**VIDYA JYOTHI INSTITUTE OF TECHNOLOGY**

**(Autonomous)**

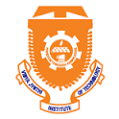
**Aziz Nagar Gate, C.B. Post, Hyderabad-75**

**Department of Computer science & Engineering**

**Year : IV B.Tech Semester : I**

MOBILE APPLICATION DEVELOPMENT

(R18)



LAB MANUAL

**List of Experiments**

|  |  |  |
| --- | --- | --- |
| **Week 1:** | Installation and configuration of Android Studio | **3-12** |
| **Week 2** | Develop an application that uses GUI components to display a “Hello World” message and change its color and font size. | 13-15 |
| **Week 3** | Develop an application that receives user’s name, contact and city and displays the same using Layout Managers and Event Listeners. | 16-20 |
| **Week 4:** | Create a native calculator application. | 21-24 |
| **Week 5** | Design an application that draws basic graphical primitives: line, circle, square, rectangle etc., on the screen | 25-27 |
| **Week 6:** | Develop a Registration and Login application that makes use of database. | 28-34 |
| **Week 7** | Develop an application that makes use of RSS Feed | 35-54 |
| **Week 8:** | Create an application that implements Multi threading. | 55-57 |
| **Week 9:** | Develop a native application that uses GPS location information. | 58-62 |
| **Week 10:** | Implement an application that writes data to the SD card. | 63-66 |
| **Week 11:** | Develop an application that creates notification upon receiving a message. | 67-69 |
| **Week 12:** | Create an alarm clock mobile application. | 70-74 |

**Week1: Installation and configuration of Android Studio**

Aim: Installation and configuration of Android Studio

**Step 1 - System Requirements**

The required tools to develop Android applications are open source and can be downloaded from the Web. Following is the list of software's you will need before you start your Android application programming.

Java JDK5 or later version

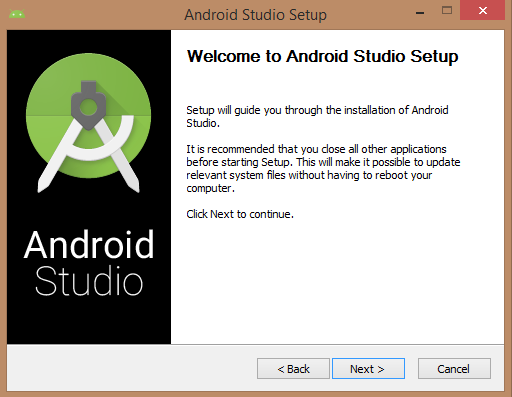
Java Runtime Environment (JRE) 6 Android Studio

**Step 2 - Setup Android Studio**

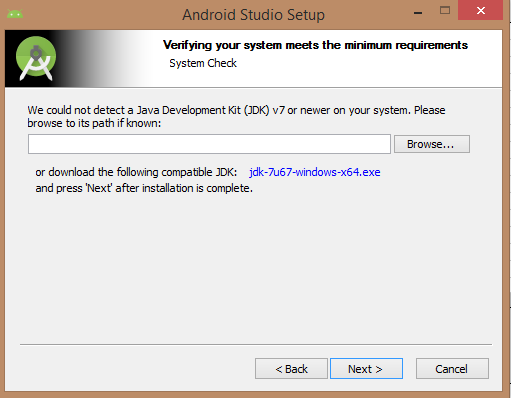
Android Studio is the official IDE for android application development.It works based on IntelliJ IDEA, You can download the latest version of android studio from [Android Studio 2.2 Download,](https://developer.android.com/sdk/index.html) If you are new to installing Android Studio on windows,you will find a file, which is named as android-studio-bundle-143.3101438-windows.exe.So just download and run on windows machine according to android studio wizard guideline.

If you are installing Android Studio on Mac or Linux, You can download the latest version from [Android Studio Mac Download,](https://dl.google.com/dl/android/studio/install/1.1.0/android-studio-ide-135.1740770-mac.dmg)or [Android Studio Linux Download,](https://dl.google.com/dl/android/studio/ide-zips/1.1.0/android-studio-ide-135.1740770-linux.zip) check the instructions provided along with the downloaded file for Mac OS and Linux. This tutorial will consider that you are going to setup your environment on Windows machine having Windows 8.1 operating system. Installation

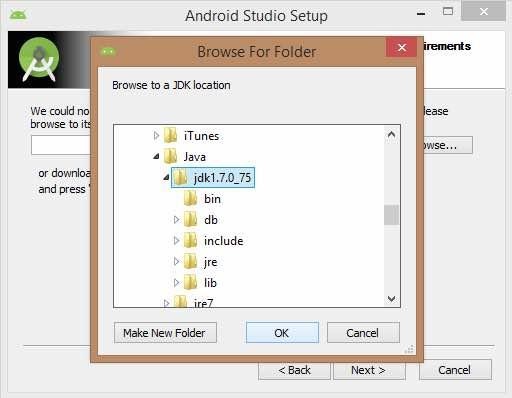
So let's launch Android Studio.exe,Make sure before launch Android Studio, Our Machine should required installed Java JDK. To install Java JDK,take a references of [Android environment setup](https://www.tutorialspoint.com/android/android_environment_setup.htm)



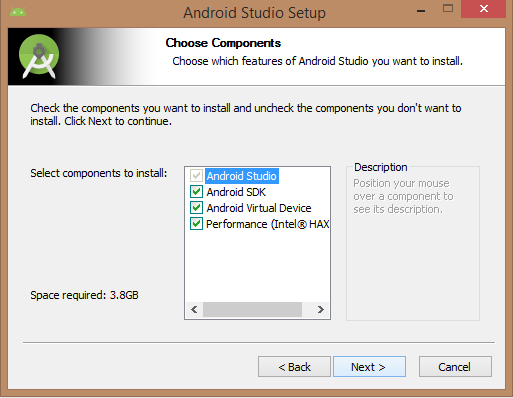
Once you launched Android Studio, its time to mention JDK7 path or later version in android studio installer.



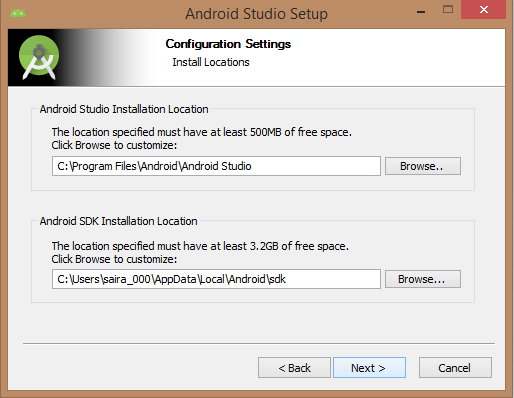
Below is the image initiating JDK to android SDK



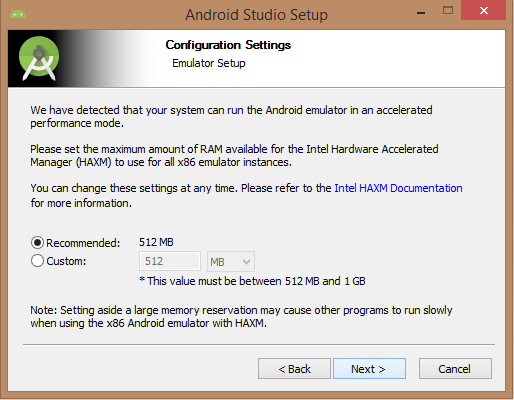
Need to check the components, which are required to create applications, below the image has selected Android Studio, Android SDK, Android Virtual Machine and performance(Intel chip).



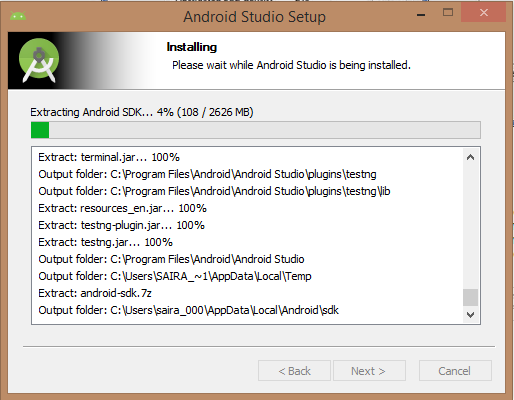
Need to specify the location of local machine path for Android studio and Android SDK, below the image has taken default location of windows 8.1 x64 bit architecture.



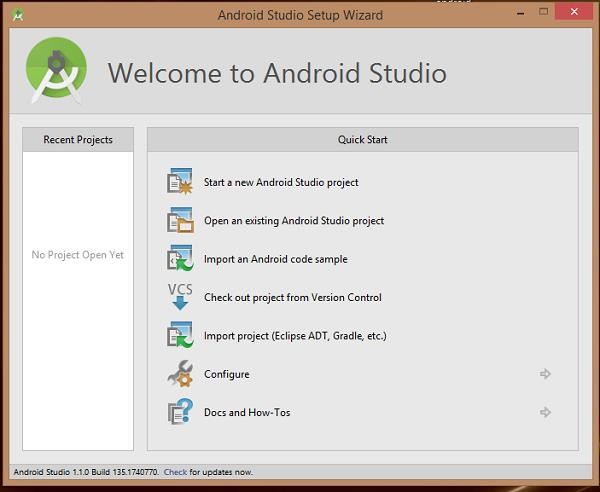
Need to specify the ram space for Android emulator by default it would take 512MB of local machine RAM.



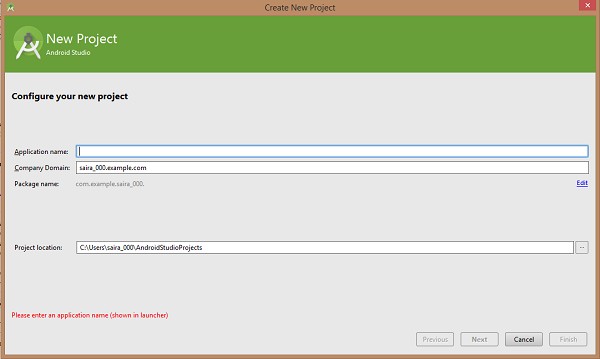
At final stage, it would extract SDK packages into our local machine, it would take a while time to finish the task and would take 2626MB of Hard disk space.



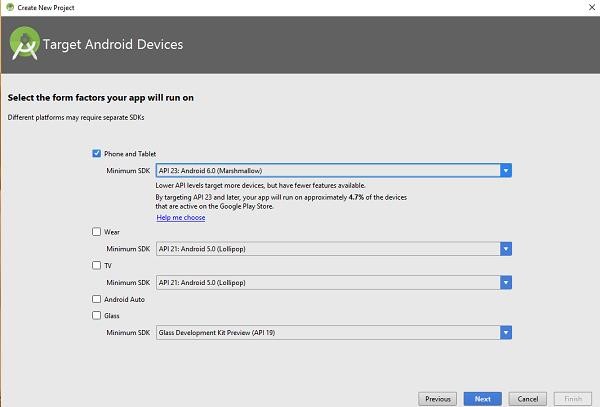
After done all above steps perfectly, you must get finish button and it gonna be open android studio project with Welcome to android studio message as shown below



You can start your application development by calling start a new android studio project. in a new installation frame should ask Application name, package information and location of the project.



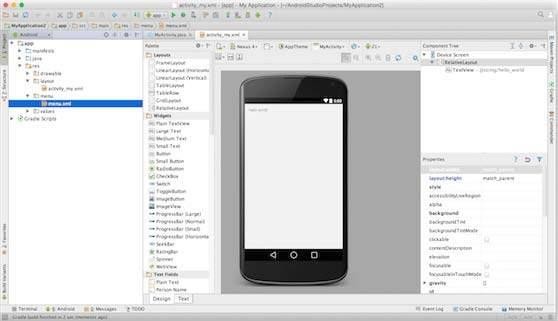
After entered application name, it going to be called select the form factors your application runs on, here need to specify Minimum SDK, in our tutorial, I have declared as API23: Android 6.0(Mashmallow)



The next level of installation should contain selecting the activity to mobile, it specifies the default layout for Applications

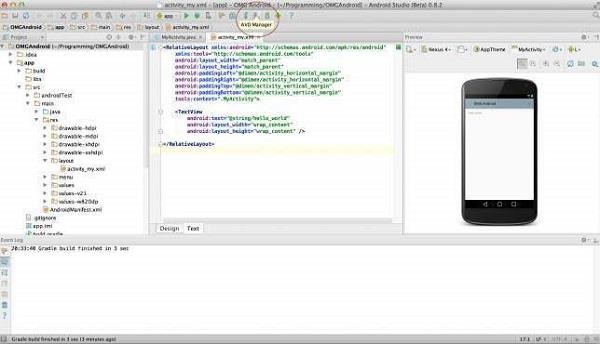


At the final stage it going to be open development tool to write the application code.

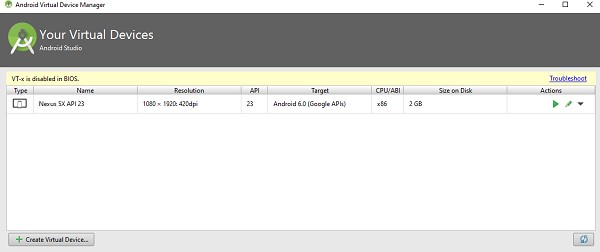


**Step 3 - Create Android Virtual Device**

To test your Android applications, you will need a virtual Android device. So before we start writing our code, let us create an Android virtual device. Launch Android AVD Manager Clicking AVD\_Manager icon as shown below



After Click on a virtual device icon, it going to be shown by default virtual devices which are present on your SDK, or else need to create a virtual device by clicking Create new Virtual device button



If your AVD is created successfully it means your environment is ready for Android application development. If you like, you can close this window using top-right cross button. Better you re- start your machine and once you are done with this last step, you are ready to proceed for your first Android example but before that we will see few more important concepts related to Android Application Development.

**Week 2**

**Aim: Develop an application that uses GUI components to display a “Hello World” message and change its color and font size.**

**Procedure:**

1) Open eclipse or android studio and select new android project

2) Give project name and select next

1. Choose the android version.Choose the lowest android version(Android 2.2) and select next

4) Enter the package name.package name must be two word seprated by comma and click finish

5) Go to package explorer in the left hand side.select our project.

6) Go to res folder and select layout.Double click the main.xml file

7) Now you can see the Graphics layout window.

1. Click the main.xml file and type the code below

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

<LinearLayout xmlns:android[="http](http://schemas.android.com/apk/res/android):/[/schemas.android.com/apk/res/android](http://schemas.android.com/apk/res/android)" android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent" android:orientation="vertical" >

<TextView android:id="@+id/textView1" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:layout\_margin="20sp" android:gravity="center" android:text="HELLO WORLD" android:textSize="20sp" android:textStyle="bold" />

<Button android:id="@+id/button1"

android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:gravity="center" android:text="Change font size" android:textSize="20sp" />

<Button android:id="@+id/button2"

android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:gravity="center" android:text="Change color" android:textSize="20sp" />

<Button android:id="@+id/button3"

android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:gravity="center" android:text="Change font" android:textSize="20sp" />

</LinearLayout>

1. Again click the graphics layout tab and screen layout is look like below
2. Go to project explorer and select src folder.Now select mainactivity.java file and type the

following code

PROGRAM

//import android.R;

import android.app.Activity; import android.graphics.Color; import android.graphics.Typeface; import android.os.Bundle;

import android.view.View; import android.widget.Button; import android.widget.TextView;

public class AndroidActivity extends Activity { float font =24;

int i=1; @Override

public void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.main);

final TextView t1=(TextView) findViewById(R.id.textView1); Button b1 = (Button) findViewById(R.id.button1); b1.setOnClickListener(new View.OnClickListener() {

public void onClick(View view) { t1.setTextSize(font);

font=font+4; if(font==40) font=20;

}

});

Button b2 = (Button) findViewById(R.id.button2); b2.setOnClickListener(new View.OnClickListener() { public void onClick(View view) {

switch(i)

{

case 1: t1.setTextColor(Color.parseColor("#0000FF")); break;

case 2: t1.setTextColor(Color.parseColor("#00FF00")); break;

case 3: t1.setTextColor(Color.parseColor("#FF0000")); break;

case 4: t1.setTextColor(Color.parseColor("#800000")); break;

} i++;

if(i==5) i=1;

}

});

}

}

1. Now go to main.xml and right click .select run as option and select run configuration
2. Android output is present in the android emulator as shown in below.

# Output:



**Week 3**

**Aim: Develop an application that receives user’s name, contact and city and displays the same using Layout Managers and Event Listeners.**

**Procedure:**

1)Open android studio and select new android project

2)Give project name and select next

3) Choose the android version.Choose the lowest android version(Android 2.2) and select next

4) Enter the package name.package name must be two word seprated by comma and click finish

5) Go to package explorer in the left hand side.select our project.

6) Go to res folder and select layout.Double click the main.xml file.Add the code below

<RelativeLayout xmlns:android[="http://sch](http://schemas.android.com/apk/res/android)em[as.android.com/apk/res/android](http://schemas.android.com/apk/res/android)" android:id="@+id/relativeLayout1"

android:layout\_width="fill\_parent" android:layout\_height="fill\_parent" >

<LinearLayout android:id="@+id/linearLayout1" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignParentLeft="true" android:layout\_alignParentRight="true" android:layout\_alignParentTop="true" >

<TextView android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_gravity="center"

android:text="ADDITION" android:textSize="20dp" >

</TextView>

</LinearLayout>

<LinearLayout android:id="@+id/linearLayout2" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignParentLeft="true" android:layout\_alignParentRight="true" android:layout\_below="@+id/linearLayout1" >

<TextView android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"

android:text="ENTER NO 1" >

</TextView>

<EditText android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_weight="0.20" android:id="@+id/edittext1" android:inputType="number">

</EditText>

</LinearLayout>

<LinearLayout android:id="@+id/linearLayout3" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignParentLeft="true" android:layout\_alignParentRight="true" android:layout\_below="@+id/linearLayout2" >

<TextView android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="ENTER NO 2" >

</TextView>

<EditText android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_weight="0.20" android:id="@+id/edittext2" android:inputType="number">

</EditText>

</LinearLayout>

<LinearLayout android:id="@+id/linearLayout4" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignParentLeft="true" android:layout\_alignParentRight="true" android:layout\_below="@+id/linearLayout3" >

<Button android:layout\_width="wrap\_content" android:id="@+id/button1" android:layout\_height="wrap\_content" android:text="Addition" android:layout\_weight="0.50" />

<Button android:layout\_width="wrap\_content" android:id="@+id/button3" android:layout\_height="wrap\_content" android:text="subtraction"

android:layout\_weight="0.50" />

<Button android:layout\_width="wrap\_content" android:id="@+id/button2" android:layout\_height="wrap\_content" android:text="CLEAR" android:layout\_weight="0.50" />

</LinearLayout>

<View android:layout\_height="2px" android:layout\_width="fill\_parent"

android:layout\_below="@+id/linearLayout4" android:background="#DDFFDD"/>

</RelativeLayout>

1. Now select mainactivity.java file and type the following code. package layout.ne;

import android.app.Activity; 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 LAYOUTActivity extends Activity {

/\*\* Called when the activity is first created. \*/ EditText txtData1,txtData2;

float num1,num2,result1,result2; @Override

public void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.main);

Button add = (Button) findViewById(R.id.button1);

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

try

{

txtData1 = (EditText) findViewById(R.id.edittext1); txtData2 = (EditText) findViewById(R.id.edittext2); num1 = Float.parseFloat(txtData1.getText().toString()); num2 = Float.parseFloat(txtData2.getText().toString()); result1=num1+num2;

Toast.makeText(getBaseContext(),"ANSWER:"+result1,Toast.LENGTH\_SHORT). show();

catch(Exception e)

{

Toast.makeText(getBaseContext(), e.getMessage(), Toast.LENGTH\_SHORT).show();

}

}

});

Button sub = (Button) findViewById(R.id.button3); sub.setOnClickListener(new OnClickListener() { public void onClick(View v) {

try

{

txtData1 = (EditText) findViewById(R.id.edittext1); txtData2 = (EditText) findViewById(R.id.edittext2); num1 = Float.parseFloat(txtData1.getText().toString()); num2 = Float.parseFloat(txtData2.getText().toString()); result2=num1-num2;

Toast.makeText(getBaseContext(),"ANSWER:"+result2,Toast.LENGTH\_SHORT). show();}

catch(Exception e)

{

Toast.makeText(getBaseContext(), e.getMessage(), Toast.LENGTH\_SHORT).show();

}

}

});

Button clear = (Button) findViewById(R.id.button2); clear.setOnClickListener(new OnClickListener() { public void onClick(View v) {

try

{

txtData1.setText(""); txtData2.setText("");

}

catch(Exception e)

{

Toast.makeText(getBaseContext(), e.getMessage(), Toast.LENGTH\_SHORT).show();

}

}

});

}

}

1. Now go to main.xml and right click. select run as option and select run configuration
2. Android output is present in the android emulator as shown in below.

# Output:



**Week 4:**

**Aim: Create a native calculator application.**

**Procedure:**

1) Open android studio and select new android project

2) Give project name and select next

3) Choose the android version.Choose the lowest android version(Android 2.2) and select next

1. Enter the package name.package name must be two word seprated by comma and click finish
2. Go to package explorer in the left hand side.select our project.
3. Go to res folder and select layout.Double click the main.xml file.Add the code below

Main.xml coding

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

<LinearLayout xmlns:android[="ht](http://schemas.android.com/apk/res/android)tp[://schemas.android.com/apk/res/android](http://schemas.android.com/apk/res/android)" android:orientation="vertical" android:layout\_width="fill\_parent" android:layout\_height="fill\_parent">

<LinearLayout android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:id="@+id/linearLayout1" android:layout\_marginLeft="10pt" android:layout\_marginRight="10pt" android:layout\_marginTop="3pt">

<EditText android:layout\_weight="1"

android:layout\_height="wrap\_content" android:layout\_marginRight="5pt" android:id="@+id/etNum1" android:layout\_width="match\_parent" android:inputType="numberDecimal">

</EditText>

<EditText android:layout\_height="wrap\_content" android:layout\_weight="1" android:layout\_marginLeft="5pt" android:id="@+id/etNum2" android:layout\_width="match\_parent" android:inputType="numberDecimal">

</EditText>

</LinearLayout>

<LinearLayout android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:id="@+id/linearLayout2" android:layout\_marginTop="3pt" android:layout\_marginLeft="5pt" android:layout\_marginRight="5pt">

<Button android:layout\_height="wrap\_content" android:layout\_width="match\_parent" android:layout\_weight="1" android:text="+" android:textSize="15pt" android:id="@+id/btnAdd">

</Button>

<Button android:layout\_height="wrap\_content"

android:layout\_width="match\_parent" android:layout\_weight="1" android:text="-" android:textSize="15pt" android:id="@+id/btnSub">

</Button>

<Button android:layout\_height="wrap\_content" android:layout\_width="match\_parent" android:layout\_weight="1" android:text="\*" android:textSize="15pt" android:id="@+id/btnMult">

</Button>

<Button android:layout\_height="wrap\_content" android:layout\_width="match\_parent" android:layout\_weight="1" android:text="/" android:textSize="15pt" android:id="@+id/btnDiv">

</Button>

</LinearLayout>

<TextView android:layout\_height="wrap\_content" android:layout\_width="match\_parent" android:layout\_marginLeft="5pt" android:layout\_marginRight="5pt" android:textSize="12pt"

android:layout\_marginTop="3pt" android:id="@+id/tvResult" android:gravity="center\_horizontal">

</TextView>

</LinearLayout>

**MainActivity.java coding**

**package** CALCU.CALU;

import android.app.Activity; import android.os.Bundle; import android.text.TextUtils; import android.view.View;

import android.view.View.OnClickListener; import android.widget.Button;

import android.widget.EditText; import android.widget.TextView;

public class CALCULATORActivity extends Activity implements OnClickListener

{

EditText input1; EditText input2; Button addition; Button subtraction; Button multiplication; Button division; TextView tvResult; String oper = ""; @Override

public void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.main);

input1 = (EditText) findViewById(R.id.etNum1); input2 = (EditText) findViewById(R.id.etNum2);

addition = (Button) findViewById(R.id.btnAdd); subtraction = (Button) findViewById(R.id.btnSub); multiplication = (Button) findViewById(R.id.btnMult); division = (Button) findViewById(R.id.btnDiv);

tvResult = (TextView) findViewById(R.id.tvResult);

// set a listener addition.setOnClickListener(this); subtraction.setOnClickListener(this); multiplication.setOnClickListener(this); division.setOnClickListener(this);

}

@Override

public void onClick(View v) {

// TODO Auto-generated method stub float num1 = 0;

float num2 = 0; float result = 0;

// check if the fields are empty

if (TextUtils.isEmpty(input1.getText().toString())

|| TextUtils.isEmpty(input2.getText().toString())) { return;

}

// read EditText and fill variables with numbers num1 = Float.parseFloat(input1.getText().toString()); num2 = Float.parseFloat(input2.getText().toString());

// defines the button that has been clicked and performs the corresponding operation

// write operation into oper, we will use it later for output switch (v.getId()) {

case R.id.btnAdd: oper = "+";

result = num1 + num2;

break;

case R.id.btnSub:

oper = "-";

result = num1 - num2; break;

case R.id.btnMult:

oper = "\*";

result = num1 \* num2; break;

case R.id.btnDiv:

oper = "/";

result = num1 / num2; break;

default:

break;

}

// form the output line

tvResult.setText(num1 + " " + oper + " " + num2 + " = " + result);

}

}

# Output:



**Week 5**

**Aim: Design an application that draws basic graphical primitives: line, circle, square, rectangle etc., on the screen.**

# Procedure:

1. Open eclipse or android studio and select new android project
2. Give project name and select next
3. Choose the android version.Choose the lowest android version(Android 2.2) and select

next

1. Enter the package name.package name must be two word seprated by comma and click

finish

1. Go to package explorer in the left hand side.select our project.
2. Go to res folder and select layout.Double click the main.xml file.Don't change anything

in layout.Leave as default.

1. Now select MainActivity.java file and type the following code.

package basic.primitive;

import android.app.Activity; import android.content.Context; import android.graphics.Canvas; import android.graphics.Color; import android.graphics.Paint; import android.os.Bundle; import android.view.View;

public class BasicprimitiveActivity extends Activity {

/\*\* Called when the activity is first created. \*/

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState); setContentView(new myview(this));

}

private class myview extends View

{

public myview(Context context)

{

super(context);

}

@Override

protected void onDraw(Canvas canvas)

{

super.onDraw(canvas); Paint paint=new Paint(); paint.setTextSize(40);

paint.setColor(Color.GREEN); canvas.drawText("Circle", 55, 30, paint); paint.setColor(Color.RED); canvas.drawCircle(100, 150,100, paint); paint.setColor(Color.GREEN); canvas.drawText("Rectangle", 255, 30, paint); paint.setColor(Color.YELLOW); canvas.drawRect(250, 50,400,350, paint); paint.setColor(Color.GREEN); canvas.drawText("SQUARE", 55, 430, paint); paint.setColor(Color.BLUE); canvas.drawRect(50, 450,150,550, paint); paint.setColor(Color.GREEN); canvas.drawText("LINE", 255, 430, paint);

paint.setColor(Color.CYAN); canvas.drawLine(250, 500, 350, 500, paint);

}

}

}

1. Now go to main.xml and right click .select run as option and select run configuration
2. Android output is present in the android emulator as shown in below.

# Output:



**Week 6**

**Aim: Develop a Registration and Login application that makes use of database.**

**Procedure:**

1)Open eclipse or android studio and select new android project

2)Give project name and select next

1. Choose the android version.Choose the lowest android version(Android 2.2) and select next
2. Enter the package name.package name must be two word seprated by comma and click finish
3. Go to package explorer in the left hand side.select our project.
4. Go to res folder and select layout.Double click the main.xml file.Add the code below

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

<AbsoluteLayout xmlns:android[="ht](http://schemas.android.com/apk/res/android)tp[://schemas.android.com/apk/res/android](http://schemas.android.com/apk/res/android)" android:id="@+id/myLayout"

android:stretchColumns="0" android:layout\_width="fill\_parent" android:layout\_height="fill\_parent">

<TextView android:text="@string/title" android:layout\_x="110dp" android:layout\_y="10dp" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"/>

<TextView android:text="@string/empid" android:layout\_x="30dp" android:layout\_y="50dp" android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

<EditText android:id="@+id/editEmpid" android:inputType="number" android:layout\_x="150dp" android:layout\_y="50dp" android:layout\_width="150dp" android:layout\_height="40dp"/>

<TextView android:text="@string/name" android:layout\_x="30dp" android:layout\_y="100dp" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"/>

<EditText android:id="@+id/editName" android:inputType="text" android:layout\_x="150dp" android:layout\_y="100dp" android:layout\_width="150dp" android:layout\_height="40dp"/>

<TextView android:text="@string/salary" android:layout\_x="30dp" android:layout\_y="150dp" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"/>

<EditText android:id="@+id/editsalary" android:inputType="number" android:layout\_x="150dp" android:layout\_y="150dp" android:layout\_width="150dp" android:layout\_height="40dp"/>

<Button android:id="@+id/btnAdd" android:text="@string/add"

android:layout\_x="30dp"

android:layout\_y="200dp" android:layout\_width="130dp" android:layout\_height="40dp"/>

<Button android:id="@+id/btnDelete" android:text="@string/delete" android:layout\_x="160dp" android:layout\_y="200dp" android:layout\_width="130dp" android:layout\_height="40dp"/>n

<Button android:id="@+id/btnModify" android:text="@string/modify" android:layout\_x="30dp" android:layout\_y="250dp" android:layout\_width="130dp" android:layout\_height="40dp"/>

<Button android:id="@+id/btnView" android:text="@string/view" android:layout\_x="160dp" android:layout\_y="250dp" android:layout\_width="130dp" android:layout\_height="40dp"/>

<Button android:id="@+id/btnViewAll" android:text="@string/view\_all" android:layout\_x="85dp" android:layout\_y="300dp" android:layout\_width="150dp" android:layout\_height="40dp"/>

</AbsoluteLayout>

1. Go to values folder and select string.xml file.Replace the code below

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

<resources>

<string name="app\_name">Employee detail1</string>

<string name="hello">Hello World, Employee detail Activity!</string>

<string name="title">Employee Details</string>

<string name="empid">Enter Employee ID: </string>

<string name="name">Enter Name: </string>

<string name="salary">Enter salary: </string>

<string name="add">Add Employee</string>

<string name="delete">Delete Employee</string>

<string name="modify">Modify Employee</string>

<string name="view">View Employee</string>

<string name="view\_all">View All Employee</string>

</resources>

1. Now select mainactivity.java file and type the following code.In my coding maniactivity name

is EmployeedetailActivity. package employee.detail;

//import android.R;

import android.app.Activity;

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 EmployeedetailActivity extends Activity implements OnClickListener

{

EditText editEmpid,editName,editsalary;

Button btnAdd,btnDelete,btnModify,btnView,btnViewAll; SQLiteDatabase db;

/\*\* Called when the activity is first created. \*/

@Override

public void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState); setContentView(R.layout.main); editEmpid=(EditText)findViewById(R.id.editEmpid); editName=(EditText)findViewById(R.id.editName); editsalary=(EditText)findViewById(R.id.editsalary); btnAdd=(Button)findViewById(R.id.btnAdd); btnDelete=(Button)findViewById(R.id.btnDelete); btnModify=(Button)findViewById(R.id.btnModify); btnView=(Button)findViewById(R.id.btnView); btnViewAll=(Button)findViewById(R.id.btnViewAll); btnAdd.setOnClickListener(this); btnDelete.setOnClickListener(this); btnModify.setOnClickListener(this); btnView.setOnClickListener(this); btnViewAll.setOnClickListener(this);

db=openOrCreateDatabase("EmployeeDB", Context.MODE\_PRIVATE, null); db.execSQL("CREATE TABLE IF NOT EXISTS employee(empid VARCHAR,name

VARCHAR,salary VARCHAR);");

}

public void onClick(View view)

{

if(view==btnAdd)

{

if(editEmpid.getText().toString().trim().length()==0|| editName.getText().toString().trim().length()==0|| editsalary.getText().toString().trim().length()==0)

{

showMessage("Error", "Please enter all values");

return;

}

db.execSQL("INSERT INTO employee VALUES('"+editEmpid.getText()+"','"+editName.getText()+ "','"+editsalary.getText()+"');");

showMessage("Success", "Record added"); clearText();

}

if(view==btnDelete)

{

if(editEmpid.getText().toString().trim().length()==0)

{

showMessage("Error", "Please enter Employee id"); return;

}

Cursor c=db.rawQuery("SELECT \* FROM employee WHERE empid='"+editEmpid.getText()+"'", null);

if(c.moveToFirst())

{

db.execSQL("DELETE FROM employee WHERE empid='"+editEmpid.getText()+"'"); showMessage("Success", "Record Deleted");

}

else

{

showMessage("Error", "Invalid Employee id"

clearText();

}

if(view==btnModify)

{

if(editEmpid.getText().toString().trim().length()==0)

{

showMessage("Error", "Please enter Employee id"); return;

}

Cursor c=db.rawQuery("SELECT \* FROM employee WHERE empid='"+editEmpid.getText()+"'", null);

if(c.moveToFirst())

{

db.execSQL("UPDATE employee SET name='"+editName.getText()+"',salary='"+editsalary.getText()+ "' WHERE empid='"+editEmpid.getText()+"'"); showMessage("Success", "Record Modified");

}

else

{

showMessage("Error", "Invalid Rollno");

}

clearText();

}

if(view==btnView)

{

if(editEmpid.getText().toString().trim().length()==0)

{

showMessage("Error", "Please enter Employee id"); return;

Cursor c=db.rawQuery("SELECT \* FROM employee WHERE empid='"+editEmpid.getText()+"'", null);

if(c.moveToFirst())

{

editName.setText(c.getString(1)); editsalary.setText(c.getString(2));

}

else

{

showMessage("Error", "Invalid Employee id"); clearText();

}

}

if(view==btnViewAll)

{

Cursor c=db.rawQuery("SELECT \* FROM employee", null); if(c.getCount()==0)

{

showMessage("Error", "No records found"); return;

}

StringBuffer buffer=new StringBuffer(); while(c.moveToNext())

{

buffer.append("Employee id: "+c.getString(0)+"\n"); buffer.append("Name: "+c.getString(1)+"\n"); buffer.append("salary: "+c.getString(2)+"\n\n");

}

showMessage("Employee details 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()

{

editEmpid.setText(""); editName.setText(""); editsalary.setText(""); editEmpid.requestFocus();

}

}

1. Now go to main.xml and right click .select run as option and select run configuration
2. Android output is present in the android emulator as shown in below.

**Week 7**

**Aim: Develop an application that makes use of RSS Feed.**

Create an activity **MainActivity.java** with the following code. Clicking on the button of this activity redirect to RSSFeedActivity.java class and load the content of a given URL.

**activity\_main.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"**>**

**<Button**

        android:id="@+id/btnRediff"

        android:layout\_width="300dp"

        android:layout\_height="wrap\_content"

        android:layout\_alignParentBottom="true"

        android:layout\_centerHorizontal="true"

        android:layout\_marginBottom="101dp"

        android:text="REDIFF RSS FEED" **/>**

**<Button**

        android:id="@+id/btnCinemaBlend"

        android:layout\_width="300dp"

        android:layout\_height="wrap\_content"

        android:layout\_alignLeft="@+id/btnRediff"

        android:layout\_alignParentBottom="true"

        android:layout\_alignStart="@+id/btnRediff"

        android:layout\_marginBottom="28dp"

    android:text="CINEMA BLEND RSS FEED" **/>**

**</RelativeLayout>**

**MainActivity.java**

**package** example.rssfeed.com.androidrssfeed;

**import** android.support.v7.app.AppCompatActivity;

**import** android.os.Bundle;

**import** android.content.Intent;

**import** android.view.View;

**import** android.widget.Button;

**import** java.util.ArrayList;

**public** **class** MainActivity **extends** AppCompatActivity **implements** View.OnClickListener {

      ArrayList<String> rssLinks = **new** ArrayList<>();

    @Override

**protected** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_main);

          Button btnRediff = findViewById(R.id.btnRediff);

        Button btnCinemaBlend = findViewById(R.id.btnCinemaBlend);

        btnRediff.setOnClickListener(**this**);

        btnCinemaBlend.setOnClickListener(**this**);

        rssLinks.add("http://www.rediff.com/rss/moviesreviewsrss.xml");

        rssLinks.add("http://www.cinemablend.com/rss\_review.php");

    }

      @Override

**public** **void** onClick(View view) {

        Intent intent = **new** Intent(getApplicationContext(), RSSFeedActivity.**class**);

**switch** (view.getId()) {

**case** R.id.btnRediff:

                intent.putExtra("rssLink", rssLinks.get(0));

                startActivity(intent);

**break**;

**case** R.id.btnCinemaBlend:

                intent.putExtra("rssLink", rssLinks.get(1));

                startActivity(intent);

**break**;

        }

    }

}

Create a layout **rss\_item\_list\_row.xml** which contains the fields of a newsletter (page URL, title, publishing date) which are displayed in RSS Feed.

**rss\_item\_list\_row.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:padding="8dip"**>**

**<TextView**

        android:id="@+id/page\_url"

        android:layout\_width="fill\_parent"

        android:layout\_height="wrap\_content"

        android:visibility="gone" **/>**

**<TextView**

        android:id="@+id/title"

        android:layout\_width="fill\_parent"

        android:layout\_height="wrap\_content"

        android:paddingBottom="1dip"

        android:textColor="#212121"

        android:textSize="18sp"

        android:textStyle="bold" **/>**

**<TextView**

        android:id="@+id/pub\_date"

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:layout\_below="@id/title"

        android:paddingBottom="3dip"

        android:textColor="#9b737775"

        android:textSize="14sp" **/>**

**</RelativeLayout>**

**RSSItem.java**

**package** example.rssfeed.com.androidrssfeed;

**public** **class** RSSItem {

**public** String title;

**public** String link;

**public** String description;

**public** String pubdate;

**public** String guid;

**public** RSSItem(String title, String link, String description, String pubdate, String guid) {

**this**.title = title;

**this**.link = link;

**this**.description = description;

**this**.pubdate = pubdate;

**this**.guid = guid;

    }

}

Create **RSSParser.java** class, in this class; we will use the instance of DocumentBuilderFactory class to parse the XML document.

**RSSParser.java**

**package** example.rssfeed.com.androidrssfeed;

**import** android.util.Log;

**import** org.apache.http.HttpEntity;

**import** org.apache.http.HttpResponse;

**import** org.apache.http.client.ClientProtocolException;

**import** org.apache.http.client.methods.HttpGet;

**import** org.apache.http.impl.client.DefaultHttpClient;

**import** org.apache.http.util.EntityUtils;

**import** org.w3c.dom.Document;

**import** org.w3c.dom.Element;

**import** org.w3c.dom.Node;

**import** org.w3c.dom.NodeList;

**import** org.xml.sax.InputSource;

**import** org.xml.sax.SAXException;

**import** java.io.IOException;

**import** java.io.StringReader;

**import** java.io.UnsupportedEncodingException;

**import** java.util.ArrayList;

**import** java.util.List;

**import** javax.xml.parsers.DocumentBuilder;

**import** javax.xml.parsers.DocumentBuilderFactory;

**import** javax.xml.parsers.ParserConfigurationException;

**public** **class** RSSParser {

   // RSS XML document CHANNEL tag

**private** **static** String TAG\_CHANNEL = "channel";

**private** **static** String TAG\_TITLE = "title";

**private** **static** String TAG\_LINK = "link";

**private** **static** String TAG\_DESRIPTION = "description";

**private** **static** String TAG\_ITEM = "item";

**private** **static** String TAG\_PUB\_DATE = "pubDate";

**private** **static** String TAG\_GUID = "guid";

**public** RSSParser() {

    }

**public** List<RSSItem> getRSSFeedItems(String rss\_url) {

        List<RSSItem> itemsList = **new** ArrayList<RSSItem>();

        String rss\_feed\_xml;

          rss\_feed\_xml = **this**.getXmlFromUrl(rss\_url);

**if** (rss\_feed\_xml != **null**) {

**try** {

                Document doc = **this**.getDomElement(rss\_feed\_xml);

                NodeList nodeList = doc.getElementsByTagName(TAG\_CHANNEL);

                Element e = (Element) nodeList.item(0);

                  NodeList items = e.getElementsByTagName(TAG\_ITEM);

**for** (**int** i = 0; i < items.getLength(); i++) {

                    Element e1 = (Element) items.item(i);

                    String title = **this**.getValue(e1, TAG\_TITLE);

                    String link = **this**.getValue(e1, TAG\_LINK);

                    String description = **this**.getValue(e1, TAG\_DESRIPTION);

                    String pubdate = **this**.getValue(e1, TAG\_PUB\_DATE);

                    String guid = **this**.getValue(e1, TAG\_GUID);

                      RSSItem rssItem = **new** RSSItem(title, link, description, pubdate, guid);

                    // adding item to list

                    itemsList.add(rssItem);

                }

            } **catch** (Exception e) {

                // Check log for errors

                e.printStackTrace();

            }

        }

**return** itemsList;

    }

**public** String getXmlFromUrl(String url) {

        String xml = **null**;

**try** {

            DefaultHttpClient httpClient = **new** DefaultHttpClient();

            HttpGet httpGet = **new** HttpGet(url);

            HttpResponse httpResponse = httpClient.execute(httpGet);

            HttpEntity httpEntity = httpResponse.getEntity();

            xml = EntityUtils.toString(httpEntity);

        } **catch** (UnsupportedEncodingException e) {

            e.printStackTrace();

        } **catch** (ClientProtocolException e) {

            e.printStackTrace();

        } **catch** (IOException e) {

            e.printStackTrace();

        }

**return** xml;

    }

**public** Document getDomElement(String xml) {

        Document doc = **null**;

        DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();

**try** {

              DocumentBuilder db = dbf.newDocumentBuilder();

            InputSource is = **new** InputSource();

            is.setCharacterStream(**new** StringReader(xml));

            doc = db.parse(is);

        } **catch** (ParserConfigurationException e) {

            Log.e("Error: ", e.getMessage());

**return** **null**;

        } **catch** (SAXException e) {

            Log.e("Error: ", e.getMessage());

**return** **null**;

        } **catch** (IOException e) {

            Log.e("Error: ", e.getMessage());

**return** **null**;

        }

**return** doc;

    }

**public** **final** String getElementValue(Node elem) {

        Node child;

**if** (elem != **null**) {

**if** (elem.hasChildNodes()) {

**for** (child = elem.getFirstChild(); child != **null**; child = child

                        .getNextSibling()) {

**if** (child.getNodeType() == Node.TEXT\_NODE || (child.getNodeType() == Node.CDATA\_SECTION\_NODE)) {

**return** child.getNodeValue();

                    }

                }

            }

        }

**return** "";

    }

**public** String getValue(Element item, String str) {

        NodeList n = item.getElementsByTagName(str);

**return** **this**.getElementValue(n.item(0));

    }

}

Now, create an activity **RSSFeedActivity.java** with the following code. This class uses AsyncTask class to load the items of RSS Feed from the URL in the background.

### activity\_rssfeed.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:id="@+id/relativeLayout"

    android:orientation="vertical"**>**

**<ListView**

        android:id="@android:id/list"

        android:layout\_width="match\_parent"

        android:layout\_height="wrap\_content"

        android:dividerHeight="1dp" **/>**

**</RelativeLayout>**

### RSSFeedActivity.java

**package** example.rssfeed.com.androidrssfeed;

**import** android.app.ListActivity;

**import** android.content.Intent;

**import** android.os.AsyncTask;

**import** android.os.Bundle;

**import** android.view.View;

**import** android.widget.AdapterView;

**import** android.widget.ListAdapter;

**import** android.widget.ListView;

**import** android.widget.ProgressBar;

**import** android.widget.RelativeLayout;

**import** android.widget.SimpleAdapter;

**import** android.widget.TextView;

**import** android.widget.Toast;

**import** java.text.ParseException;

**import** java.text.SimpleDateFormat;

**import** java.util.ArrayList;

**import** java.util.Date;

**import** java.util.HashMap;

**import** java.util.List;

**import** java.util.Locale;

**public** **class** RSSFeedActivity **extends** ListActivity {

**private** ProgressBar pDialog;

    ArrayList<HashMap<String, String>> rssItemList = **new** ArrayList<>();

    RSSParser rssParser = **new** RSSParser();

    List<RSSItem> rssItems = **new** ArrayList<>();

**private** **static** String TAG\_TITLE = "title";

**private** **static** String TAG\_LINK = "link";

**private** **static** String TAG\_PUB\_DATE = "pubDate";

      @Override

**public** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_rssfeed);

        String rss\_link = getIntent().getStringExtra("rssLink");

**new** LoadRSSFeedItems().execute(rss\_link);

        ListView lv = getListView();

       lv.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {

**public** **void** onItemClick(AdapterView<?> parent, View view,

**int** position, **long** id) {

                Intent in = **new** Intent(getApplicationContext(), WebActivity.**class**);

                String page\_url = ((TextView) view.findViewById(R.id.page\_url)).getText().toString().trim();

                in.putExtra("url", page\_url);

                startActivity(in);

            }

        });

    }

**public** **class** LoadRSSFeedItems **extends** AsyncTask<String, String, String> {

          @Override

**protected** **void** onPreExecute() {

**super**.onPreExecute();

            pDialog = **new** ProgressBar(RSSFeedActivity.**this**, **null**, android.R.attr.progressBarStyleLarge);

            RelativeLayout relativeLayout = findViewById(R.id.relativeLayout);

            RelativeLayout.LayoutParams layoutParams = **new** RelativeLayout.LayoutParams(

                    RelativeLayout.LayoutParams.WRAP\_CONTENT,

                    RelativeLayout.LayoutParams.WRAP\_CONTENT

            );

             layoutParams.addRule(RelativeLayout.CENTER\_IN\_PARENT);

            pDialog.setLayoutParams(layoutParams);

            pDialog.setVisibility(View.VISIBLE);

            relativeLayout.addView(pDialog);

        }

        @Override

**protected** String doInBackground(String... args) {

            // rss link url

            String rss\_url = args[0];

            // list of rss items

            rssItems = rssParser.getRSSFeedItems(rss\_url);

            // looping through each item

**for** (**final** RSSItem item : rssItems) {

                // creating new HashMap

**if** (item.link.toString().equals(""))

**break**;

                HashMap<String, String> map = **new** HashMap<String, String>();

                // adding each child node to HashMap key => value

                String givenDateString = item.pubdate.trim();

                SimpleDateFormat sdf = **new** SimpleDateFormat("EEE, d MMM yyyy HH:mm:ss Z");

**try** {

                    Date mDate = sdf.parse(givenDateString);

                    SimpleDateFormat sdf2 = **new** SimpleDateFormat("EEEE, dd MMMM yyyy - hh:mm a", Locale.US);

                    item.pubdate = sdf2.format(mDate);

                  } **catch** (ParseException e) {

                    e.printStackTrace();

                  }

                  map.put(TAG\_TITLE, item.title);

                map.put(TAG\_LINK, item.link);

                map.put(TAG\_PUB\_DATE, item.pubdate);

                // adding HashList to ArrayList

                rssItemList.add(map);

            }

              // updating UI from Background Thread

            runOnUiThread(**new** Runnable() {

**public** **void** run() {

                    ListAdapter adapter = **new** SimpleAdapter(

                            RSSFeedActivity.**this**,

                            rssItemList, R.layout.rss\_item\_list\_row,

**new** String[]{TAG\_LINK, TAG\_TITLE, TAG\_PUB\_DATE},

**new** **int**[]{R.id.page\_url, R.id.title, R.id.pub\_date});

                    // updating listview

                    setListAdapter(adapter);

                }

            });

**return** **null**;

        }

**protected** **void** onPostExecute(String args) {

            pDialog.setVisibility(View.GONE);

        }

    }

}

Create an activity **WebActivity.java** containing WebView, that load the content of the link clicked in the previous activity.

### activity\_web.xml

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

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

    android:layout\_width="match\_parent"

    android:id="@+id/relativeLayout"

    android:layout\_height="match\_parent"**>**

**<android.support.v4.widget.NestedScrollView**

        android:layout\_width="match\_parent"

        android:layout\_height="match\_parent"**>**

**<WebView**

            android:id="@+id/webView"

            android:layout\_width="match\_parent"

            android:layout\_height="match\_parent" **/>**

**</android.support.v4.widget.NestedScrollView>**

**</RelativeLayout>**

### WebActivity.java

**package** example.rssfeed.com.androidrssfeed;

**import** android.content.Context;

**import** android.content.Intent;

**import** android.graphics.Bitmap;

**import** android.os.Bundle;

**import** android.support.v7.app.AppCompatActivity;

**import** android.text.TextUtils;

**import** android.webkit.WebChromeClient;

**import** android.webkit.WebResourceError;

**import** android.webkit.WebResourceRequest;

**import** android.webkit.WebView;

**import** android.webkit.WebViewClient;

**import** android.widget.Toast;

**public** **class** WebActivity **extends** AppCompatActivity {

      WebView webView;

    String url;

      @Override

**public** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_web);

        Intent in = getIntent();

        url = in.getStringExtra("url");

**if** (TextUtils.isEmpty(url)) {

            Toast.makeText(getApplicationContext(), "URL not found", Toast.LENGTH\_SHORT).show();

            finish();

        }

        webView = findViewById(R.id.webView);

        initWebView();

        webView.loadUrl(url);

    }

**private** **void** initWebView() {

        webView.setWebChromeClient(**new** MyWebChromeClient(**this**));

        webView.clearCache(**true**);

        webView.getSettings().setJavaScriptEnabled(**true**);

        webView.setHorizontalScrollBarEnabled(**false**);

        webView.setWebViewClient(**new** WebViewClient() {

            @Override

**public** **void** onPageStarted(WebView view, String url, Bitmap favicon) {

**super**.onPageStarted(view, url, favicon);

            }

            @Override

**public** **boolean** shouldOverrideUrlLoading(WebView view, String url) {

                webView.loadUrl(url);

**return** **true**;

            }

            @Override

**public** **void** onPageFinished(WebView view, String url) {

**super**.onPageFinished(view, url);

            }

            @Override

**public** **void** onReceivedError(WebView view, WebResourceRequest request, WebResourceError error) {

**super**.onReceivedError(view, request, error);

                invalidateOptionsMenu();

            }

        });

        webView.clearCache(**true**);

        webView.clearHistory();

        webView.getSettings().setJavaScriptEnabled(**true**);

        webView.setHorizontalScrollBarEnabled(**false**);

    }

**private** **class** MyWebChromeClient **extends** WebChromeClient {

        Context context;

**public** MyWebChromeClient(Context context) {

**super**();

**this**.context = context;

        }

    }

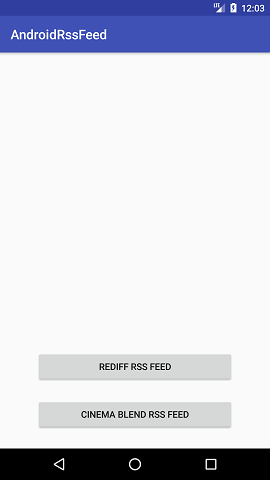
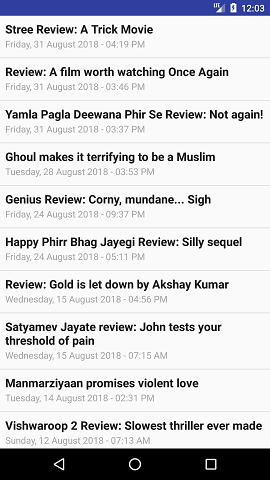
}

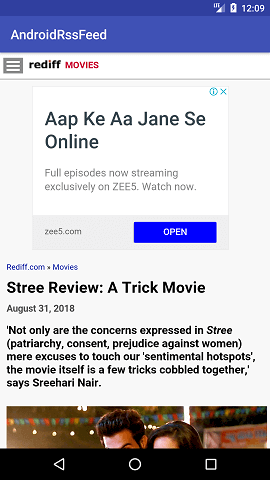
### Permission Required

Add the INTERNET permission in **AndroidMenifest.xml** file.

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

**Output**



**Week 8:**

**Aim: Create an application that implements Multi threading.**

**Procedure:**

1. Open eclipse or android studio and select new android project
2. Give project name and select next
3. Choose the android version.Choose the lowest android version(Android 2.2) and select next
4. Enter the package name.package name must be two word seprated by comma and click finish
5. Go to package explorer in the left hand side.select our project.
6. Go to res folder and select layout.Double click the main.xml file.Add the code below

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

<LinearLayout xmlns:android[="http](http://schemas.android.com/apk/res/android):/[/schemas.android.com/apk/res/android](http://schemas.android.com/apk/res/android)" android:layout\_width="match\_parent" android:layout\_height="match\_parent"

android:orientation="vertical" android:id="@+id/info" >

<Button android:id="@+id/button1"

android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:onClick="fetchData" android:text="Start MULTITHREAD" />

<TextView android:id="@+id/textView1" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"

android:text="Main thread" />

</LinearLayout>

1. Now select mainactivity.java file and type the following

code. package multi.threading;

//import your.first.R;

import android.app.Activity; import android.os.Bundle; import android.os.Handler; import android.view.View;

import android.widget.TextView;

public class MultiThreadingActivity extends Activity { private TextView tvOutput;

private static final int t1 = 1; private static final int t2 = 2; private static final int t3 = 3; @Override

public void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.main);

tvOutput = (TextView) findViewById(R.id.textView1);

}

public void fetchData(View v) { tvOutput.setText("Main thread"); thread1.start();

thread2.start(); thread3.start();

}

Thread thread1 = new Thread(new Runnable() { @Override

public void run() {

for (int i = 0; i < 5; i++) {

try { Thread.sleep(1000);

} catch (InterruptedException e) { e.printStackTrace();

}

handler.sendEmptyMessage(t1);

}

}

});

Thread thread2 = new Thread(new Runnable() { @Override

public void run() {

for (int i = 0; i < 5; i++) { try {

Thread.sleep(1000);

} catch (InterruptedException e) { e.printStackTrace();

}

handler.sendEmptyMessage(t2);

}

}

});

Thread thread3 = new Thread(new Runnable() { @Override

public void run() {

for (int i = 0; i < 5; i++) { try {

Thread.sleep(1000);

} catch (InterruptedException e) { e.printStackTrace();

}

handler.sendEmptyMessage(t3);

}

}

});

Handler handler = new Handler() {

public void handleMessage(android.os.Message msg) { if(msg.what == t1) {

tvOutput.append("\nIn thread 1");

}

if(msg.what == t2) { tvOutput.append("\nIn thread 2");

}

if(msg.what == t3) { tvOutput.append("\nIn thread 3");

}

}

};

}

1. Now go to main.xml and right click .select run as option and select run configuration
2. Android output is present in the android emulator as shown in below.

**Output:**



**Week 9**

**Aim: Develop a native application that uses GPS location information.**

**Procedure:**

1)Open eclipse or android studio and select new android project

2)Give project name and select next

1. Choose the android version.Choose the lowest android version(Android 2.2) and select next
2. Enter the package name.package name must be two word seprated by comma and click finish
3. Go to package explorer in the left hand side.select our project.
4. Go to res folder and select layout.Double click the main.xml file.Add the code below

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

<RelativeLayout xmlns:android[="http://sch](http://schemas.android.com/apk/res/android)em[as.android.com/apk/res/android](http://schemas.android.com/apk/res/android)" android:id="@+id/relativeLayout1"

android:layout\_width="match\_parent" android:layout\_height="match\_parent" >

<Button android:id="@+id/show\_Location" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content android:text="Show\_Location" android:layout\_centerVertical="true" android:layout\_centerHorizontal="true"

/>

</RelativeLayout>

1. Now select mainactivity.java file and type the following code.In my coding maniactivity name

is GPSlocationActivity.

package gps.location;

//import android.R;

import android.app.Activity; import android.os.Bundle; import android.view.View; import android.widget.Button; import android.widget.Toast;

public class GPSlocationActivity extends Activity {

/\*\* Called when the activity is first created. \*/ Button btnShowLocation;

GPStrace gps; @Override

public void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.main); btnShowLocation=(Button)findViewById(R.id.show\_Location); btnShowLocation.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

// TODO Auto-generated method stub gps=new GPStrace(GPSlocationActivity.this); if(gps.canGetLocation()){

double latitude=gps.getLatitude(); double longitude=gps.getLongtiude();

Toast.makeText(getApplicationContext(),"Your Location is

\nLat:"+latitude+"\nLong:"+longitude, Toast.LENGTH\_LONG).show();

}

else

{

gps.showSettingAlert();

}

}

});

}

}

1. Go to src folder and Right Click on your package folder and choose new class and give the

class nams as GPStrace

1. Select the GPStrace.java file and paste the following code.

package gps.location;

import android.app.AlertDialog; import android.app.Service; import android.content.Context;

import android.content.DialogInterface; import android.content.Intent;

import android.location.Location;

import android.location.LocationListener;

import android.location.LocationManager;

import android.os.Bundle;

import android.os.IBinder; import android.provider.Settings;

public class GPStrace extends Service implements LocationListener{ private final Context context;

boolean isGPSEnabled=false; boolean canGetLocation=false; boolean isNetworkEnabled=false; Location location;

double latitude; double longtitude;

private static final long MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES=10; private static final long MIN\_TIME\_BW\_UPDATES=1000\*60\*1;

protected LocationManager locationManager;

public GPStrace(Context context)

{

this.context=context; getLocation();

}

public Location getLocation()

{

try{ locationManager=(LocationManager)

context.getSystemService(LOCATION\_SERVICE); isGPSEnabled=locationManager.isProviderEnabled(LocationManager.GPS\_PROV IDER);

isNetworkEnabled=locationManager.isProviderEnabled(LocationManager.NETWO RK\_PROVI

DER);

if(!isGPSEnabled && !isNetworkEnabled){

}else{ this.canGetLocation=true; if(isNetworkEnabled){

locationManager.requestLocationUpdates( LocationManager.NETWORK\_PROVIDER, MIN\_TIME\_BW\_UPDATES, MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES,this);

}

if(locationManager!=null){ location=locationManager.getLastKnownLocation(LocationManager.NETWORK\_ PROVIDER)

;

if(location !=null){ latitude=location.getLatitude();

longtitude=location.getLongitude();

}

}

}

if(isGPSEnabled){ if(location==null){

locationManager.requestLocationUpdates(LocationManager.GPS\_PROVIDER,MI N\_TIME\_B

W\_UPDATES, MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES, this);

if(locationManager!=null){ location=locationManager.getLastKnownLocation(LocationManager.GPS\_PROVI DER);

if(location!=null){ latitude=location.getLatitude(); longtitude=location.getLongitude();

}

}

}

}

}

catch(Exception e)

{

e.printStackTrace();

}

return location;

}

public void stopUsingGPS(){ if(locationManager!=null){ locationManager.removeUpdates(GPStrace.this);

}

}

public double getLatitude(){

if(location!=null){ latitude=location.getLatitude();

}

return latitude;

}

public double getLongtiude(){ if(location!=null){ longtitude=location.getLatitude();

}

return longtitude;

}

public boolean canGetLocation(){ return this.canGetLocation;

}

public void showSettingAlert(){

AlertDialog.Builder alertDialog=new AlertDialog.Builder(context); alertDialog.setTitle("GPS is settings");

alertDialog.setMessage("GPS is not enabled.Do you want to go to setting menu?"); alertDialog.setPositiveButton("settings", new DialogInterface.OnClickListener() { @Override

public void onClick(DialogInterface dialog,int which){

Intent intent=new Intent(Settings.ACTION\_LOCATION\_SOURCE\_SETTINGS); context.startActivity(intent);

}

});

alertDialog.setNegativeButton("cancel", new DialogInterface.OnClickListener() { @Override

public void onClick(DialogInterface dialog, int which) {

// TODO Auto-generated method stub dialog.cancel();

}

});

alertDialog.show();

}

@Override

public void onLocationChanged(Location location) {

// TODO Auto-generated method stub

}

@Override

public void onProviderDisabled(String provider) {

// TODO Auto-generated method stub

}

@Override

public void onProviderEnabled(String provider) {

// TODO Auto-generated method stub

}

@Override

public void onStatusChanged(String provider, int status, Bundle extras) {

// TODO Auto-generated method stub

}

@Override

public IBinder onBind(Intent intent) {

// TODO Auto-generated method stub return null;

}

}

1. Go to manifest.xml file and add the code below

<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION"/>

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

1. Now go to main.xml and right click .select run as option and select run configuration
2. Android output is present in the android emulator as shown in below.

**Week 10**

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

# Procedure:

1. Open eclipse or android studio and select new android project
2. Give project name and select next
3. Choose the android version.Choose the lowest android version(Android 2.2) and select next
4. Enter the package name.package name must be two word seprated by comma and click finish
5. Go to package explorer in the left hand side.select our project.
6. Go to res folder and select layout.Double click the main.xml file.Add the code below

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

<LinearLayout xmlns:android[="http](http://schemas.android.com/apk/res/android):/[/schemas.android.com/apk/res/android](http://schemas.android.com/apk/res/android)" android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent" android:background="#ff0000ff" android:orientation="vertical" >

<EditText android:id="@+id/editText1" android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" >

<requestFocus />

</EditText>

<Button android:id="@+id/button1"

android:layout\_width="match\_parent" android:layout\_height="wrap\_content"

android:text="SAVE DATA" />

<Button android:id="@+id/button2"

android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:text="SHOW DATA" />

<TextView android:id="@+id/textView1" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"

/>

</LinearLayout>

1. Now select mainactivity.java file and type the following code. package save.sd;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException; import java.io.FileOutputStream; import java.io.IOException;

import java.io.InputStreamReader; import java.io.OutputStreamWriter; import android.app.Activity; import android.os.Bundle;

import android.os.Environment; import android.view.View; import android.widget.Button; import android.widget.EditText; import android.widget.TextView; import android.widget.Toast;

public class SavedatasdcardActivity extends Activity {

/\*\* Called when the activity is first created. \*/

Button save,load; EditText message;

TextView t1; String Message1; @Override

public void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.main);

save=(Button) findViewById(R.id.button1); load=(Button) findViewById(R.id.button2); message=(EditText) findViewById(R.id.editText1); t1=(TextView) findViewById(R.id.textView1); save.setOnClickListener(new View.OnClickListener(){ public void onClick(View v){

//Get message from user store in message1 variable Message1 =message.getText().toString();

try{

//Create a new folder called MyDirectory in SDCard

File sdcard=Environment.getExternalStorageDirectory();

File directory=new File(sdcard.getAbsolutePath()+"/MyDirectory"); directory.mkdirs();

//Create a new file name textfile.txt inside MyDirectory File file=new File(directory,"textfile.txt");

//Create File Outputstream to read the file FileOutputStream fou=new FileOutputStream(file); OutputStreamWriter osw=new OutputStreamWriter(fou); try{

//write a user data to file osw.append(Message1); osw.flush();

osw.close();

Toast.makeText(getBaseContext(),"Data Saved",Toast.LENGTH\_LONG).show();

}catch(IOException e){

e.printStackTrace();

}

}catch (FileNotFoundException e){ e.printStackTrace();

}

}

});

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

try{

File sdcard=Environment.getExternalStorageDirectory();

File directory=new File(sdcard.getAbsolutePath()+"/MyDirectory"); File file=new File(directory,"textfile.txt");

FileInputStream fis=new FileInputStream(file); InputStreamReader isr=new InputStreamReader(fis); char[] data=new char[100];

String final\_data=""; int size;

try{ while((size=isr.read(data))>0)

{

//read a data from file

String read\_data=String.copyValueOf(data,0,size); final\_data+=read\_data;

data=new char[100];

}

//display the data in output

Toast.makeText(getBaseContext(),"Message:"+final\_data,Toast.LENGTH\_LONG)

.show();

}catch(IOException e){ e.printStackTrace();

}

}catch (FileNotFoundException e){ e.printStackTrace();

}

}

});

}

}

1. Next step is to set permission to write data in sd card.So go to AndroidManifest.xml file. Copy

and paste the following coding.The code should come before <application> tab.

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"></uses- permission>

1. Now go to main.xml and right click .select run as option and select run configuration
2. Android output is present in the android emulator as shown in below.

**Week 11**

**Aim: Develop an application that creates notification upon receiving a message.**

**Procedure:**

1) Open eclipse or android studio and select new android project

2) Give project name and select next

1. Choose the android version.Choose the lowest android version(Android 2.2) and select next
2. Enter the package name.package name must be two word seprated by comma and click finish
3. Go to package explorer in the left hand side.select our project.
4. Go to res folder and select layout.Double click the main.xml file.Add the code below

<ScrollView xmlns:android[="http://s](http://schemas.android.com/apk/res/android)ch[emas.android.com/apk/res/android](http://schemas.android.com/apk/res/android)" android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content" android:scrollbars="vertical" >

<TableLayout android:layout\_width="match\_parent" android:layout\_height="wrap\_content"

android:shrinkColumns="\*" android:stretchColumns="\*" android:background="#000000">

<TableRow android:layout\_height="wrap\_content" android:layout\_width="match\_parent" android:gravity="center\_horizontal">

<TextView android:id="@+id/Title" android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content" android:layout\_margin="5px" android:focusable="false" android:focusableInTouchMode="false" android:gravity="center\_vertical|center\_horizontal" android:text="QUIZ"

android:textSize="25sp" android:textStyle="bold" />

<View android:layout\_height="2px" android:layout\_marginTop="5dip"

android:layout\_marginBottom="5dip" android:background="#DDFFDD"/>

</TableRow>

<TableRow android:layout\_height="wrap\_content" android:layout\_width="match\_parent" android:gravity="center\_horizontal">

<TextView android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:textSize="18sp" android:text="1.CAPTIAL OF INDIA" android:layout\_span="4" android:padding="18dip" android:textColor="#ffffff"/>

</TableRow>

<TableRow android:id="@+id/tableRow1" android:layout\_height="wrap\_content" android:layout\_width="match\_parent">

<RadioGroup

android:id="@+id/answer1" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:layout\_weight="0.4" >

<RadioButton android:id="@+id/answer1A" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:textColor="#ffffff" android:text="CHENNAI" />

<RadioButton android:id="@+id/answer1B" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:textColor="#ffffff" android:text="NEW DELHI" />

<RadioButton android:id="@+id/answer1C" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:textColor="#ffffff" android:text="MUMBAI" />

<RadioButton android:id="@+id/answer1D" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:textColor="#ffffff" android:text="HYDERBAD" />

</RadioGroup>

</TableRow>

<TableRow android:layout\_height="wrap\_content"

android:layout\_width="match\_parent" android:gravity="center\_horizontal">

<TextView

android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:textSize="18sp"

android:text="2. CAPTIAL OF RUSSIA?" android:layout\_span="4" android:padding="18dip"

android:textColor="#ffffff"/>

</TableRow>

<TableRow android:id="@+id/tableRow2" android:layout\_height="wrap\_content" android:layout\_width="match\_parent">

<RadioGroup android:id="@+id/answer2" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:layout\_weight="0.4" >

<RadioButton android:id="@+id/answer2A" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:textColor="#ffffff" android:text="WARSAW " />

<RadioButton android:id="@+id/answer2B" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:textColor="#ffffff" android:text="BERLIN" />

<RadioButton android:id="@+id/answer2C"

android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:textColor="#ffffff" android:text="MASCOW " />

<RadioButton android:id="@+id/answer2D" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:textColor="#ffffff" android:text="CANEBRA " />

</RadioGroup>

</TableRow>

<TableRow android:layout\_height="wrap\_content" android:layout\_width="match\_parent" android:gravity="center\_horizontal">

<Button android:id="@+id/submit"

android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:gravity="center" android:text="Submit" />

</TableRow>

</TableLayout>

</ScrollView>

1. Now select mainactivity.java file and type the following code.In my coding maniactivity name is

Alert1Activity.

1. Now go to main.xml and right click. select run as option and select run configuration
2. Android output is present in the android emulator as shown in below.

**69Week 12**

**Aim: Create an alarm clock mobile application.**

# AndroidManift

# AndroidManifest.xml

We need to give uses-permission for WAKE\_LOCK, other than that the AndroidManifest.xml is

pretty standard one. Just need to include the service and receiver.

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

<manifest xmlns:android[="http://sche](http://schemas.android.com/apk/res/android)m[as.an](http://schemas.android.com/apk/res/android)d[roid.com/apk/res/android](http://schemas.android.com/apk/res/android)" package="com.javapapers.androidalarmclock">

<uses-permission android:name="android.permission.WAKE\_LOCK" />

<application android:allowBackup="true" android:icon="@drawable/ic\_launcher" android:label="@string/app\_name" android:theme="@style/AppTheme">

<activity android:name=".AlarmActivity" android:label="@string/app\_name">

<intent-filter>

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

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

</intent-filter>

</activity>

<service android:name=".AlarmService" android:enabled="true" />

<receiver android:name=".AlarmReceiver" />

</application> </manifest>

# Android Activity activity\_my.xml

The Android Activity is designed to be simple. We have a TimePicker component followed by a

ToggleButton. That’s it. Choose the time to set the alarm and toggle the switch to on. The alarm

will work.

<RelativeLayout xmlns:android[="http://sch](http://schemas.android.com/apk/res/android)em[as.android.com/apk/res/android](http://schemas.android.com/apk/res/android)" xmlns:too[ls="h](http://schemas.android.com/tools)ttp[://sch](http://schemas.android.com/tools)e[mas.android.com/tools"](http://schemas.android.com/tools) android:layout\_width="match\_parent"

android:layout\_height="match\_parent" android:paddingLeft="@dimen/activity\_horizontal\_margin" android:paddingRight="@dimen/activity\_horizontal\_margin" android:paddingTop="@dimen/activity\_vertical\_margin" android:paddingBottom="@dimen/activity\_vertical\_margin" tools:context=".MyActivity">

<TimePicker android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:id="@+id/alarmTimePicker" android:layout\_alignParentTop="true" android:layout\_centerHorizontal="true" />

<ToggleButton android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="Alarm On/Off" android:id="@+id/alarmToggle" android:layout\_centerHorizontal="true" android:layout\_below="@+id/alarmTimePicker" android:onClick="onToggleClicked" />

<TextView android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"

android:textAppearance="?android:attr/textAppearanceLarge" android:text=""

android:id="@+id/alarmText" android:layout\_alignParentBottom="true" android:layout\_centerHorizontal="true" android:layout\_marginTop="20dp" android:layout\_below="@+id/alarmToggle" />

</RelativeLayout>

# AlarmActivity.java

AlarmActivity uses the AlarmManager to set the alarm and send notification on alarm trigger.

package com.javapapers.androidalarmclock; import android.app.Activity;

import android.app.AlarmManager; import android.app.PendingIntent; import android.content.Intent; import android.os.Bundle;

import android.util.Log; import android.view.View;

import android.widget.TextView; import android.widget.TimePicker; import android.widget.ToggleButton; import java.util.Calendar;

public class AlarmActivity extends Activity { AlarmManager alarmManager;

private PendingIntent pendingIntent; private TimePicker alarmTimePicker; private static AlarmActivity inst;

private TextView alarmTextView; public static AlarmActivity instance() { return inst;

}

@Override

public void onStart() { super.onStart();

inst = this;

}

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_my);

alarmTimePicker = (TimePicker) findViewById(R.id.alarmTimePicker); alarmTextView = (TextView) findViewById(R.id.alarmText); ToggleButton alarmToggle = (ToggleButton) findViewById(R.id.alarmToggle);

alarmManager = (AlarmManager) getSystemService(ALARM\_SERVICE);

}

public void onToggleClicked(View view) { if (((ToggleButton) view).isChecked()) { Log.d("MyActivity", "Alarm On"); Calendar calendar = Calendar.getInstance(); calendar.set(Calendar.HOUR\_OF\_DAY, alarmTimePicker.getCurrentHour()); calendar.set(Calendar.MINUTE, alarmTimePicker.getCurrentMinute());

Intent myIntent = new Intent(AlarmActivity.this, AlarmReceiver.class);

pendingIntent = PendingIntent.getBroadcast(AlarmActivity.this, 0, myIntent, 0);

alarmManager.set(AlarmManager.RTC, calendar.getTimeInMillis(),

pendingIntent);

} else { alarmManager.cancel(pendingIntent); setAlarmText(""); Log.d("MyActivity", "Alarm Off");

}

}

public void setAlarmText(String alarmText) { alarmTextView.setText(alarmText);

}

}

# Alarm Receiver AlarmReceiver.java

AlarmReceiver is a WakefulBroadcasReceiver, this is the one that receives the alarm trigger on

set time. From here we initiate different actions to notify the user as per our choice.

I have given

three type of notifications, first show a message to user in the activity UI, second play the alarm

ringtone and third send an Android notification message. So this is the place to add enhancement

for different types of user notifications. package com.javapapers.androidalarmclock; import android.app.Activity;

import android.content.ComponentName; import android.content.Context;

import android.content.Intent; import android.media.Ringtone;

import android.media.RingtoneManager; import android.net.Uri;

import android.support.v4.content.WakefulBroadcastReceiver;

public class AlarmReceiver extends WakefulBroadcastReceiver { @Override

public void onReceive(final Context context, Intent intent) {

//this will update the UI with message AlarmActivity inst = AlarmActivity.instance(); inst.setAlarmText("Alarm! Wake up! Wake up!");

//this will sound the alarm tone

//this will sound the alarm once, if you wish to

//raise alarm in loop continuously then use MediaPlayer and setLooping(true)

Uri alarmUri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE\_ALARM); if (alarmUri == null) {

alarmUri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE\_NOTIFICATION);

}

Ringtone ringtone = RingtoneManager.getRingtone(context, alarmUri); ringtone.play();

//this will send a notification message

ComponentName comp = new ComponentName(context.getPackageName(), AlarmService.class.getName());

startWakefulService(context, (intent.setComponent(comp))); setResultCode(Activity.RESULT\_OK);

}

}

# Alarm Notification Message AlarmService.java

The receiver will start the following IntentService to send a standard notification to the user.

package com.javapapers.androidalarmclock; import android.app.IntentService;

import android.app.NotificationManager; import android.app.PendingIntent; import android.content.Context;

import android.content.Intent;

import android.support.v4.app.NotificationCompat; import android.util.Log;

public class AlarmService extends IntentService { private NotificationManager alarmNotificationManager; public AlarmService() {

super("AlarmService");

}

@Override

public void onHandleIntent(Intent intent) { sendNotification("Wake Up! Wake Up!");

}

private void sendNotification(String msg) { Log.d("AlarmService", "Preparing to send notification...: " + msg); alarmNotificationManager = (NotificationManager) this

.getSystemService(Context.NOTIFICATION\_SERVICE); PendingIntent contentIntent = PendingIntent.getActivity(this, 0, new Intent(this, AlarmActivity.class), 0); NotificationCompat.Builder alamNotificationBuilder = new NotificationCompat.Builder(

this).setContentTitle("Alarm").setSmallIcon(R.drawable.ic\_launcher)

.setStyle(new NotificationCompat.BigTextStyle().bigText(msg))

.setContentText(msg); alamNotificationBuilder.setContentIntent(contentIntent); alarmNotificationManager.notify(1, alamNotificationBuilder.build()); Log.d("AlarmService", "Notification sent.");

}

}