drop the following components:

a. One TextView with text as Current Location

b. Two TextViews without any texts.

7. Again go to package explorer in the left hand side. Select the project Ex_No_7.
8. Go to src folder. Double click the MainActivity.java file.
9. In java file write the activities done by the application such as finding current location and print them.
10. Get the following permission in AndroidManifest.xml file:

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

11. Finally run the android application.

PROGRAMS:

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"
        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.ex_no_7.MainActivity">
<TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="114dp"
        android:text=""
        android:textAppearance="?android:attr/textAppearanceMedium"
        tools:ignore="HardcodedText" />
<TextView
```

```
android:id="@+id/textView2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/textView1"
android:layout_alignParentRight="true"
android:layout_below="@+id/textView1"
android:layout_marginTop="51dp"
```

```
        android:text=""
        android:textAppearance="?android:attr/textAppearanceMedium"
        tools:ignore="HardcodedText" />
<TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="47dp"
        android:text="Current Location"
        android:textAppearance="?android:attr/textAppearanceLarge"
        tools:ignore="HardcodedText" />
</RelativeLayout>
```

MainActivity.java:

```
package com.example.ex_no_7;
import android.support.v7.app.ActionBarActivity;
import android.content.Context;
import android.location.Criteria;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.widget.TextView;
import android.widget.Toast;
public class MainActivity extends ActionBarActivity implements
        LocationListener{@Override
        protected void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);setContentV
                iew(R.layout.activity_main);
                LocationManager
                lm=(LocationManager)getSystemService(Context.LOCATION_SERVICE); Criteria c=new
                Criteria();
                String s=lm.getBestProvider(c, false);
                if(s!=null && !s.equals(""))
                {
                        Location l=lm.getLastKnownLocation(s);
                        lm.requestLocationUpdates(s, 20000, 1, this);
                        if(l!=null)
                                onLocationChanged(l);
                        else
                                Toast.makeText(getApplicationContext(), "Location can't be
                                retrieved !!!", Toast.LENGTH_LONG).show();
                }
                else
```

```
                Toast.makeText(getApplicationContext(), "Provider not found !!!",
                Toast.LENGTH_LONG).show();
        }
        @Override
        public void onLocationChanged(Location arg0) {
                // TODO Auto-generated method stub
                TextView
                t1=(TextView)findViewById(R.id.textView1);
                t1.setText("Latitude : \n"+arg0.getLatitude()); TextView
                t2=(TextView)findViewById(R.id.textView2);
                t2.setText("Longitude : \n"+arg0.getLongitude());
        }
        @Override
        public void onProviderDisabled(String arg0) {
                // TODO Auto-generated method stub
```

```
	}
	@Override
	public void onProviderEnabled(String arg0) {
		// TODO Auto-generated method stub
	}
	@Override
	public void onStatusChanged(String arg0, int arg1, Bundle arg2) {//
		TODO Auto-generated method stub
	}
}
```

OUTPUT

EXPERIMENT-8

Implement an application that writes data to the SD card.

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

PROCEDURE:

1. Open Eclipse IDE.
2. Create the project Ex_No_8.
3. Go to package explorer in the left hand side. Select the project Ex_No_8.
4. Go to res folder and select layout. Double click the activity_main.xml file.
5. Now you can see the Graphical layout window.
6. Drag and drop the following components:
    a. Two EditTexts
    b. Two Buttons with labeled as READ and SAVE
7. Again go to package explorer in the left hand side. Select the project Ex_No_8.
8. Go to src folder. Double click the MainActivity.java file.
9. In java file write the activities done by the application such as actions of buttons.
10. Get the following permission in AndroidManifest.xml file:

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

11. Finally run the android application.

PROGRAMS:

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"
    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.ex_no_8.MainActivity">
<EditText
    android:id="@+id/editText1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:ems="10"
    android:hint="Path"
    tools:ignore="TextFields,HardcodedText">
<requestFocus />
</EditText>
<Button
```

```
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignTop="@+id/editText1"
        android:layout_toRightOf="@+id/editText1"
        android:text="READ"
        tools:ignore="HardcodedText" />
<EditText
```

```
        android:id="@+id/editText2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/editText1"
        android:layout_centerVertical="true"
        android:ems="10"
        android:hint="Contents of File"
        android:inputType="textMultiLine"
        tools:ignore="HardcodedText" />
<Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_centerVertical="true"
        android:text="SAVE"
        tools:ignore="HardcodedText" />
</RelativeLayout>
```

MainActivity.java:

```
package com.example.ex_no_8;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import android.support.v7.app.ActionBarActivity;
import android.annotation.SuppressLint; import
android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
public class MainActivity extends ActionBarActivity
        {@SuppressLint("SdCardPath")
        @Override
        protected void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);setContentV
                iew(R.layout.activity_main);
                final  EditText  e1=(EditText)findViewById(R.id.editText1);
                final EditText e2=(EditText)findViewById(R.id.editText2);
                Button b1=(Button)findViewById(R.id.button1);
                Button b2=(Button)findViewById(R.id.button2);
```

```
String path=getPreferences(MODE_PRIVATE).getString("fpath", "/sdcard/file1");
e1.setText(path);
b1.setOnClickListener(
        new OnClickListener()
        {
                @Override
                public void onClick(View arg0) {
                        // TODO Auto-generated method stub
                        File f=new File(e1.getText().toString());
                        String s="";
                        StringBuilder sb=new StringBuilder();
                        FileReader fr = null;
                        try {
```

```
                fr = new FileReader(f);
            } catch (FileNotFoundException e) {
                //   TODO Auto-generated catch
                blocke.printStackTrace();
            }
            BufferedReader br=new BufferedReader(fr);
            try {
                while((s=br.readLine())!=null)
                {
                    sb.append(s+"\n");
                }
            } catch (IOException e) {
                //   TODO Auto-generated catch block
                e.printStackTrace();
            }
            Toast.makeText(getApplicationContext(), "File Read
            Successfully !!!", Toast.LENGTH_LONG).show();
            e2.setText(sb);
        }
    });
b2.setOnClickListener(
    new OnClickListener()
    {
        @Override
        public void onClick(View arg0) {
            // TODO Auto-generated method stub
            File f=new File(e1.getText().toString());
            FileWriter fw = null;
            try {
                fw = new FileWriter(f);
            } catch (IOException e3) {
                //   TODO Auto-generated catch block
                e3.printStackTrace();
            }
            try {
                fw.write(e2.getText().toString());
            } catch (IOException e2) {
                //   TODO Auto-generated catch block
                e2.printStackTrace();
            }
            try {
                 fw.close();
            } catch (IOException e2) {
                //   TODO Auto-generated catch block
                e2.printStackTrace();
            }
            SharedPreferences.Editor
            e=getPreferences(MODE_PRIVATE).edit();
```

```
                e.putString("fpath", f.getPath());
                e.commit();
                Toast.makeText(getApplicationContext(), "Saved
                Successfully !!!", Toast.LENGTH_LONG).show();
            }
        });
    }
}
```

OUTPUT


/sdcard/file1
READ
This app is developed by MUNIYA RAJ M
SAVE
Saved Successfully !!!
/sdcard/file1
READ
This app is developed by MUNIYA RAJ M
SAVE
File Read Successfully !!!

EXPERIMENT-9

Design a gaming application

Styles.xml

```
<resources>

    <!-- Base application theme. -->
    <style name="AppTheme"
        parent="Theme.AppCompat.Light.NoActionBar"><item
        name="windowNoTitle">true</item>
        <item name="windowActionBar">false</item>
        <item name="android:windowFullscreen">true</item>
        <item
    name="android:windowContentOverlay">@null</item></style>

</resources>
```

activity_main.xml

```
<?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"
    android:background="@drawable/splash"
    tools:context="net.simplifiedcoding.simplegame.MainActivity">

    <ImageButton
        android:id="@+id/buttonPlay"
        android:background="@drawable/playnow"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/buttonScore"
        android:layout_centerHorizontal="true" />

    <ImageButton
        android:id="@+id/buttonScore"
```

```
android:background="@drawable/highscore"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
```

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

    </RelativeLayout>
```

- When we tap the Play Now button our Game Activity will start.
- Now come inside MainActivity.java and write the following code.

MainActivity.java

```
package net.simplifiedcoding.simplegame;

import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.media.Image;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;

public class MainActivity extends AppCompatActivity implements View.OnClickListener{

    //image button
    private ImageButton buttonPlay;

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

        //setting the orientation to landscape
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);

        //getting the button
        buttonPlay = (ImageButton) findViewById(R.id.buttonPlay);

        //adding a click listener
        buttonPlay.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
```

//starting game activity

```
        startActivity(new Intent(this, GameActivity.class));
    }
}
```

GameView.java

```
public class GameView extends SurfaceView implements Runnable {

    //boolean variable to track if the game is playing or not
    volatile boolean playing;

    //the game thread
    private Thread gameThread = null;


    //Class constructor
    public GameView(Context context) {
        super(context);

    }

    @Override
    public void run() {
        while (playing) {
      //to update the frame
            update();

      //to draw the frame
            draw();

      //to control
            control();
        }
    }


    private void update() {

    }

    private void draw() {

    }
```

```
private void control() {
```

```
        try {
            gameThread.sleep(17);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }

    public void pause() {
   //when the game is paused
   //setting the variable to false
        playing = false;
        try {
       //stopping the thread
            gameThread.join();
        } catch (InterruptedException e) {
        }
    }

    public void resume() {
   //when the game is resumed
   //starting the thread again
        playing = true;
        gameThread = new Thread(this);
        gameThread.start();
    }
}
```

- The above class is our GameView class. It is the actual game panel where we will play the game. The class is implementing Runnable interface. We have a volatile boolean type variable running that will track whether the game is running or not. After that we have our gameThread, it is the main game loop. Then we have the constructor to the class. We are not doing anything inside the constructor right now. Then we have the overriden method run(), here we are running a loop until the playing variable running is true. Inside the loop we are calling the following methods.
- update() -> Here we will update the coordinate of our characters.
- draw() -> Here we will draw the characters to the canvas.
- control() -> This method will control the frames per seconds drawn. Here we are calling the delay method of Thread. And this is actually making our frame rate to aroud 60fps.
- After these we have two more methods.
- pause() -> To pause the game, we are stopping the gameThread here.
- resume() -> To resume the game, here we are starting the gameThread.

GameActivity.java

```
package net.simplifiedcoding.spacefighter;

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

public class GameActivity extends AppCompatActivity {

   //declaring gameview
   private GameView gameView;

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

      //Initializing game view object
      gameView = new GameView(this);

      //adding it to contentview
      setContentView(gameView);
   }

   //pausing the game when activity is paused
   @Override
   protected void onPause() {
      super.onPause();
      gameView.pause();
   }

   //running the game when activity is resumed
   @Override
   protected void onResume() {
      super.onResume();
      gameView.resume();
   }
}
```

Player.java

```
package net.simplifiedcoding.spacefighter;
```

```
import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
```

```
public class Player {
    //Bitmap to get character from image
    private Bitmap bitmap;

    //coordinates
    private int x;
    private int y;

    //motion speed of the character
    private int speed = 0;

    //constructor
    public Player(Context context) {
        x = 75;
        y = 50;
        speed = 1;

        //Getting bitmap from drawable resource
        bitmap = BitmapFactory.decodeResource(context.getResources(), R.drawable.player);
    }

    //Method to update coordinate of character
    public void update(){
        //updating x coordinate
        x++;
    }

    /*
    * These are getters you can generate it autmaticallyl
    * right click on editor -> generate -> getters
    * */
    public Bitmap getBitmap() {
        return bitmap;
    }

    public int getX() {
        return x;
    }

    public int getY() {
        return y;
    }
```

```
    public int getSpeed() {
        return speed;
```

```
    }
}
```

Drawing Player to GameView: To draw the player to our GameView you need to come back to theGameView.java class and modify it as below.

GameView.java

```
public class GameView extends SurfaceView implements Runnable {

    volatile boolean playing;
    private Thread gameThread = null;

    //adding the player to this class
    private Player player;

    //These objects will be used for drawing
    private Paint paint;
    private Canvas canvas;
    private SurfaceHolder surfaceHolder;

    public GameView(Context context) {
        super(context);

        //initializing player object
        player = new Player(context);

        //initializing drawing objects
        surfaceHolder = getHolder();
        paint = new Paint();
    }

    @Override
    public void run() {
        while (playing) {
            update();
            draw();
            control();
        }
    }

    private void update() {
        //updating player position
```

```
        player.update();
    }
```

```
    private void draw() {
        //checking if surface is valid
        if (surfaceHolder.getSurface().isValid()) {
            //locking the canvas
            canvas = surfaceHolder.lockCanvas();
            //drawing a background color for canvas
            canvas.drawColor(Color.BLACK);
            //Drawing the player
            canvas.drawBitmap(
                    player.getBitmap(),
                    player.getX(),
                    player.getY(),
                    paint);
            //Unlocking the canvas
            surfaceHolder.unlockCanvasAndPost(canvas);
        }
    }

    private void control() {
        try {
            gameThread.sleep(17);
        } catch (InterruptedException e)
            { e.printStackTrace();
        }
    }

    public void pause() {
        playing = false;
        try {
            gameThread.join();
        } catch (InterruptedException e) {
        }
    }

    public void resume() {
        playing = true;
        gameThread = new Thread(this);
        gameThread.start();
    }
}
```

Output without Control:

Adding Controls:

```
@Override
public boolean onTouchEvent(MotionEvent motionEvent) {
   switch (motionEvent.getAction() & MotionEvent.ACTION_MASK)
      { case MotionEvent.ACTION_UP:
         //When the user presses on the screen
         //we will do something here
```

```
            break;
        case MotionEvent.ACTION_DOWN:
            //When the user releases the screen
            //do something here
            break;
    }
    return true;
}
```

Player.java

```
public class Player {
    private Bitmap bitmap;
    private int x;
    private int y;
    private int speed = 0;

    //boolean variable to track the ship is boosting or not
    private boolean boosting;

    //Gravity Value to add gravity effect on the ship
    private final int GRAVITY = -10;

    //Controlling Y coordinate so that ship won't go outside the screen
    private int maxY;
    private int minY;

    //Limit the bounds of the ship's speed
    private final int MIN_SPEED = 1;
    private final int MAX_SPEED = 20;

    public Player(Context context) {
        x = 75;
        y = 50;
        speed = 1;
        bitmap = BitmapFactory.decodeResource(context.getResources(), R.drawable.player);

        //setting the boosting value to false initially
        boosting = false;
    }


    //setting boosting true
```

```
public void setBoosting() {
```

```
    boosting = true;
}

//setting boosting false
public void stopBoosting() {
    boosting = false;
}

public void update() {
    //if the ship is boosting
    if (boosting) {
        //speeding up the ship
        speed += 2;
    } else {
        //slowing down if not boosting
        speed -= 5;
    }
    //controlling the top speed
    if (speed > MAX_SPEED) {
        speed = MAX_SPEED;
    }
    //if the speed is less than min speed
    //controlling it so that it won't stop completely
    if (speed < MIN_SPEED) {
        speed = MIN_SPEED;
    }

    //moving the ship down
    y -= speed + GRAVITY;

    //but controlling it also so that it won't go off the
    screen if (y < minY) {
        y = minY;
    }
    if (y > maxY) {
        y = maxY;
    }
}

public Bitmap getBitmap() {
    return bitmap;
}
```

```
public int getX() {
    return x;
```

```
    }

    public int getY() {
        return y;
    }

    public int getSpeed() {
        return speed;
    }
}
```

GameActivity.java

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

        //Getting display object
        Display display = getWindowManager().getDefaultDisplay();

        //Getting the screen resolution into point object
        Point size = new Point();
        display.getSize(size);

        //Initializing game view object
        //this time we are also passing the screen size to the GameView constructor
        gameView = new GameView(this, size.x, size.y);

        //adding it to contentview
        setContentView(gameView);
    }
```

Now to complete adding the boosters come inside GameView.java file and modify the onTouchEvent() as follows.

```
    @Override
    public boolean onTouchEvent(MotionEvent motionEvent) {
        switch (motionEvent.getAction() & MotionEvent.ACTION_MASK)
            { case MotionEvent.ACTION_UP:
                //stopping the boosting when screen is released
                player.stopBoosting();
                break;
            case MotionEvent.ACTION_DOWN:
```

```
//boosting the space jet when screen is pressed
player.setBoosting();
```

```
            break;
        }
        return true;
    }
```

Now we will add background stars to make the background looks animating.

```
package net.simplifiedcoding.spacefighter;

import java.util.Random;

public class Star {
    private int x;
    private int y;
    private int speed;

    private int maxX;
    private int maxY;
    private int minX;
    private int minY;



    public Star(int screenX, int screenY) {
        maxX = screenX;
        maxY = screenY;
        minX = 0;
        minY = 0;
        Random generator = new Random();
        speed = generator.nextInt(10);

        //generating a random coordinate
        //but keeping the coordinate inside the screen size
        x = generator.nextInt(maxX); y =
        generator.nextInt(maxY);
    }

    public void update(int playerSpeed) {
        //animating the star horizontally left side
        //by decreasing x coordinate with player speed
        x -= playerSpeed;
        x -= speed;
        //if the star reached the left edge of the screen
```

```
if (x < 0) {
    //again starting the star from right edge
```

```
            //this will give a infinite scrolling background
            effect x = maxX;
            Random generator = new Random();
            y = generator.nextInt(maxY);
            speed = generator.nextInt(15);
        }
    }

    public float getStarWidth() {
        //Making the star width random so that
        //it will give a real look
        float minX = 1.0f;
        float maxX = 4.0f;
        Random rand = new Random();
        float finalX = rand.nextFloat() * (maxX - minX) +
        minX; return finalX;
    }

    public int getX() {
        return x;
    }

    public int getY() {
        return y;
    }
}
```

GameView.java

```
public class GameView extends SurfaceView implements Runnable {

    volatile boolean playing;
    private Thread gameThread = null;
    private Player player;

    private Paint paint;
    private Canvas canvas;
    private SurfaceHolder surfaceHolder;

    //Adding an stars list
    private ArrayList<Star> stars = new
            ArrayList<Star>();
```

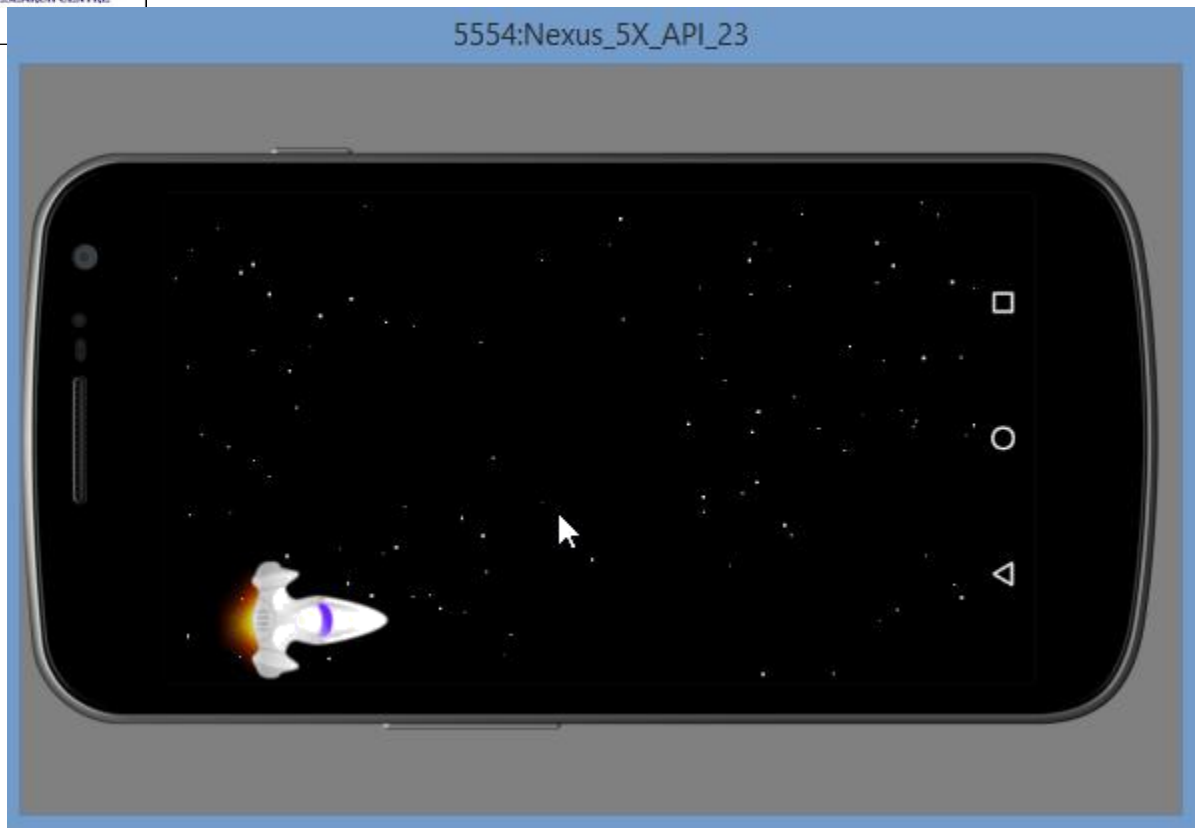
```
public GameView(Context context, int screenX, int screenY) {
    super(context);
```

```
    player = new Player(context, screenX, screenY);

    surfaceHolder = getHolder();
    paint = new Paint();

    //adding 100 stars you may increase the number
    int starNums = 100;
    for (int i = 0; i < starNums; i++) {
        Star s = new Star(screenX, screenY);
        stars.add(s);
    }
}

@Override
public void run() {
    while (playing) {
        update();
        draw();
        control();
    }
}

private void update() {
    player.update();

    //Updating the stars with player speed
    for (Star s : stars) {
        s.update(player.getSpeed());
    }
}

private void draw() {
    if (surfaceHolder.getSurface().isValid()) {
        canvas = surfaceHolder.lockCanvas();
        canvas.drawColor(Color.BLACK);

        //setting the paint color to white to draw the stars
        paint.setColor(Color.WHITE);

        //drawing all stars
        for (Star s : stars) {
            paint.setStrokeWidth(s.getStarWidth());
```

```
        canvas.drawPoint(s.getX(), s.getY(), paint);
    }
```

```
        canvas.drawBitmap(
                player.getBitmap(),
                player.getX(),
                player.getY(),
                paint);
        surfaceHolder.unlockCanvasAndPost(canvas);
    }
}

private void control() {
    try {
        gameThread.sleep(17);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}

public void pause() {
    playing = false;
    try {
        gameThread.join();
    } catch (InterruptedException e) {
    }
}

public void resume() {
    playing = true;
    gameThread = new Thread(this);
    gameThread.start();
}


@Override
public boolean onTouchEvent(MotionEvent motionEvent) {
    switch (motionEvent.getAction() & MotionEvent.ACTION_MASK)
        { case MotionEvent.ACTION_UP:
            player.stopBoosting();
            break;
        case MotionEvent.ACTION_DOWN:
            player.setBoosting();
            break;
    }
    return true;
```

```
    }
}
```



Create a new java class named Enemy and write the following code.

```
package net.simplifiedcoding.spacefighter;

import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Rect;

import java.util.Random;

public class Enemy {

    //bitmap for the enemy
    //we have already pasted the bitmap in the drawable folder
    private Bitmap bitmap;

    //x and y coordinates
    private int x;
    private int y;
```

```
//enemy speed
private int speed = 1;

//min and max coordinates to keep the enemy inside the screen
private int maxX;
private int minX;

private int maxY;
private int minY;


public Enemy(Context context, int screenX, int screenY)
   { //getting bitmap from drawable resource
   bitmap = BitmapFactory.decodeResource(context.getResources(), R.drawable.enemy);

   //initializing min and max coordinates
   maxX = screenX;
   maxY = screenY;
   minX = 0;
   minY = 0;

   //generating a random coordinate to add enemy
   Random generator = new Random();
   speed = generator.nextInt(6) + 10;
   x = screenX;
   y = generator.nextInt(maxY) - bitmap.getHeight();

}

public void update(int playerSpeed) {
   //decreasing x coordinate so that enemy will move right to left
   x -= playerSpeed;
   x -= speed;
   //if the enemy reaches the left edge
   if (x < minX - bitmap.getWidth()) {
      //adding the enemy again to the right edge
      Random generator = new Random();
      speed = generator.nextInt(10) + 10;
      x = maxX;
      y = generator.nextInt(maxY) - bitmap.getHeight();
```

```
    }
}
```

```
    //getters
    public Bitmap getBitmap() {
        return bitmap;
    }

    public int getX() {
        return x;
    }

    public int getY() {
        return y;
    }

    public int getSpeed() {
        return speed;
    }

}
```

We need to add the enemies in the GameView now. So come inside GameView.java and modify the code as follows.

```
public class GameView extends SurfaceView implements Runnable {

    volatile boolean playing;
    private Thread gameThread = null;
    private Player player;

    private Paint paint;
    private Canvas canvas;
    private SurfaceHolder surfaceHolder;

    //Adding enemies object array
    private Enemy[] enemies;

    //Adding 3 enemies you may increase the size
    private int enemyCount = 3;

    private ArrayList<Star> stars = new
            ArrayList<Star>();
```

```
public GameView(Context context, int screenX, int screenY) {
    super(context);
    player = new Player(context, screenX, screenY);
```

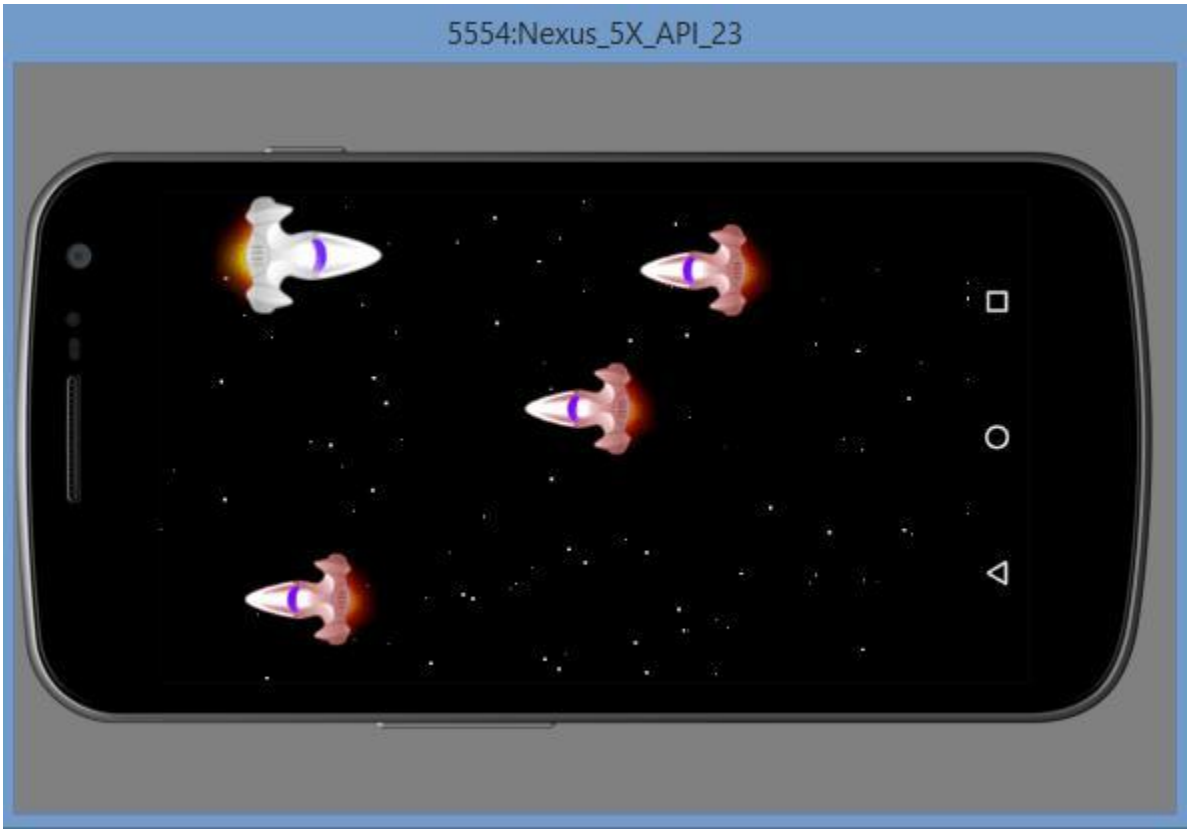
```
    surfaceHolder = getHolder();
    paint = new Paint();

    int starNums = 100;
    for (int i = 0; i < starNums; i++) {
        Star s = new Star(screenX, screenY);
        stars.add(s);
    }

    //initializing enemy object array
    enemies = new Enemy[enemyCount];
    for(int i=0; i<enemyCount; i++){
        enemies[i] = new Enemy(context, screenX, screenY);
    }
}

@Override
public void run() {
    while (playing) {
        update();
        draw();
        control();
    }
}

private void update() {
    player.update();
    for (Star s : stars) {
        s.update(player.getSpeed());
    }

    //updating the enemy coordinate with respect to player speed
    for(int i=0; i<enemyCount; i++){
        enemies[i].update(player.getSpeed());
    }
}

private void draw() {
    if (surfaceHolder.getSurface().isValid()) {
        canvas = surfaceHolder.lockCanvas();
        canvas.drawColor(Color.BLACK);

        paint.setColor(Color.WHITE);
```

```
for (Star s : stars) {
```

```
            paint.setStrokeWidth(s.getStarWidth());
            canvas.drawPoint(s.getX(), s.getY(), paint);
        }

        canvas.drawBitmap(
                player.getBitmap(),
                player.getX(),
                player.getY(),
                paint);

        //drawing the enemies
        for (int i = 0; i < enemyCount; i++) {
            canvas.drawBitmap(
                    enemies[i].getBitmap(),
                    enemies[i].getX(),
                    enemies[i].getY(),
                    paint
            );
        }

        surfaceHolder.unlockCanvasAndPost(canvas);

    }
}

private void control() {
    try {
        gameThread.sleep(17);
    } catch (InterruptedException e)
        { e.printStackTrace();
    }
}

public void pause() {
    playing = false;
    try {
        gameThread.join();
    } catch (InterruptedException e) {
    }
}

public void resume() {
    playing = true;
```

```
gameThread = new Thread(this);
gameThread.start();
```

```
    }

    @Override
    public boolean onTouchEvent(MotionEvent motionEvent) {
        switch (motionEvent.getAction() & MotionEvent.ACTION_MASK)
            { case MotionEvent.ACTION_UP:
                player.stopBoosting();
                break;
            case MotionEvent.ACTION_DOWN:
                player.setBoosting();
                break;
        }
        return true;
    }
}
```



Detecting Collision

```
public class Enemy {
    private Bitmap bitmap;
    private int x;
```

```
    private int y;
    private int speed = 1;
```

```
private int maxX;
private int minX;

private int maxY;
private int minY;

//creating a rect object
private Rect detectCollision;

public Enemy(Context context, int screenX, int screenY) {
   bitmap = BitmapFactory.decodeResource(context.getResources(), R.drawable.enemy);
   maxX = screenX;
   maxY = screenY;
   minX = 0;
   minY = 0;
   Random generator = new Random();
   speed = generator.nextInt(6) + 10;
   x = screenX;
   y = generator.nextInt(maxY) - bitmap.getHeight();

   //initializing rect object
   detectCollision = new Rect(x, y, bitmap.getWidth(), bitmap.getHeight());
}

public void update(int playerSpeed) {
   x -= playerSpeed;
   x -= speed;
   if (x < minX - bitmap.getWidth()) {
      Random generator = new Random();
      speed = generator.nextInt(10) + 10;
      x = maxX;
      y = generator.nextInt(maxY) - bitmap.getHeight();
   }

   //Adding the top, left, bottom and right to the rect object
   detectCollision.left = x;
   detectCollision.top = y;
   detectCollision.right = x + bitmap.getWidth();
   detectCollision.bottom = y +
   bitmap.getHeight();
}
```

```
//adding a setter to x coordinate so that we can change it after collision
public void setX(int x){
    this.x = x;
```

```
    }

    //one more getter for getting the rect object
    public Rect getDetectCollision() {
        return detectCollision;
    }

    //getters
    public Bitmap getBitmap() {
        return bitmap;
    }

    public int getX() {
        return x;
    }

    public int getY() {
        return y;
    }

    public int getSpeed() {
        return speed;
    }

}
```

Player.java

```
public class Player {
    private Bitmap bitmap;
    private int x;
    private int y;
    private int speed = 0;
    private boolean boosting;
    private final int GRAVITY = -10;
    private int maxY;
    private int minY;

    private final int MIN_SPEED = 1;
    private final int MAX_SPEED = 20;

    private Rect detectCollision;
```

```
public Player(Context context, int screenX, int screenY)
    { x = 75;
```

```
    y = 50;
    speed = 1;
    bitmap = BitmapFactory.decodeResource(context.getResources(), R.drawable.player);
    maxY = screenY - bitmap.getHeight();
    minY = 0;
    boosting = false;

    //initializing rect object
    detectCollision = new Rect(x, y, bitmap.getWidth(), bitmap.getHeight());
}

public void setBoosting() {
    boosting = true;
}

public void stopBoosting() {
    boosting = false;
}

public void update() {
    if (boosting) {
        speed += 2;
    } else {
        speed -=
        5;
    }

    if (speed > MAX_SPEED) {
        speed = MAX_SPEED;
    }

    if (speed < MIN_SPEED) {
        speed = MIN_SPEED;
    }

    y -= speed + GRAVITY;

    if (y < minY) {
        y = minY;
    }
    if (y > maxY) {
        y = maxY;
    }
```

```
//adding top, left, bottom and right to the rect object
detectCollision.left = x;
```

```
        detectCollision.top = y;
        detectCollision.right = x + bitmap.getWidth();
        detectCollision.bottom = y + bitmap.getHeight();

    }

    //one more getter for getting the rect object
    public Rect getDetectCollision() {
        return detectCollision;
    }

    public Bitmap getBitmap() {
        return bitmap;
    }

    public int getX() {
        return x;
    }

    public int getY() {
        return y;
    }

    public int getSpeed() {
        return speed;
    }
}
```

Now to complete the collision detection, again to inside GameView.java file and modify the update() method as follows.

```
    private void update() {
        player.update();
        for (Star s : stars) {
            s.update(player.getSpeed());
        }

        for(int i=0; i<enemyCount; i++){
            enemies[i].update(player.getSpeed());

            //if collision occurrs with player
            if (Rect.intersects(player.getDetectCollision(), enemies[i].getDetectCollision()))
                { //moving enemy outside the left edge
                enemies[i].setX(-200);
```

```
    }
}
```

```
    }
```

Adding Blast Effect

Boom.java

```
package net.simplifiedcoding.spacefighter;

import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;

public class Boom {

    //bitmap object
    private Bitmap bitmap;

    //coordinate variables
    private int x;
    private int y;

    //constructor
    public Boom(Context context) {
        //getting boom image from drawable resource
        bitmap = BitmapFactory.decodeResource
                (context.getResources(), R.drawable.boom);

        //setting the coordinate outside the screen
        //so that it won't shown up in the screen
        //it will be only visible for a fraction of second
        //after collission
        x = -250;
        y = -250;
    }

    //setters for x and y to make it visible at the place of collision
    public void setX(int x) {
        this.x = x;
    }

    public void setY(int y) {
        this.y = y;
    }
```

```
//getters
```

```
    public Bitmap getBitmap() {
        return bitmap;
    }

    public void setBitmap(Bitmap bitmap) {
        this.bitmap = bitmap;
    }

    public int getX() {
        return x;
    }

    public int getY() {
        return y;
    }

}
```

Now again come inside GameView.java file and modify the code as follow.

```
public class GameView extends SurfaceView implements Runnable {

    volatile boolean playing;
    private Thread gameThread = null;
    private Player player;

    private Paint paint;
    private Canvas canvas;
    private SurfaceHolder surfaceHolder;

    private Enemy[] enemies;

    private int enemyCount = 3;

    private ArrayList<Star> stars = new
            ArrayList<Star>();

    //defining a boom object to display blast
    private Boom boom;

    public GameView(Context context, int screenX, int screenY) {
        super(context);
        player = new Player(context, screenX, screenY);
```

```
surfaceHolder = getHolder();
```

```
    paint = new Paint();

    int starNums = 100;
    for (int i = 0; i < starNums; i++) {
        Star s = new Star(screenX, screenY);
        stars.add(s);
    }

    enemies = new Enemy[enemyCount];
    for (int i = 0; i < enemyCount; i++) {
        enemies[i] = new Enemy(context, screenX, screenY);
    }

    //initializing boom object
    boom = new Boom(context);
}

@Override
public void run() {
    while (playing) {
        update();
        draw();
        control();
    }
}

private void update() {
    player.update();

    //setting boom outside the screen
    boom.setX(-250);
    boom.setY(-250);

    for (Star s : stars) {
        s.update(player.getSpeed());
    }

    for (int i = 0; i < enemyCount; i++) {
        enemies[i].update(player.getSpeed());

        //if collision occurrs with player
        if (Rect.intersects(player.getDetectCollision(), enemies[i].getDetectCollision())) {
```

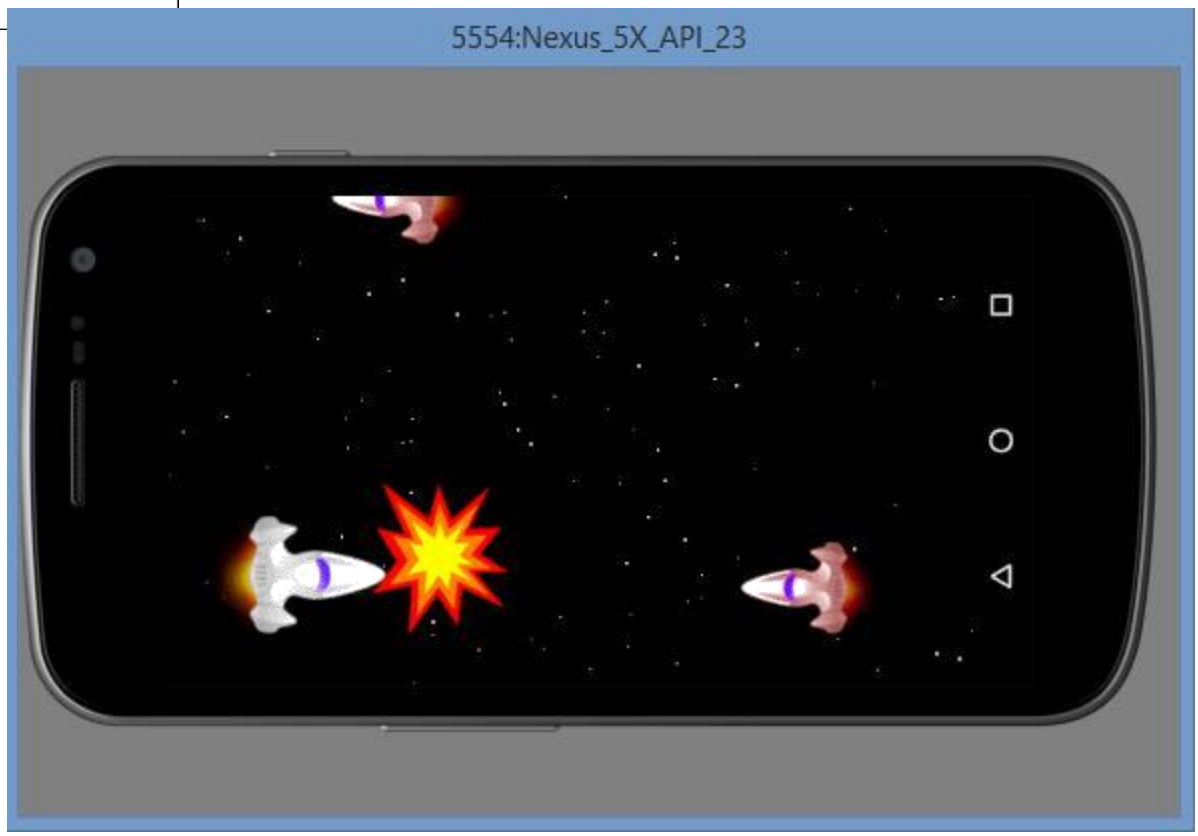
```
//displaying boom at that location
boom.setX(enemies[i].getX());
```

```
            boom.setY(enemies[i].getY());

            enemies[i].setX(-200);

        }
    }
}

private void draw() {
    if (surfaceHolder.getSurface().isValid()) {
        canvas = surfaceHolder.lockCanvas();
        canvas.drawColor(Color.BLACK);

        paint.setColor(Color.WHITE);

        for (Star s : stars) {
            paint.setStrokeWidth(s.getStarWidth());
            canvas.drawPoint(s.getX(), s.getY(), paint);
        }

        canvas.drawBitmap(
                player.getBitmap(),
                player.getX(),
                player.getY(),
                paint);

        for (int i = 0; i < enemyCount; i++) {
            canvas.drawBitmap(
                    enemies[i].getBitmap(),
                    enemies[i].getX(),
                    enemies[i].getY(),
                    paint
            );
        }

        //drawing boom image
        canvas.drawBitmap(
                boom.getBitmap(),
                boom.getX(),
                boom.getY(),
                paint
        );
```

```
surfaceHolder.unlockCanvasAndPost(canvas);
```

```
        }
    }

    private void control() {
        try {
            gameThread.sleep(17);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }

    public void pause() {
        playing = false;
        try {
            gameThread.join();
        } catch (InterruptedException e) {
        }
    }

    public void resume() {
        playing = true;
        gameThread = new Thread(this);
        gameThread.start();
    }


    @Override
    public boolean onTouchEvent(MotionEvent motionEvent) {
        switch (motionEvent.getAction() & MotionEvent.ACTION_MASK)
            { case MotionEvent.ACTION_UP:
                player.stopBoosting();
                break;
            case MotionEvent.ACTION_DOWN:
                player.setBoosting();
                break;
        }
        return true;
    }
}
```

Now again execute the application and you will see a blast effect on collision.


5554:Nexus_5X_API_23

VIVA QUESTIONS

1. What are the different versions of Android OS that you remember?

Version	Name
Android 8.0	Oreo
Android 7.0 – 7.1.2	Nougat
Android 6 – 6.0.1	Marshmallow
Android 5 – 5.1.1	Lollipop
Android 4.4 – 4.4.4	KitKat
Android 4.1 – 4.3	Jelly Bean
Android 4.0-4.0.4	Ice Cream Sandwich

2. Which components are necessary for a New Android project?

Answer: Whenever a new Android project is created, the below components are required:

manifest: It contains an XML file.
build/: It contains build output.
src/: It contains the code and resource files.
res/: It contains bitmap images, UI Strings and XML Layout i.e. all non-code resources.
assets/: It contains a file that should be compiled into a .apk file.

3. What is an Intent?

Answer: Android has an Intent class when the user has to navigate from one activity to another. Intent displays notifications from the device to the user and then the user can respond to the notification if required.
Given below are the two types:

I. Implicit Intents
II. Explicit Intents

4. What is .apk extension in Android?

Answer: It is a default file format that is used by the Android Operating System. Application Package Kit (APK) is used for the installation of mobile apps. The .apk contains resource file, certificate, manifest file, and other code.

5. Best Android Games of 2019

I. Stardew Valley.
II. Graveyard Keeper.
III. PUBG Mobile.

V. Oddmar.

VI.Holedown.

VII. BattleChasers: NightWar.

EXPERIMENT – 10

Create an application to handle images and videos according to size.

Util.java

```
_public class Util {
//SDF to generate a unique name for the compressed file.
public static final SimpleDateFormat SDF = new SimpleDateFormat(―yyyymmddhhmmss‖,
Locale.getDefault());
/*
compress the file/photo from @param <b>path</b> to a private location on the current device and return
the compressed file.
@param path = The original image path
@param context = Current android Context
*/
public static File getCompressed(Context context, String path) throws IOException
{ if(context == null)
throw new NullPointerException(―Context must not be null.‖);
//getting device external cache directory, might not be available on some devices,
// so our code fall back to internal storage cache directory, which is always available but in smaller
quantity
File cacheDir = context.getExternalCacheDir();
if(cacheDir == null)
//fall back
cacheDir = context.getCacheDir();
String rootDir = cacheDir.getAbsolutePath() + ―/ImageCompressor‖;
File root = new File(rootDir);
//Create ImageCompressor folder if it doesnt already exists.
if(!root.exists())
root.mkdirs();
//decode and resize the original bitmap from @param path.
Bitmap bitmap = decodeImageFromFiles(path, /* your desired width*/300, /*your desired height*/ 300);
//create placeholder for the compressed image file
File compressed = new File(root, SDF.format(new Date()) + ―.jpg‖ /*Your desired format*/);
//convert the decoded bitmap to stream
ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream();
/*compress bitmap into byteArrayOutputStream
Bitmap.compress(Format, Quality,
OutputStream) Where Quality ranges from 1–
100.
```

```
*/
bitmap.compress(Bitmap.CompressFormat.JPEG, 80,
byteArrayOutputStream); /*
```

Right now, we have our bitmap inside byteArrayOutputStream Object, all we need next is to write it to the compressed file we created earlier,
java.io.FileOutputStream can help us do just That!
*/
FileOutputStream fileOutputStream = new FileOutputStream(compressed);
fileOutputStream.write(byteArrayOutputStream.toByteArray());
fileOutputStream.flush();
fileOutputStream.close();
//File written, return to the caller. Done!
return compressed;
}
public static Bitmap decodeImageFromFiles(String path, int width, int height)
{ BitmapFactory.Options scaleOptions = new BitmapFactory.Options();
scaleOptions.inJustDecodeBounds = true; BitmapFactory.decodeFile(path,
scaleOptions); int scale = 1;

while (scaleOptions.outWidth / scale / 2 >= width
&& scaleOptions.outHeight / scale / 2 >= height)
{ scale *= 2;
}
// decode with the sample size
BitmapFactory.Options outOptions = new
BitmapFactory.Options(); outOptions.inSampleSize = scale;
return BitmapFactory.decodeFile(path, outOptions);
}
}‘

ImageCompressTask.java

public class ImageCompressTask implements Runnable
{ private Context mContext;
private List<String> originalPaths = new ArrayList<>();
private Handler mHandler = new Handler(Looper.getMainLooper());
private List<File> result = new ArrayList<>();
private IImageCompressTaskListener mIImageCompressTaskListener;
public ImageCompressTask(Context context, String path, IImageCompressTaskListener
compressTaskListener) {
originalPaths.add(path);
mContext = context;
mIImageCompressTaskListener = compressTaskListener;
}
public ImageCompressTask(Context context, List<String> paths, IImageCompressTaskListener
compressTaskListener) {

```
originalPaths = paths;
mContext = context;
```

```
mIImageCompressTaskListener = compressTaskListener;
}
@Override
public void run() {
try {
//Loop through all the given paths and collect the compressed file from Util.getCompressed(Context,
String)
for (String path : originalPaths) {
File file = Util.getCompressed(mContext, path);
//add it!
result.add(file);
}
//use Handler to post the result back to the main
Thread mHandler.post(new Runnable() { @Override

public void run() {
if(mIImageCompressTaskListener != null)
mIImageCompressTaskListener.onComplete(result);
}
});
}catch (final IOException ex) {
//There was an error, report the error back through the
callback mHandler.post(new Runnable() { @Override

public void run() {
if(mIImageCompressTaskListener != null)
mIImageCompressTaskListener.onError(ex);
}
});
}
}
}‘
```

Finally, create MainActivity.java, the UI for the whole sample App.

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
Button selectImage;
ImageView selectedImage;
private static final int REQUEST_STORAGE_PERMISSION =
100; private static final int REQUEST_PICK_PHOTO = 101;
//create a single thread pool to our image compression class.
```

```
private ExecutorService mExecutorService = Executors.newFixedThreadPool(1);
private ImageCompressTask imageCompressTask;
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
selectedImage = (ImageView) findViewById(R.id.iv_selected_photo);
selectImage = (Button) findViewById(R.id.btn_select_image);
selectImage.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
requestPermission();
}
});
}
void requestPermission() {
if(PackageManager.PERMISSION_GRANTED !=
ContextCompat.checkSelfPermission(this, Manifest.permission.WRITE_EXTERNAL_STORAGE))
{
if(ActivityCompat.shouldShowRequestPermissionRationale(this,
Manifest.permission.WRITE_EXTERNAL_STORAGE)) {
ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE},
REQUEST_STORAGE_PERMISSION);
}else {
//Yeah! I want both block to do the same thing, you can write your own logic, but this works for me.
ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE},
REQUEST_STORAGE_PERMISSION);
}
}else {
//Permission Granted, lets go pick photo
Intent intent = new Intent(Intent.ACTION_PICK);
intent.setAction(Intent.ACTION_GET_CONTENT);
intent.setType(―image/*‖);
startActivityForResult(intent, REQUEST_PICK_PHOTO);
}
}
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data)
{ super.onActivityResult(requestCode, resultCode, data);
if(requestCode == REQUEST_PICK_PHOTO && resultCode == RESULT_OK &&
data != null) {
//extract absolute image path from Uri
Uri uri = data.getData();
```

```
Cursor cursor = MediaStore.Images.Media.query(getContentResolver(), uri,
new String[]{MediaStore.Images.Media.DATA}); if(cursor != null) {
```

```
String path = cursor.getString(cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA));
//Create ImageCompressTask and execute with Executor.
imageCompressTask = new ImageCompressTask(this, path, iImageCompressTaskListener);
mExecutorService.execute(imageCompressTask); }

}
}
//image compress task callback
private IImageCompressTaskListener iImageCompressTaskListener = new
IImageCompressTaskListener() {
@Override
public void onComplete(List<File> compressed)
{ //photo compressed. Yay! //prepare for uploads.

File file = compressed.get(0);
selectedImage.setImageBitmap(BitmapFactory.decodeFile(file.getAbsolutePath()));
}
@Override
public void onError(Throwable error) {
//very unlikely, but it might happen on a device with extremely low storage.
//log it, log.WhatTheFuck?, or show a dialog asking the user to delete some files….etc, etc
Log.wtf(―ImageCompressor‖, ―Error occurred‖, error);
}
};
@Override
protected void onDestroy() {
super.onDestroy();
//clean up!
mExecutorService.shutdown();
mExecutorService = null;
imageCompressTask = null;
}
}‘
```

VIVA QUESTIONS

1. What Is File Compression? Why Is It Necessary To Compress Files?

Answer :

File compression is a process to reduce the disk space to store that file.
File compression enables data to be transferred quickly.

2. **How to Request storage permission?** Answer :

Answer: permissions.checkpermissions(getBaseContext());