**Activity 4 (understanding the lifecycle of the Activity)**

**1**)using eclipse ,create a new android project and name it according to the figure

****

****

2)in the ManinActivity.java file ,add the following statement

**package** pk.edu.iba;

**import** android.app.Activity;

**import** android.os.Bundle;

**import** android.util.Log;

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

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

String tag="Events";

@Override

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

**super**.onCreate(savedInstanceState);

setContentView(R.layout.*main*);

Log.*d*(tag,"in the onCreate() event");

}

**public** **void** onStart()

{

**super**.onStart();

Log.*d*(tag,"in the on start event");

}

**public** **void** onRestart()

{

**super**.onStart();

Log.*d*(tag,"in the on Re start event");

}

**public** **void** onResum()

{

**super**.onResume();

Log.*d*(tag,"in the on onResume event");

}

**public** **void** onPause()

{

**super**.onPause();

Log.*d*(tag,"in the on onPause event");

}

**public** **void** onStop()

{

**super**.onStop();

Log.*d*(tag,"in the on onstop event");

}

**public** **void** onDestroy()

{

**super**.onDestroy();

Log.*d*(tag,"in the on onDestroy event");

}

}

3) Press F11 to debug the application on the Android Emulator.

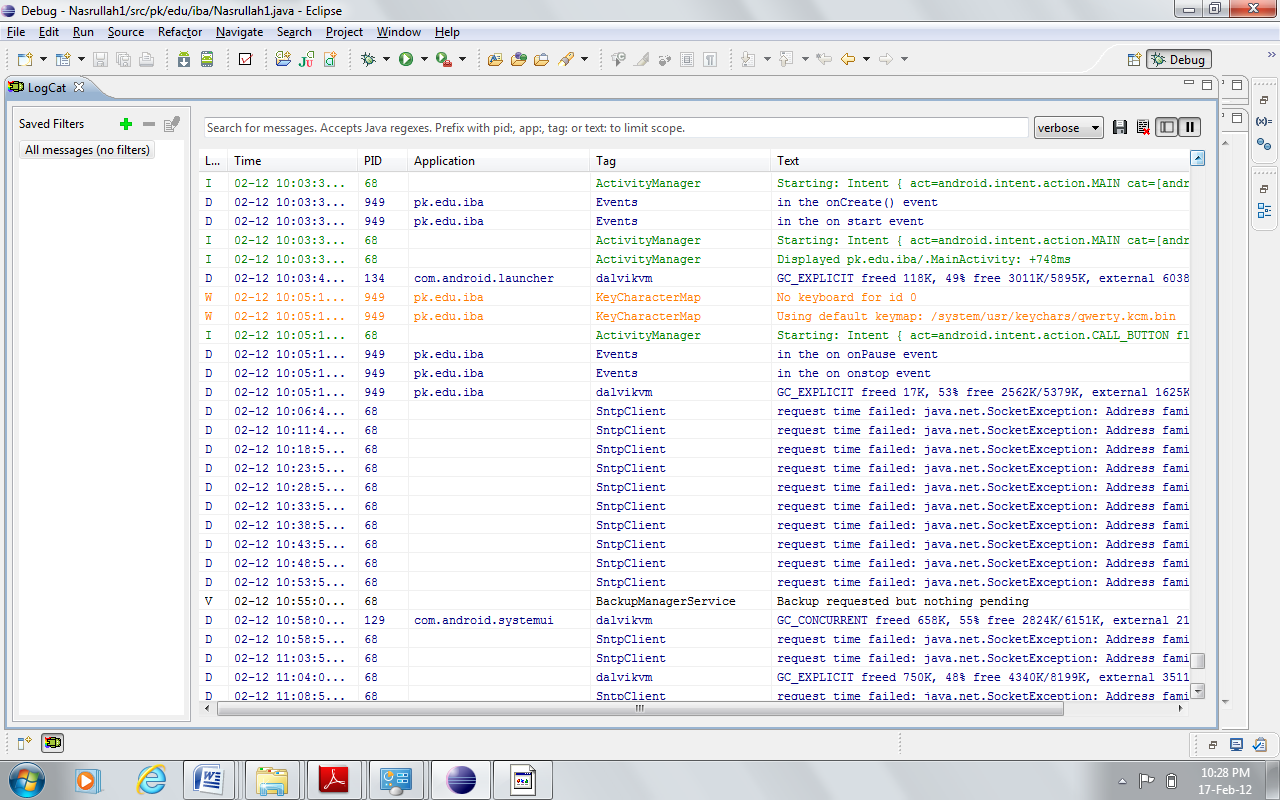
**4.)** When the activity is first loaded, you should see the following in the LogCat window (click on the

Debug perspective;

12-28 13:45:28.115: DEBUG/Events(334): In the onCreate() event

12-28 13:45:28.115: DEBUG/Events(334): In the onStart() event

12-28 13:45:28.115: DEBUG/Events(334): In the onResume() event



5)press the back button on the emulator and write down the results

6)click the home button,click on the activities icon and write down the results.

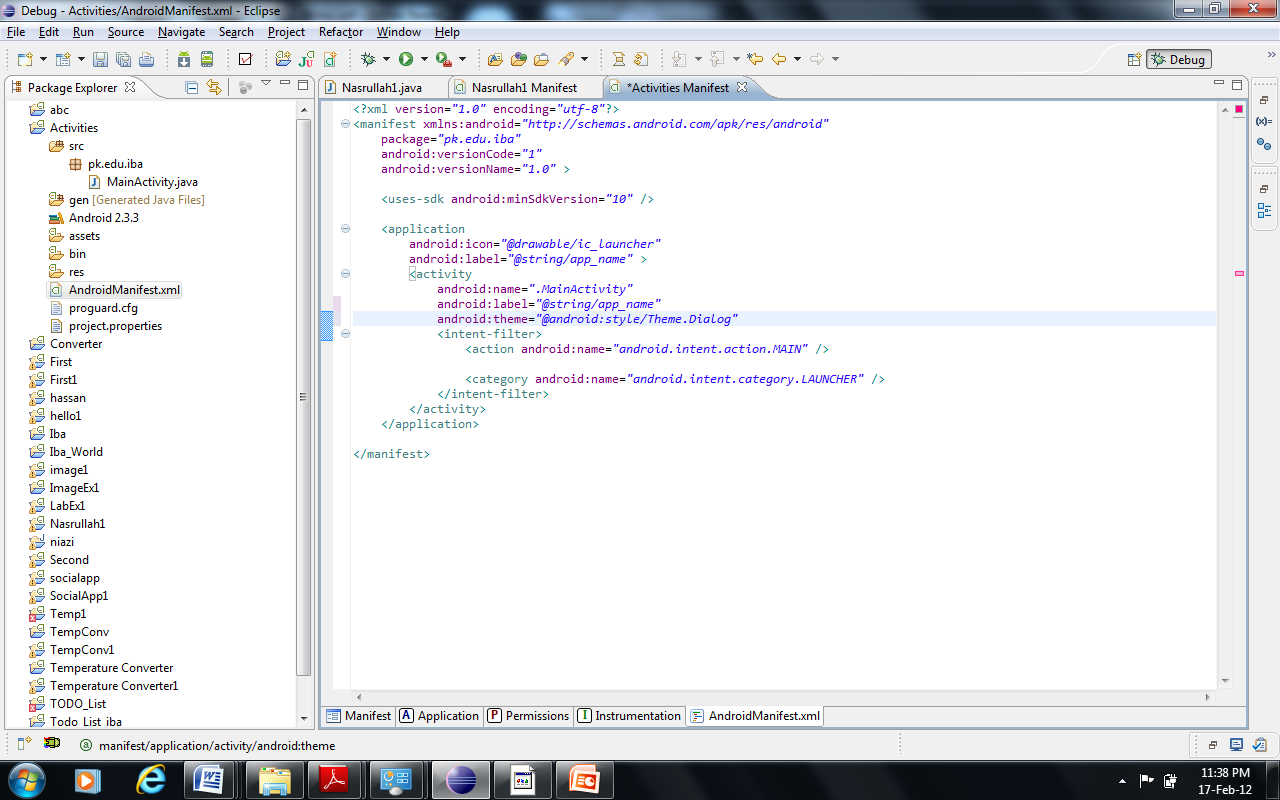
7)press the Phone button on the emulator so that activity is pushed towards the background,observe the results in the catlog and write down the results.

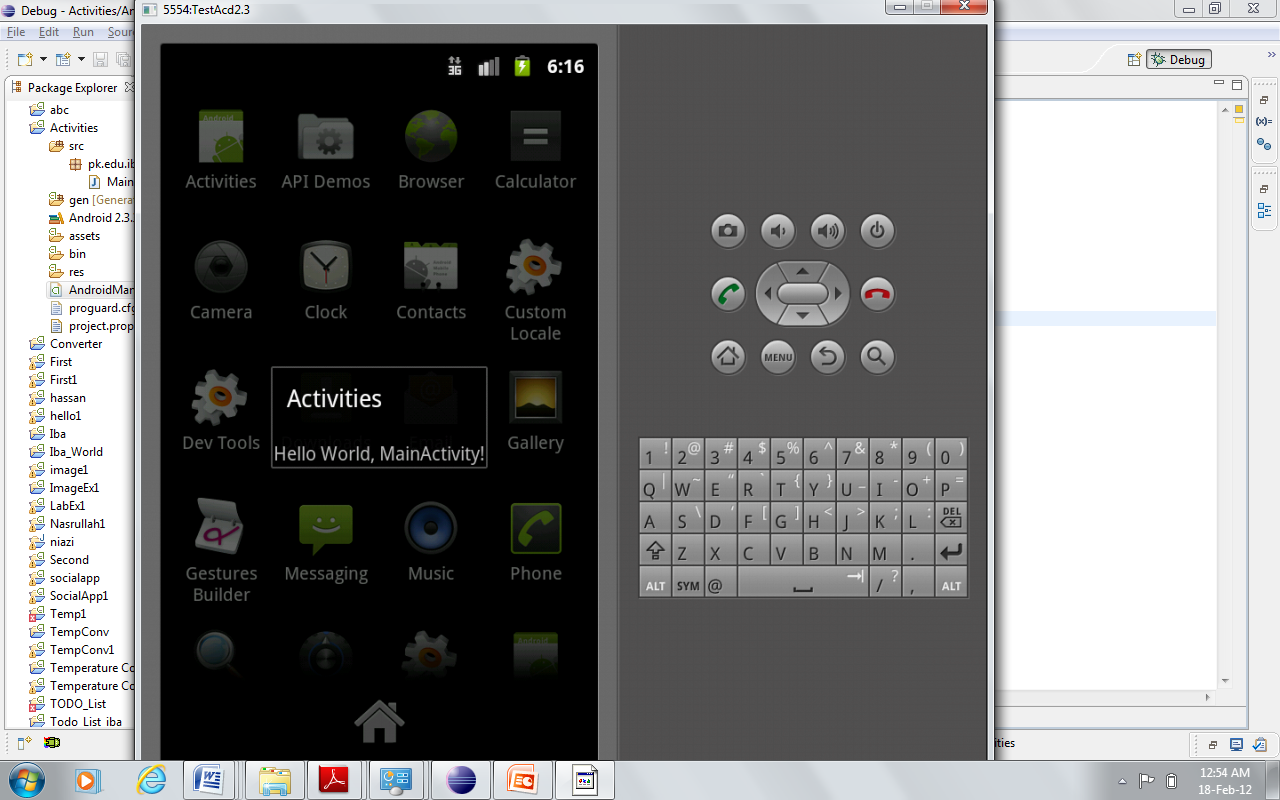
8)Exit the phone dialer by pressing the back button on the emulator ,the activity is visible again,observe the results in catlog

**Applying Styles and themes to Activity**

1. To apply a dialog theme to an activity, simply modify the <Activity> element in the AndroidManifest.xml file by adding the android:theme attribute:

android:theme=*"@android:style/Theme.Dialog"*



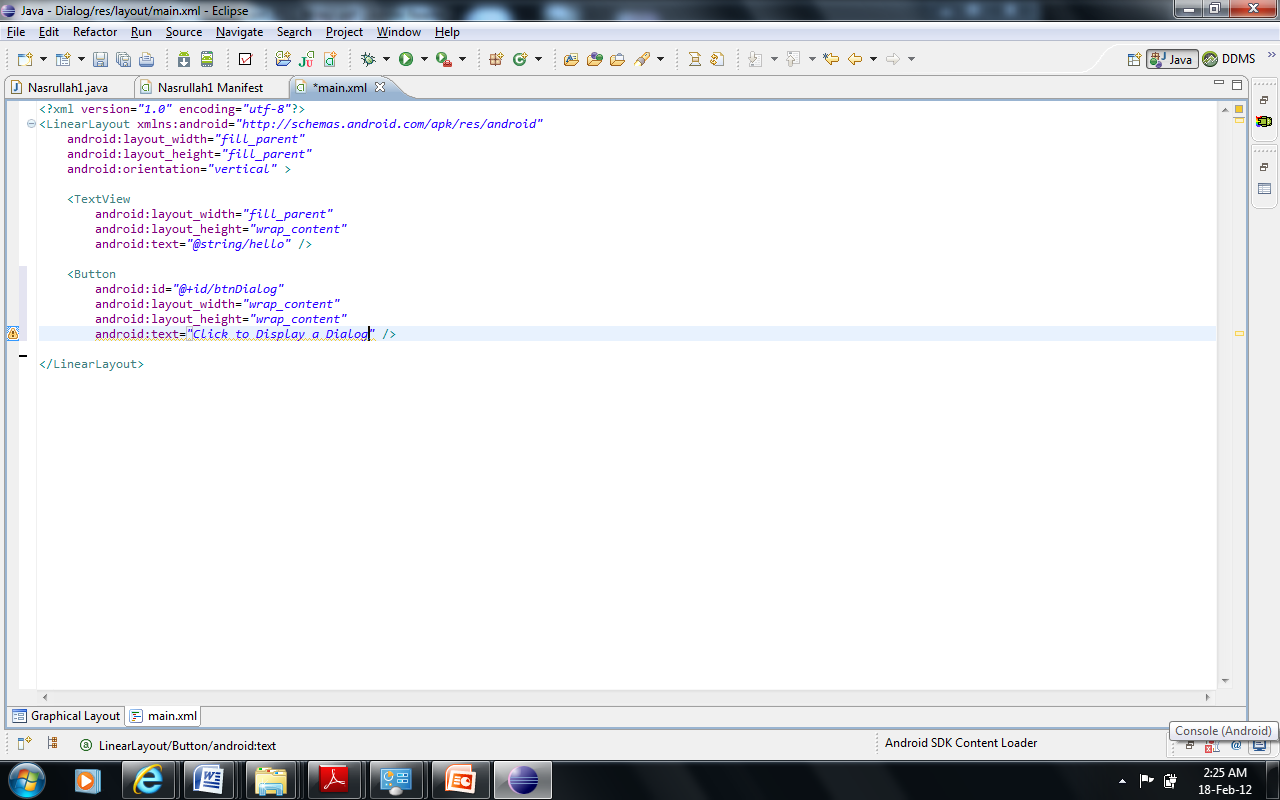


**Displaying a Dialog Window using an Activity**

1)create a new android project and name it Dialog.

2)Add the following statement in main.xml.

As shown in figure below



4

Add the following statements in bold to the MainActivity.java file:

import android.app.Activity;

import android.os.Bundle;

**import android.app.AlertDialog;**

**import android.app.Dialog;**

**import android.content.DialogInterface;**

**import android.view.View;**

**import android.widget.Button;**

**import android.widget.Toast;**

public class MainActivity extends Activity {

**CharSequence[] items = { “Google”, “Apple”, “Microsoft” };**

**boolean[] itemsChecked = new boolean [items.length];**

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

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

**Button btn = (Button) findViewById(R.id.btn\_dialog);**

**btn.setOnClickListener(new View.OnClickListener() {**

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

**showDialog(0);**

**}**

**});**

}

**@Override**

**protected Dialog onCreateDialog(int id) {**

**switch (id) {**

**case 0:**

**return new AlertDialog.Builder(this)**

**.setIcon(R.drawable.icon)**

**.setTitle(“This is a dialog with some simple text...”)**

**.setPositiveButton(“OK”, new**

**DialogInterface.OnClickListener() {**

**public void onClick(DialogInterface dialog,**

**int whichButton)**

**{**

**Toast.*makeText*(getBaseContext(),**

**“OK clicked!”, Toast.LENGTH\_SHORT).show();**

**}**

**})**

**.setNegativeButton(“Cancel”, new**

**DialogInterface.OnClickListener() {**

**public void onClick(DialogInterface dialog,**

**int whichButton)**

**{**

**Toast.*makeText*(getBaseContext(),**

**“Cancel clicked!”, Toast.LENGTH\_SHORT).show();**

**}**

**})**

**.setMultiChoiceItems(items, itemsChecked, new**

**DialogInterface.OnMultiChoiceClickListener() {**

**@Override**

**public void onClick(DialogInterface dialog, int which,**

**boolean isChecked) {**

**Toast.*makeText*(getBaseContext(),**

**items[which] + (isChecked ? “ checked!”:**

**“ unchecked!”),**

**Toast.LENGTH\_SHORT).show();**

**}**

**}**

**)**

**.create();**

**}**

**return null;**

**}**

}

Press F11 to debug the application on the Android Emulator. Click the button to display the dialog

(see Figure . Checking the various checkboxes will cause the Toast class to display the text of

the item checked/unchecked. To dismiss the dialog, click the OK or Cancel button.

