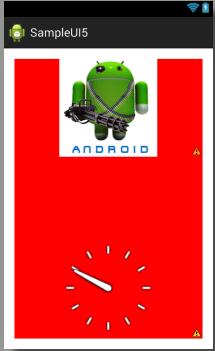
**Practical 4 (Layout)**

**Question 1 (FrameLayout)**

1. In the /layout directory, open the **activity\_main.xml** and design the interface as below using graphical layout. You should apply the **FrameLayout** in the design. You may use any image as you wish.



1. Your xml file should have something similar to xml below: **(p/s: You do not need to follow the xml as it might be different in each design.)**

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

xmlns:tools=*"http://schemas.android.com/tools"*

android:layout\_width=*"match\_parent"*

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

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.sampleui5.MainActivity"* >

<FrameLayout

android:layout\_width=*"match\_parent"*

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

<ImageView

android:id=*"@+id/imageView1"*

android:layout\_width=*"match\_parent"*

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

android:src=*"@drawable/RedBackGround"* />

<ImageView

android:id=*"@+id/imageView2"*

android:layout\_width=*"match\_parent"*

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

android:layout\_gravity=*"top"*

android:src=*"@drawable/android1"* />

<AnalogClock

android:id=*"@+id/analogClock1"*

android:layout\_width=*"match\_parent"*

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

android:layout\_gravity=*"bottom"* />

</FrameLayout>

</LinearLayout>

1. After make sure all the coding appropriate done, you may compile and run your coding .

**Question 2 (Grid Layout)**

1. Start a new project named myGrid.
2. Create 8 images and store in the res/drawable/ folder. (preferably png format)
3. Open the res/layout/main.xml file and insert the following:

<?xml version="1.0" encoding="utf-8"?>  
<GridView xmlns:android="http://schemas.android.com/apk/res/android"  
    android:id="@+id/gridview"  
    android:layout\_width="match\_parent"  
    android:layout\_height="match\_parent"  
    android:columnWidth="90dp"  
    android:numColumns="auto\_fit"  
    android:verticalSpacing="10dp"  
    android:horizontalSpacing="10dp"  
    android:stretchMode="columnWidth"  
    android:gravity="center"  
/>

1. In the MainActivity:

public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main);  
  
    GridView gridview = (GridView) findViewById(R.id.gridview);  
    gridview.setAdapter(new ImageAdapter(this));  
  
    gridview.setOnItemClickListener(new OnItemClickListener() {  
        public void onItemClick(AdapterView<?> parent, View v,  
                int position, long id) {  
            Toast.makeText(HelloGridView.this, "" + position,  
                    Toast.LENGTH\_SHORT).show();  
        }  
    });  
}

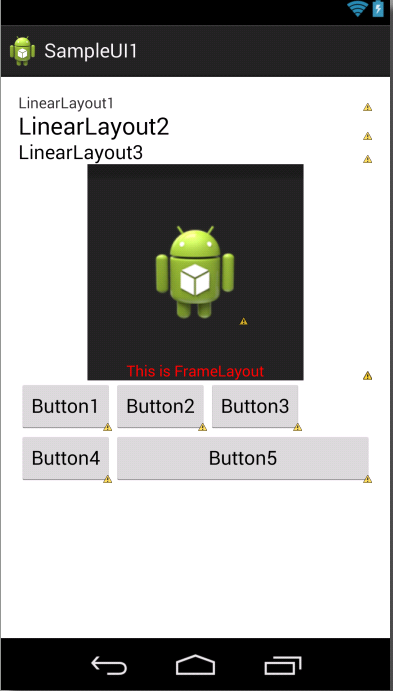
1. Create a new class called ImageAdapter that extends [BaseAdapter](http://developer.android.com/reference/android/widget/BaseAdapter.html):

public class ImageAdapter extends BaseAdapter {  
    private Context mContext;  
  
    public ImageAdapter(Context c) {  
        mContext = c;  
    }  
  
    public int getCount() {  
        return mThumbIds.length;  
    }  
  
    public Object getItem(int position) {  
        return null;  
    }  
  
    public long getItemId(int position) {  
        return 0;  
    }  
  
    // create a new ImageView for each item referenced by the Adapter  
    public View getView(int position, View convertView, ViewGroup parent) {  
        ImageView imageView;  
        if (convertView == null) {  
            // if it's not recycled, initialize some attributes  
            imageView = new ImageView(mContext);  
            imageView.setLayoutParams(new GridView.LayoutParams(85, 85));  
            imageView.setScaleType(ImageView.ScaleType.CENTER\_CROP);  
            imageView.setPadding(8, 8, 8, 8);  
        } else {  
            imageView = (ImageView) convertView;  
        }  
  
        imageView.setImageResource(mThumbIds[position]);  
        return imageView;  
    }  
  
    // references to our images  
    private Integer[] mThumbIds = {  
            R.drawable.sample\_2, R.drawable.sample\_3,  
            R.drawable.sample\_4, R.drawable.sample\_5,  
            R.drawable.sample\_6, R.drawable.sample\_7,  
            R.drawable.sample\_0, R.drawable.sample\_1,  
            R.drawable.sample\_2, R.drawable.sample\_3,  
            R.drawable.sample\_4, R.drawable.sample\_5,  
            R.drawable.sample\_6, R.drawable.sample\_7,  
            R.drawable.sample\_0, R.drawable.sample\_1,  
            R.drawable.sample\_2, R.drawable.sample\_3,  
            R.drawable.sample\_4, R.drawable.sample\_5,  
            R.drawable.sample\_6, R.drawable.sample\_7  
    };  
}

1. Run the application.

**Question 3 (Multiple Layout)**

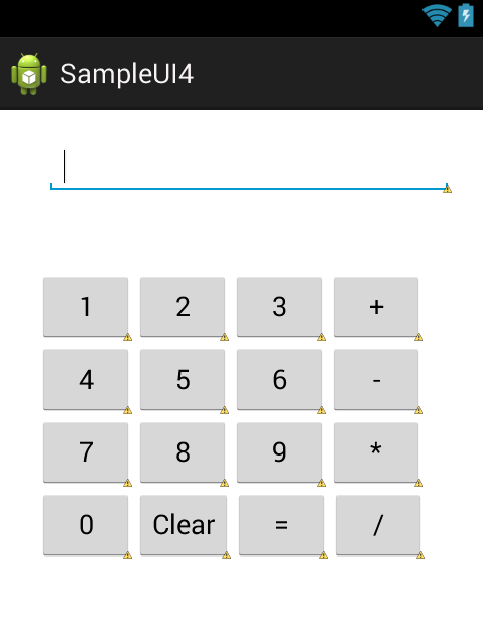
1. In this question, you are required to create an app with the following interface. In this question, you have to apply multiple layouts which include **LinearLayout, FrameLayout, and also TableLayout.**
2. In the /layout directory, open the **activity\_main.xml** and design the interface as below using graphical layout. You should apply the **appropriate layout** in the design. You may use any image as you wish.



1. After make sure all the coding appropriate done, you may compile and run your coding .

**Question 4**

1. In this question, you are required to create a calculator app by using **GridLayout**. This calculator app will perform primary mathematical operations like:
2. Addition of two numbers
3. Subtraction of two numbers
4. Multiplication of two numbers
5. Division of two numbers
6. In the /layout directory, open the activity\_main.xml and design the interface as above using graphical layout or edit the xml file. You may have different design of calculator.
7. By providing appropriate function to the button, make your calculator working. You may refer to Practical 3 for providing function to button.
8. After make sure all the coding appropriate done, you may compile and run your coding .



**Question 5 (ListView)**

1. This question is to show the working of ListView by using ArrayAdapter. In order for the ListView to work, we need to provide a template TextView to display the data from array. For this, create a **NEW** TextView in **simplerow.xml** with the following details.

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

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

android:id=*"@+id/rowTextView"*

android:layout\_width=*"fill\_parent"*

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

android:padding=*"10dp"*

android:textSize=*"16sp"* >

</TextView>

1. In your MainActivity.java, (**OnCreate() function**)provides the codes below:

**package** com.example.sampleui4;

**import** java.util.ArrayList;

**import** java.util.Arrays;

**import** android.app.Activity;

**import** android.os.Bundle;

**import** android.view.Menu;

**import** android.view.MenuItem;

**import** android.view.View;

**import** android.widget.AdapterView;

**import** android.widget.ArrayAdapter;

**import** android.widget.ListView;

**public** **class** MainActivity **extends** Activity {

**private** ListView mainListView ;

**private** ArrayAdapter<String> listAdapter ;

@Override

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

**super**.onCreate(savedInstanceState);

setContentView(R.layout.***activity\_main***);

mainListView = (ListView) findViewById( R.id.***mainListView*** );

// Create and populate a List of days.

String[] days = **new** String[] { "Sunday", "Monday", "Tuesday", "Wednesday",

"Thursday", "Friday", "Saturday"};

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

daylist.addAll( Arrays.*asList*(days) );

// Create ArrayAdapter using the day list.

listAdapter = **new** ArrayAdapter<String>(**this**, R.layout.***simplerow***, daylist);

// Add more Days. If you passed a String[] instead of a List<String>

// into the ArrayAdapter constructor, you must not add more items.

// Otherwise an exception will occur.

listAdapter.add( "BusyDay" );

listAdapter.add( "RainingDay" );

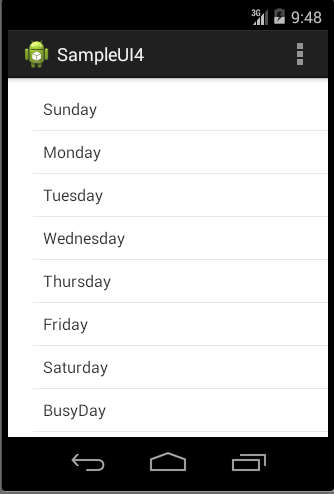
listAdapter.add( "SunnyDay" );

// Set the ArrayAdapter as the ListView's adapter.

mainListView.setAdapter( listAdapter );

}

1. After make sure all the coding appropriate done, you may compile and run your coding . You should get the following output:



1. Based on the codes above, integrate the following function so that a **Toast message** is generated when user click on each of the list element.

ListView listView = getListView();

listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {

public void onItemClick(AdapterView<?> av, View view, int i, long l) {

Toast.makeText(Activity.this, "myPos "+i, Toast.LENGTH\_LONG).show();

}

});