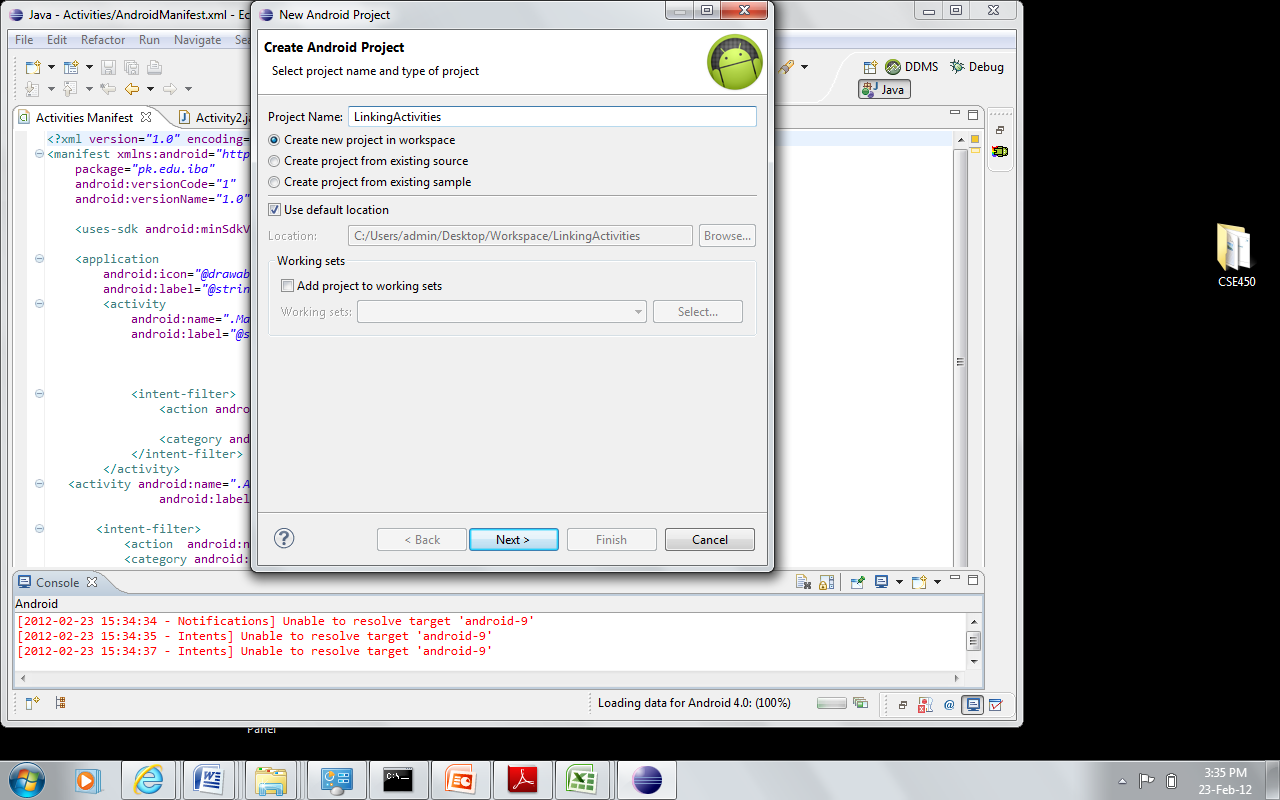
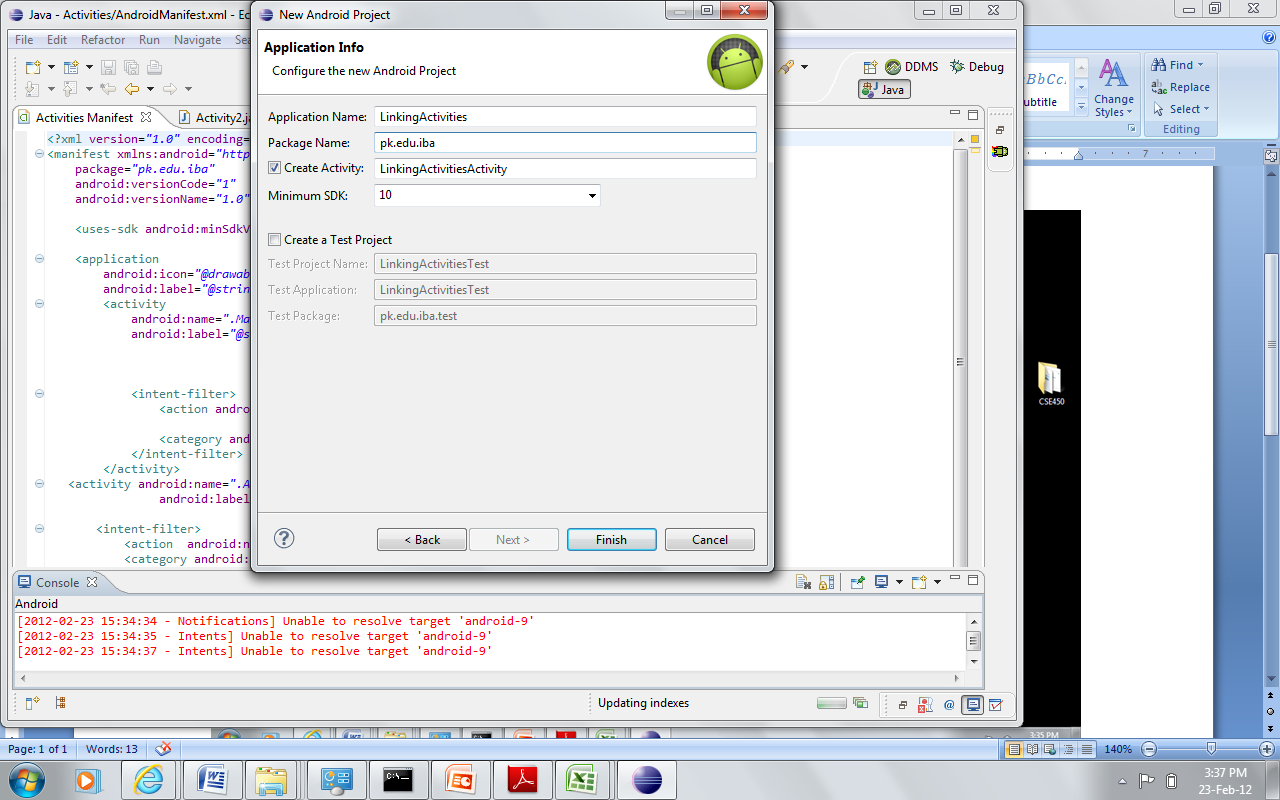
**INTENTS**

An android application can contain zero or more acitivities,when your application has more than one activity,you may need to navigate beween one activity to another activity.in android ,you navigate between activities through INTENTS.

**Activity 6(Linking Activities With intents)**

**1)**creat an android project in eclipse named LinkingActivites.





2)Add the following statements in the AndroidManifest.xml.

<activity android:name=*".Activity2"*

android:label=*"Nasrullah Activity 2"*>

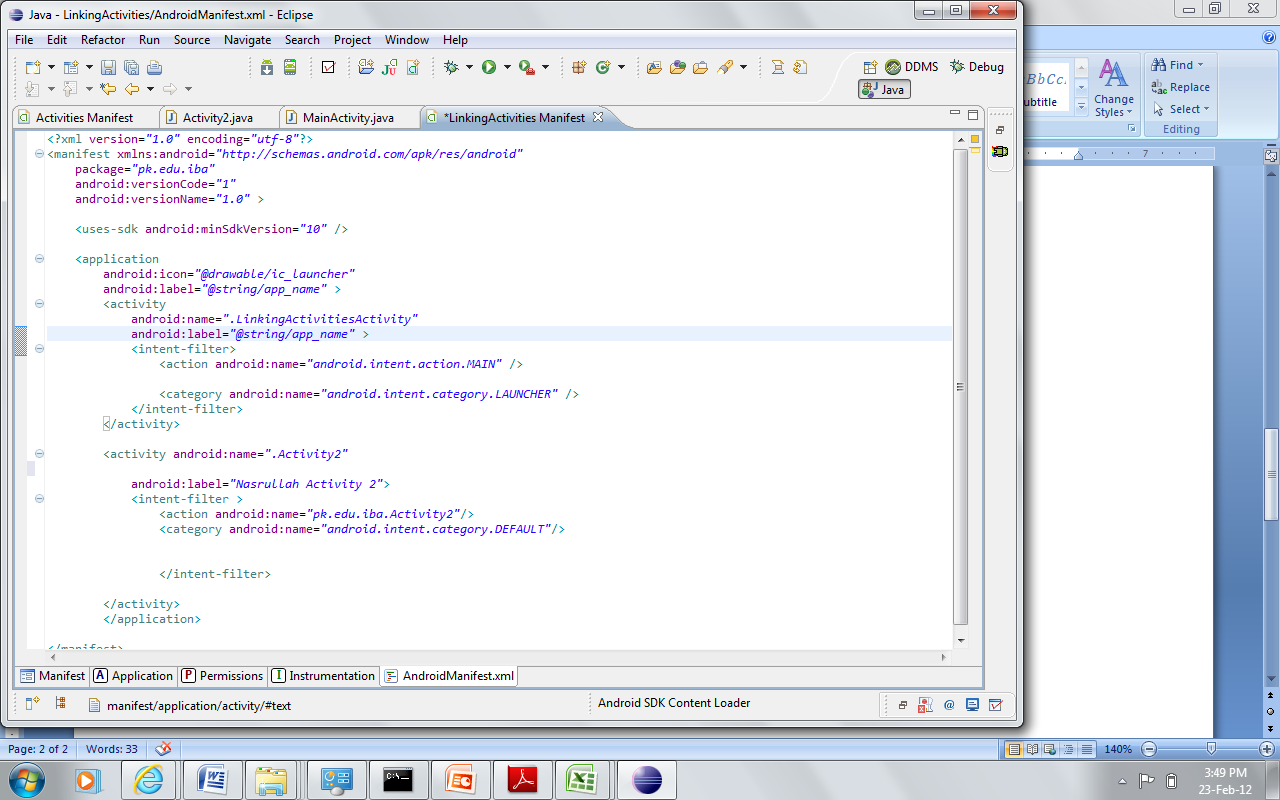
<intent-filter >

<action android:name=*"pk.edu.iba.Activity2"*/>

<category android:name=*"android.intent.category.DEFAULT"*/>

</intent-filter>

</activity> LIKE THIS FIGURE



NOTE THE FOLLOWING

YOU HAVE add the new application to the application

* The name of the new activity added is Activity2
* The label for the activity is named “Nasrullah Activity 2”
* The intent filter name for the activity is “pk.edu.iba.ACTIVITY2”. Other activities that wish

to call this activity will invoke it via this name. Ideally, you should use the reverse domain name of

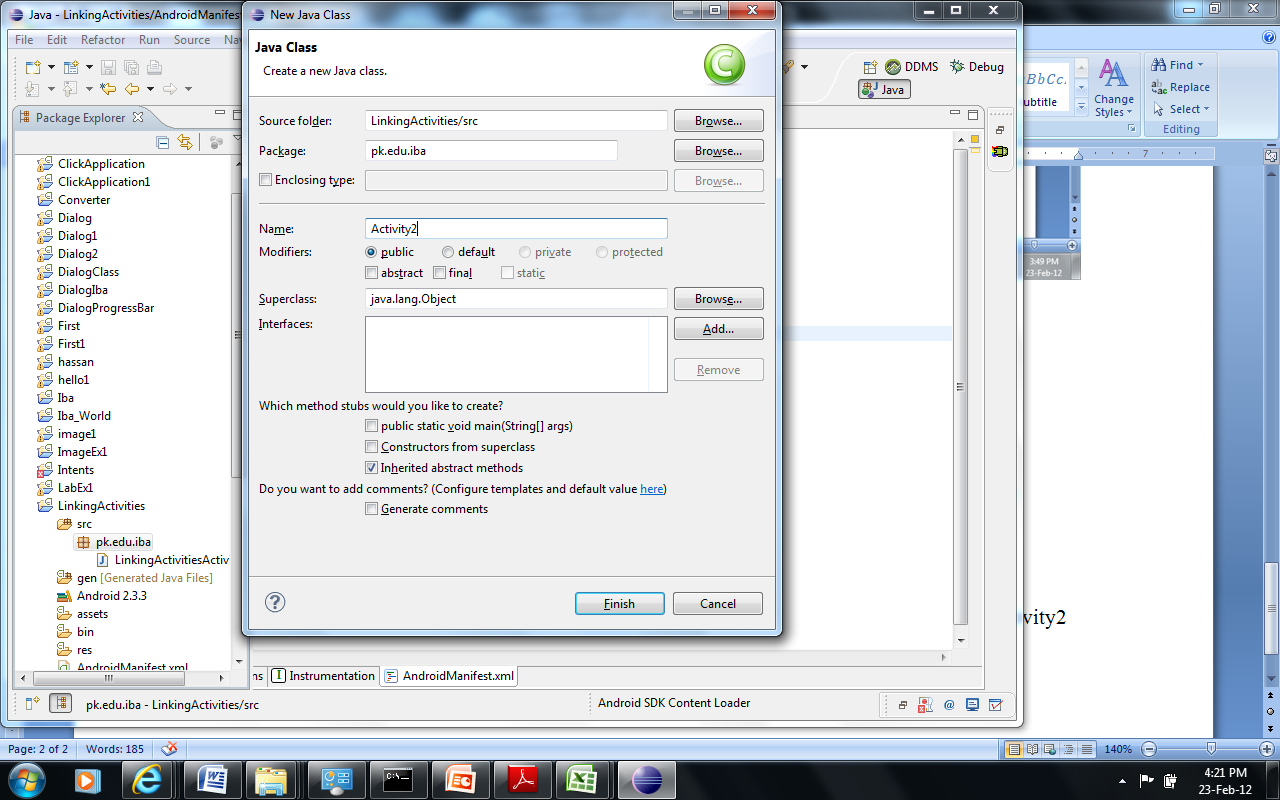
your company as we did (reverse of IBA) the intent filter name in order to reduce the chances of another application having the same intent filter.

* The category for the intent filter is “android.intent.category.DEFAULT”. You need to add this to

the intent filter so that this activity can be started by another activity using the startActivity()

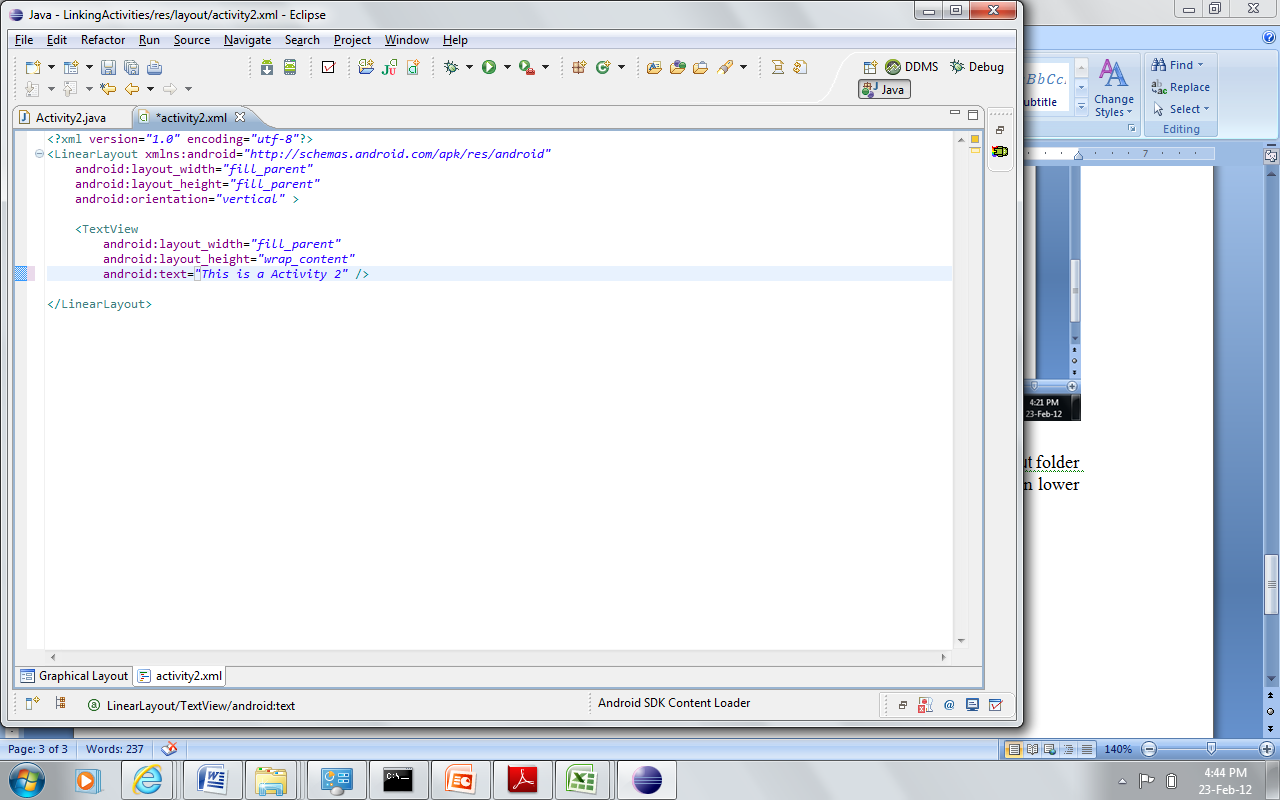
method

3)right click on the package name under src folder and select new 🡪 class ,name this class Activity2 and click finish as in figure below.



4) Make a copy of the main.xml file by right-clicking on it and selecting Copy. Then, right-click on the res/layout folder and select Paste. Name the file activity2.xml. The res/layout folder will now contain the activity2.xml file(name it in lower case otherwise it would raise error )

5)Modify the activity2.xml as follows



6) In the Activity2.java file, add the following statements in bold:

package pk.edu.iba;

**import android.app.Activity;**

**import android.os.Bundle;**

**public class Activity2 extends Activity {**

**@Override**

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

**super.onCreate(savedInstanceState);**

**setContentView(R.layout.activity2);**

**}**

**}**

**7).** Modify the MainActivity.java file as shown in bold:

package pk.edu.iba;

|  |
| --- |
| In the MainActivity.java file, you implemented the onKeyDown event handler. This event is fired whenever  the user presses one of the keys on the device. When the user presses the center key on the directional  pad (as represented by the KeyEvent.KEYCODE\_DPAD\_CENTER constant), you use the startActivity()  method to display Activity2 by creating an instance of the Intent class and passing it the intent filter  name of Activity2 (which is pk.edu.iba.ACTIVITY2): |

import android.app.Activity;

import android.os.Bundle;

import android.util.Log;

import android.view.Window;

import android.view.KeyEvent;

import android.content.Intent;

public class LinkingActivitiesActivity extends Activity {

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

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

}

public boolean onKeyDown(int keyCode,KeyEvent event){

if (keyCode==KeyEvent.KEYCODE\_DPAD\_CENTER ){

startActivity(new Intent("pk.edu.iba.Activity2"));

}

return false;

}

}

**8.** Press F11 to debug the application on the Android Emulator. When the first activity is loaded, click

the center of the directional pad (see Figure on a real device this can be achieved by pressing

down the trackball). The second activity will now be loaded.



**NOTES**

In the MainActivity.java file, you implemented the onKeyDown event handler. This event is fired whenever

the user presses one of the keys on the device. When the user presses the center key on the directional

pad (as represented by the KeyEvent.KEYCODE\_DPAD\_CENTER constant), you use the startActivity()

method to display Activity2 by creating an instance of the Intent class and passing it the intent filter

name of Activity2 (which is pk.edu.iba.ACTIVITY2):

public boolean onKeyDown(int keyCode, KeyEvent event)

{

if (keyCode == KeyEvent.KEYCODE\_DPAD\_CENTER)

{

**startActivity(new Intent(“iba.edu.pk.ACTIVITY2”));**

}

return false;

}

Activities in Android can be invoked by any application running on the device. For example, you can

create a new Android project and then display Activity2 by using its pk.edu.iba.ACTIVITY2

intent filter. This is one of the fundamental concepts in Android that enables an application to invoke

another easily.

If the activity that you want to invoke is defined within the same project, you can rewrite the preceding

statement like this:

Intents are instances of the android.content.Intent class.

**activities, services, and broadcast receivers — are activated through messages, called *intents*.**

**The intent itself, an** Intent **object, is a passive data structure holding an abstract description of an operation to be performed**

Intent filters are used by activities to describe the type of intents they want to receive.

**How to install your apk files on the emulator (use of android debug bridge)**

1. Go to sdk/platform-tools with the command cd /home/android-sdk/platform-tools

2List the devices with the command ./adb devices

3.You can install your file with ./adb install .filename.apk

