Notes from Week 1

Using TextView, EditText, CheckBox, and Buttons

**final** EditText edtValue = (EditText) findViewById(R.id.***edtValue***);  
**final** TextView txtResult = (TextView) findViewById(R.id.***txtResult***);  
Button btn1 = (Button) findViewById(R.id.***btn1***);  
Button btn2 = (Button) findViewById(R.id.***btn2***);  
Button btn3 = (Button) findViewById(R.id.***btn3***);  
Button btn4 = (Button) findViewById(R.id.***btn4***);  
**final** CheckBox toggle = (CheckBox) findViewById(R.id.***checkBox***);

Determining if a CheckBox is checked

**if** (toggle.isChecked())

Creating a listener for a Button

btnResult.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
   
 }  
});

Converting EditText input to a string and then to a double

**double** Value = Double.*parseDouble*(edtValue.getText().toString());

Converting a double to a string and displaying it in a TextView with other text

txtResult.setText(String.*format*(**"%.2f Grams = %.2f Oz"**, Value, Result));

To change Styles so that the Button will allow lowercase. Add shaded lines

<**style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar"**>  
 *<!-- Customize your theme here. -->* Other lines here …

<**item name="android:buttonStyle"**>@style/Button</**item**>  
</**style**>  
  
<**style name="Button" parent="Widget.AppCompat.Button"**>  
 <**item name="android:textAllCaps"**>false</**item**>  
</**style**>

To add XML file for styles – put this in the drawable folder

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

<selector xmlns:android="http://schemas.android.com/apk/res/android">

<item>

<shape android:shape="rectangle">

<solid android:color="#fcff4a"/>

<corners android:radius="2dp"/>

</shape>

</item>

</selector>

Then add the following to the XML of the component

**android:background="@drawable/edit\_bg" where edit\_bg.xml is the name of the file**

**Android Table Layout**.

* **Android TableLayout** arranges groups of views into rows and columns.
* You will use the <TableRow> element to build a row in the **table**.
* Each row has zero or more cells; each cell can hold one View object.
* Containers do not display border lines for their rows, columns, or cells.
* The table will have as many columns as the row with the most cells.
* A table can leave cells empty.
* Cells can span multiple columns, as they can in HTML.
* You can span columns by using the span keyword: e.g. android:layout\_span="2"
* android:gravity positions views within the cells.
* android:padding can indent views from the margins of the cells.

Calling new activities, calling them and passing parameters to the activity and receiving return values.

* Intents are used to call new Activities
* Activities are a combination of a new Java file and the associated XML files.
* Create new activities in Android with the New/Activity menu option. We will create new Basic Activities.
* This will create the appropriate Java and XML files for the new activity and put the necessary code in the Manifest for you.

Below is sample Java code showing calling an activity with no data being passed in either direction (About), calling an activity with data being only passed back to the current activity (CalcTip) and calling an activity that both receives and returns data (CalcShare).

public class MainActivity extends AppCompatActivity {

public static final int CalcTipCode = 1;

public static final int CalcShareCode = 2;

public static String strTip;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);

setSupportActionBar(toolbar);

}

protected void CalcTip(View V) { // Call CalculateTip and prepare for a return value

Intent i = new Intent(this, CalculateTip.class);

startActivityForResult(i, CalcTipCode); // Start activity so that we can receive data

}

**// Call CalculateShare both sending a value and preparing for a return value**

protected void CalcShare(View V) {

Intent i = new Intent(this, CalculateShare.class);

i.putExtra("tip",strTip); // send data to CalcShare activity

startActivityForResult(i, CalcShareCode); // Start activity so that we can receive data

}

**// Call About without sending or receiving data**

protected void About (View V) {

Intent i = new Intent(this, About.class);

startActivity(i);

}

**// This method is used to parse the return values that were received from the various //activities. Use requestCode to determine which activity returned this value**

public void onActivityResult(int requestCode, int resultCode, Intent data) {

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

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

String strShare="";

super.onActivityResult(requestCode, resultCode, data);

if(resultCode == RESULT\_OK) {

if (requestCode == CalcTipCode) {

strTip = data.getStringExtra("tipAmount");

txtTip.setText(strTip);

}

else if (requestCode == CalcShareCode) {

strShare = data.getStringExtra("newAmount");

txtShare.setText(strShare);

}

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CalculateTip start \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

public class CalculateTip extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_calculate\_tip);

Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);

setSupportActionBar(toolbar);

}

**// Returns a value to Main. sendData is called from a button using onClick**

protected void sendData(View V) {

EditText tipAmount = (EditText) findViewById(R.id.edtNum);

Intent retData=new Intent();

retData.putExtra("tipAmount", tipAmount.getText().toString());

setResult(RESULT\_OK, retData);

finish();

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CalculateTip end \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CalculateShare start \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

public class CalculateShare extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

String value="";

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_calculate\_share);

Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);

setSupportActionBar(toolbar);

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

**// Get the data passed to us from the MainActivity**

Bundle extras = getIntent().getExtras();

if (extras != null) {

value = extras.getString("tip");

}

txtAmount.setText(value);

}

**// Returns a value to Main. SendData is called from a button using onClick**

protected void sendData(View V) {

EditText newAmount = (EditText) findViewById(R.id.edtNewAmount);

Intent retData=new Intent();

retData.putExtra("newAmount", newAmount.getText().toString());

setResult(RESULT\_OK, retData);

finish();

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CalculateShare end \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* About start \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

public class About extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_about);

Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);

setSupportActionBar(toolbar);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* About end \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Spinner and GridView code

**public class** MainActivity **extends** AppCompatActivity {  
  
 **void** showToast(CharSequence msg) {  
 Toast.*makeText*(**this**, msg, Toast.***LENGTH\_SHORT***).show();  
 }  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 **final** String[] Strings = getResources().getStringArray(***StringList***);  
 **final** String[] Gridstrings = getResources().getStringArray(***GridList***);  
  
  
 *// This is to change the look of the spinner. It also adds the strings to the list* ArrayAdapter adapter = ArrayAdapter.*createFromResource*(**this**, R.array.***StringList***, R.layout.***spinner\_control***);  
 Spinner provinces = (Spinner) findViewById(R.id.***spinner***);  
 provinces.setAdapter(adapter);  
  
 provinces.setOnItemSelectedListener(  
 **new** AdapterView.OnItemSelectedListener() {  
 **public void** onItemSelected(  
 AdapterView<?> parent, View view, **int** position, **long** id) {  
 showToast(**"Spinner value = "** + Strings[position]);  
 }  
  
 **public void** onNothingSelected(AdapterView<?> parent) {  
  
 }  
 });  
  
*// This is to change the look of the grid. It also adds the strings to the grid* ArrayAdapter gridadapter = ArrayAdapter.*createFromResource*(**this**, R.array.***GridList***, R.layout.***grid\_control***);  
 GridView provincegrid = (GridView) findViewById(R.id.***grid1***);  
 provincegrid.setAdapter(gridadapter);  
  
 provincegrid.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  
 **public void** onItemClick(AdapterView<?> parent, View view,  
 **int** position, **long** id) {  
  
 showToast(**"Grid value = "** + Gridstrings[position]);  
  
  
 }  
 });  
 }  
}

ListView Code

**public class** MainActivity **extends** ListActivity {  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
  
 ListAdapter adapter = createAdapter();  
 setListAdapter(adapter);  
 }  
  
 */\*\*  
 \* Creates and returns a list adapter for the current list activity  
 \** ***@return*** *\*/* **protected** ListAdapter createAdapter()  
 {  
 **final** String[] Strings = getResources().getStringArray(***StringList***);  
  
 */\* Set ListView for Simple List  
 ListAdapter adapter = new ArrayAdapter<String>(this, android.R.layout.simple\_list\_item\_1, Strings);  
 \*/  
  
 /\* Set ListView for Single Choice  
 ListAdapter adapter = new ArrayAdapter<String>(this, android.R.layout.simple\_list\_item\_single\_choice, Strings);  
 ListView listView = getListView();  
 listView.setChoiceMode(AbsListView.CHOICE\_MODE\_SINGLE);  
 \*/  
  
 /\* Set listview for Multiple Choice mode  
 ListAdapter adapter = new ArrayAdapter<String>(this, android.R.layout.simple\_list\_item\_multiple\_choice, Strings);  
 ListView listView = getListView();  
 listView.setChoiceMode(AbsListView.CHOICE\_MODE\_MULTIPLE);  
 \*/*

*/\* Use custom XML for ListView\*/* ListAdapter adapter = **new** ArrayAdapter<String>(**this**, R.layout.***list\_layout***, R.id.***txtView***, Strings);  
   
 **return** adapter;  
  
 }  
  
 @Override  
 **protected void** onListItemClick (ListView l, View v, **int** position, **long** id) {  
 **final** String[] Strings = getResources().getStringArray(***StringList***);  
  
 Toast.*makeText*(**this**, **"Clicked "** + Strings[position], Toast.***LENGTH\_SHORT***).show();  
  
 }  
 }

XML Layouts for the above Views

list\_layout.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**TextView xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/txtView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:padding="10dp"  
 android:text="TextView"  
 android:textColor="@android:color/holo\_red\_light"  
 android:textSize="24sp"** />

spinner\_control.xml

*<?***xml version="1.0" encoding="utf-8"***?>* <**TextView xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/text1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="24sp"  
 android:textColor="#2E86C1"** />

grrid\_control.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**TextView xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/text1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="14sp"  
 android:textColor="#ff0000"** />

**Playing music and displaying web pages in a browser**

To play a song (Note that the code to release the song is not shown)

// Read songs into an array in the res/raw folder

Integer[] Songs = {R.raw.*song1*, R.raw.*song2,* R.raw.*song3*};

MediaPlayer mpSong = **new** MediaPlayer();

// Load the first song into the MediaPlayer

mpSong = MediaPlayer.*create*(**this**, Songs[0]);

mpSong.start();

To display a web page

Intent myIntent = new Intent();  
myIntent.setAction(Intent.ACTION\_VIEW);  
myIntent.setData(android.net. Uri.parse(website\_string));

-- or --

Intent myIntent = new Intent(Intent.ACTION\_VIEW, Uri.parse(website\_string));

**Shared Preferences**

**Saving**

**public void** onStop() {  
 **super**.onStop();  
  
 EditText empname = (EditText) findViewById(R.id.***edtName***);  
 EditText empage = (EditText) findViewById(R.id.***edtAge***);  
 ToggleButton empfulltime = (ToggleButton) findViewById(R.id.***fullTime***);  
  
 SharedPreferences.Editor editor = getSharedPreferences(***EMP\_DATA***,***MODE\_PRIVATE***).edit();  
  
 *// Write the shared preferences values and commit them* editor.putString(**"name"**,empname.getText().toString());  
 editor.putInt(**"age"**,Integer.*parseInt*(empage.getText().toString()));  
 editor.putBoolean(**"fulltime"**, empfulltime.isChecked());  
 editor.commit();  
}

**Accessing**

**public static final** String ***EMP\_DATA*** = **"MyPrefsFile"**;  
@Override  
**protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
   
 *// Declare Shared Preferences instance* SharedPreferences prefs = getSharedPreferences(***EMP\_DATA***, ***MODE\_PRIVATE***);  
  
 *// Read the Shared Preferences vales* String name = prefs.getString(**"name"**,**"none"**);  
 **int** age = prefs.getInt(**"age"**,0);  
 **boolean** fulltime = prefs.getBoolean(**"fulltime"**,**false**);  
  
 EditText empname = (EditText) findViewById(R.id.***edtName***);  
 EditText empage = (EditText) findViewById(R.id.***edtAge***);  
 ToggleButton empfulltime = (ToggleButton) findViewById(R.id.***fullTime***);  
  
 *// If the values are not default, then they exist so use them* **if** (!Objects.*equals*(name, **"none"**))  
 empname.setText(name);  
  
 **if** (age > 0) {  
 empage.setText(Integer.*toString*(age));  
 empfulltime.setChecked(fulltime);  
 }

}

**Android Files**

**Saving to a text file**

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

EditText empname = (EditText) findViewById(R.id.***edtName***);  
 String data = empname.getText().toString();  
 **try** {  
 // Character Reading

/\* *FileOutputStream fOut = getApplicationContext().openFileOutput("myfile.txt", MODE\_PRIVATE);  
 fOut.write(data.getBytes());  
 fOut.close();  
 Toast.makeText(getBaseContext(),"File Saved", Toast.LENGTH\_SHORT).show();\*/*

*// Line Reading* FileOutputStream fOut = getApplicationContext().openFileOutput(**"myfile.txt"**, ***MODE\_APPEND***);  
 BufferedWriter writer = **new** BufferedWriter(**new** OutputStreamWriter(fOut));  
 writer.write(data);  
 writer.newLine();  
 writer.flush();  
 writer.close();  
  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
}

**Reading from a text file**

**public void** read(View v){  
 **try** {

// Character reading */\* FileInputStream fin = getApplicationContext().openFileInput("myfile.txt");  
 int c;  
 String line="";  
 while( (c = fin.read()) != -1){  
 line += Character.toString((char)c);  
 }  
 TextView inString = (TextView) findViewById(R.id.txtRead);  
 inString.setText(line);  
 Toast.makeText(getBaseContext(), "File Read",  
 Toast.LENGTH\_SHORT).show();  
 \*/  
  
 // Line Reading* FileInputStream fin = getApplicationContext().openFileInput(**"myfile.txt"**);  
 BufferedReader reader = **new** BufferedReader(**new** InputStreamReader(fin));  
 TextView inString = (TextView) findViewById(R.id.***txtRead***);  
 String line, data=**""**;  
 **while**((line = reader.readLine()) != **null**){  
 data = data + line + **"\n"**;  
 }  
 inString.setText(data);  
  
  
 } **catch**(Exception e) { }  
}

**Fragments**

**Main activity Java code**

*public class MainActivity extends AppCompatActivity {*

*protected void onCreate(Bundle savedInstanceState) {*

*super.onCreate(savedInstanceState);*

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

*final EditText textValue = (EditText) findViewById(R.id.edtText);*

*final EditText textSize = (EditText) findViewById(R.id.edtSize);*

*Button btnText = (Button) findViewById(R.id.btnText);*

*Button btnSquare = (Button) findViewById(R.id.btnSquare);*

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

*public void onClick(View v) {*

*TextFragment textFragment = (TextFragment) getFragmentManager().findFragmentById(R.id.text\_fragment);*

*SquareFragment squareFragment = (SquareFragment) getFragmentManager().findFragmentById(R.id.square\_fragment);*

*int fontsize = Integer.parseInt(textSize.getText().toString());*

*String text = textValue.getText().toString();*

*textFragment.changeTextProperties(fontsize, text);*

*FragmentTransaction ft = getFragmentManager().beginTransaction();*

*ft.hide(squareFragment);*

*ft.show(textFragment);*

*ft.commit();*

*}*

*});*

**Fragment Java code**

*public class TextFragment extends Fragment {*

*private static TextView textview;*

*public View onCreateView(LayoutInflater inflater,ViewGroup container, Bundle savedInstanceState) {*

*View view = inflater.inflate(R.layout.textfragment,container, false);*

*textview = (TextView) view.findViewById(R.id.textView1);*

*return view;*

*}*

*public void changeTextProperties(int fontsize, String text)*

*{*

*textview.setTextSize(fontsize);*

*textview.setText(text);*

*}*

*}*

**Fragment 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" >**

**<TextView**

**android:id="@+id/textView1"**

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

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

**android:text="Fragment One"**

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

**android:layout\_centerVertical="true"**

**android:layout\_alignParentEnd="true"**

**android:layout\_marginEnd="66dp" />**

**<TextView**

**android:id="@+id/textView2"**

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

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

**android:layout\_alignBottom="@+id/textView1"**

**android:layout\_marginEnd="42dp"**

**android:layout\_toStartOf="@+id/textView1"**

**android:text="Number:"**

**android:textSize="24sp" />**

**</RelativeLayout>**

**Main activity XML for fragment**

**<fragment**

**android:id="@+id/text\_fragment"**

**android:name="com.rpmicro.multiplefragments.TextFragment"**

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

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

**android:layout\_marginTop="12dp"**

**android:layout\_below="@+id/btnSquare"**

**android:layout\_centerHorizontal="true" />**

**Setting up a List Fragment**

**XML creating fragment in main activity**

// When the fragment is created, celect the MyListFragment class to base it on

<**fragment  
 android:id="@+id/fragment"  
 android:name="com.heritage.rpatterson.orientations.MyListFragment"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"** />

**MyListFragment.java**

**public class** MyListFragment **extends** ListFragment **implements** AdapterView.OnItemClickListener {  
  
 **public** View onCreateView(LayoutInflater inflater,  
 ViewGroup container, Bundle savedInstanceState) {  
 View view = inflater.inflate(R.layout.***list\_fragment***, container, **false**);  
 **return** view;  
 }  
  
 **public void** onActivityCreated(Bundle savedInstanceState) {  
 **super**.onActivityCreated(savedInstanceState);  
 ArrayAdapter adapter = ArrayAdapter.*createFromResource*(getActivity(),  
 R.array.***Planets***, android.R.layout.***simple\_list\_item\_1***);  
 setListAdapter(adapter);  
 getListView().setOnItemClickListener(**this**);  
 }  
  
 **public void** onItemClick(AdapterView<?> parent, View view, **int** position,**long** id) {  
 Toast.*makeText*(getActivity(), **"Item: "** +

position,Toast.***LENGTH\_SHORT***).show();  
 }

}

**list\_fragment.xml**

<**LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"** >  
  
 <**ListView  
 android:id="@android:id/list"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"** >  
 </**ListView**>  
   
</**LinearLayout**>

**Add this line to the activity section of the Manifest file to allow orientation changes**

**android:configChanges="keyboardHidden|orientation|screenSize"**

**Reading a text file stored in the assets folder. Access one line at a time.**

**public void** readfile(View v) {  
 AssetManager assetManager = getAssets();  
 **try** {  
 InputStream inputStream = assetManager.open(**"easywords.txt"**);  
 InputStreamReader inputStreamReader = **new** InputStreamReader(inputStream);  
 BufferedReader bufferedReader = **new** BufferedReader(inputStreamReader);  
 TextView out = (TextView) findViewById(R.id.***txtOut***);  
 String mLine, newmLine;  
 **while** ((mLine = bufferedReader.readLine()) != **null**) {  
 newmLine = out.getText().toString() + **"\n"** + mLine;  
 out.setText(newmLine);  
 }  
 }  
 **catch** (IOException ex) {  
 ex.printStackTrace();  
 }  
}

**Note:** The file must be in the asset folder at this location “app\source\main\assets”

**Accessing sqLite databases**

**public class** DatabaseHandler **extends** SQLiteOpenHelper {

**public static final int *DATABASE\_VERSION*** = 1;  
 **public static final** String ***DATABASE\_NAME*** = **"MyContacts.db"**;  
 **public static final** String ***CONTACTS\_TABLE\_NAME*** = **"contacts"**;  
 **public static final** String ***CONTACTS\_COLUMN\_ID*** = **"id"**;  
 **public static final** String ***CONTACTS\_COLUMN\_NAME*** = **"name"**;  
 **public static final** String ***CONTACTS\_COLUMN\_PHONE*** = **"phone"**;  
  
 **public** DatabaseHandler(Context context) {  
 **super**(context, ***DATABASE\_NAME***, **null**, ***DATABASE\_VERSION***);  
 }  
  
 @Override  
 **public void** onCreate(SQLiteDatabase db) {  
 db.execSQL(  
 **"create table "** + ***CONTACTS\_TABLE\_NAME*** +  
 **" ("** + ***CONTACTS\_COLUMN\_ID*** + **" integer primary key, "** +  
 ***CONTACTS\_COLUMN\_NAME*** + **" text, "** +  
 ***CONTACTS\_COLUMN\_PHONE*** + **" text)"** );  
 }  
  
 @Override  
 **public void** onUpgrade(SQLiteDatabase db, **int** oldVersion, **int** newVersion) {  
 *//* ***TODO Auto-generated method stub*** db.execSQL(**"DROP TABLE IF EXISTS "** + ***CONTACTS\_TABLE\_NAME***);  
 onCreate(db);  
 }  
  
 **public boolean** insertContact (String name, String phone)  
 {  
 SQLiteDatabase db = **this**.getWritableDatabase();  
 ContentValues contentValues = **new** ContentValues();  
  
 contentValues.put(***CONTACTS\_COLUMN\_NAME***, name);  
 contentValues.put(***CONTACTS\_COLUMN\_PHONE***, phone);  
  
 db.insert(***CONTACTS\_TABLE\_NAME***, **null**, contentValues);  
 **return true**;  
 }  
  
 **public** Cursor getData(**int** id){  
 SQLiteDatabase db = **this**.getReadableDatabase();  
 Cursor res = db.rawQuery( **"select \* from "** + ***CONTACTS\_TABLE\_NAME*** + **" where id="**+id+**""**, **null** );  
 **return** res;  
 }  
  
 **public int** numberOfRows(){  
 SQLiteDatabase db = **this**.getReadableDatabase();  
 **return** (**int**) DatabaseUtils.*queryNumEntries*(db, ***CONTACTS\_TABLE\_NAME***);  
 }  
  
 **public boolean** updateContact (Integer id, String name, String phone)  
 {  
 SQLiteDatabase db = **this**.getWritableDatabase();  
 ContentValues contentValues = **new** ContentValues();  
 contentValues.put(***CONTACTS\_COLUMN\_NAME***, name);  
 contentValues.put(***CONTACTS\_COLUMN\_PHONE***, phone);  
 db.update(***CONTACTS\_TABLE\_NAME***, contentValues, **"id = ? "**, **new** String[] { Integer.*toString*(id) } );  
 **return true**;  
 }  
  
 **public boolean** deleteContact (Integer id)  
 {  
 SQLiteDatabase db = **this**.getWritableDatabase();  
 db.delete(***CONTACTS\_TABLE\_NAME***, **"id = ? "**, **new** String[]{Integer.*toString*(id)});  
 **return true**;  
 }

**public** ArrayList getAllIDs()  
 {  
 ArrayList array\_list = **new** ArrayList();  
 SQLiteDatabase db = **this**.getReadableDatabase();  
 Cursor res = db.rawQuery( **"select \* from "** + ***CONTACTS\_TABLE\_NAME***, **null** );  
 res.moveToFirst();  
 **while**(res.isAfterLast() == **false**){  
 String contact = res.getString(res.getColumnIndex(***CONTACTS\_COLUMN\_ID***));  
 array\_list.add(contact);  
 res.moveToNext();  
 }  
 **return** array\_list;  
 }  
  
}