In Class Practice Test

Due Date: Wednesday, Week 13.

Purpose: The purpose of this practice test to:

1. Develop an Android application with data access capabilities.

References: Textbook, ppt slides. This material provides the necessary information that you need to complete the exercises.

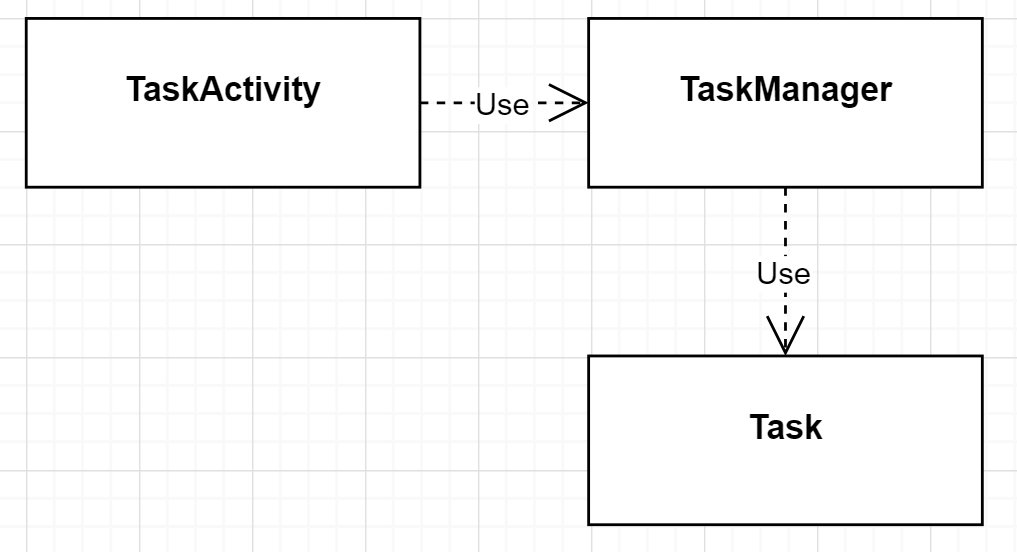
Be sure to read the following general instructions carefully:

- This practice test must be completed individually by all the students.

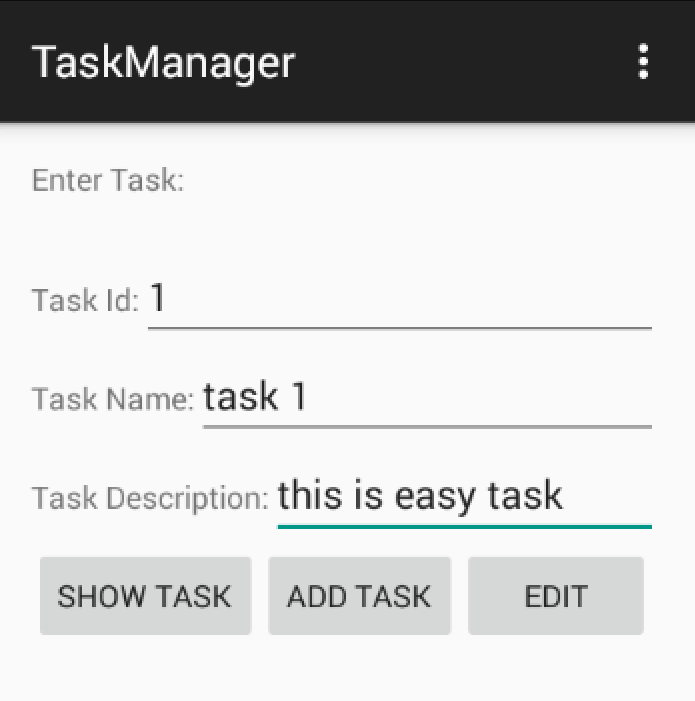
- You will have to **implement &** **demonstrate your solution in a scheduled lab session**.

**Exercise 1**

**In this exercise you are going to create a task manager app:**

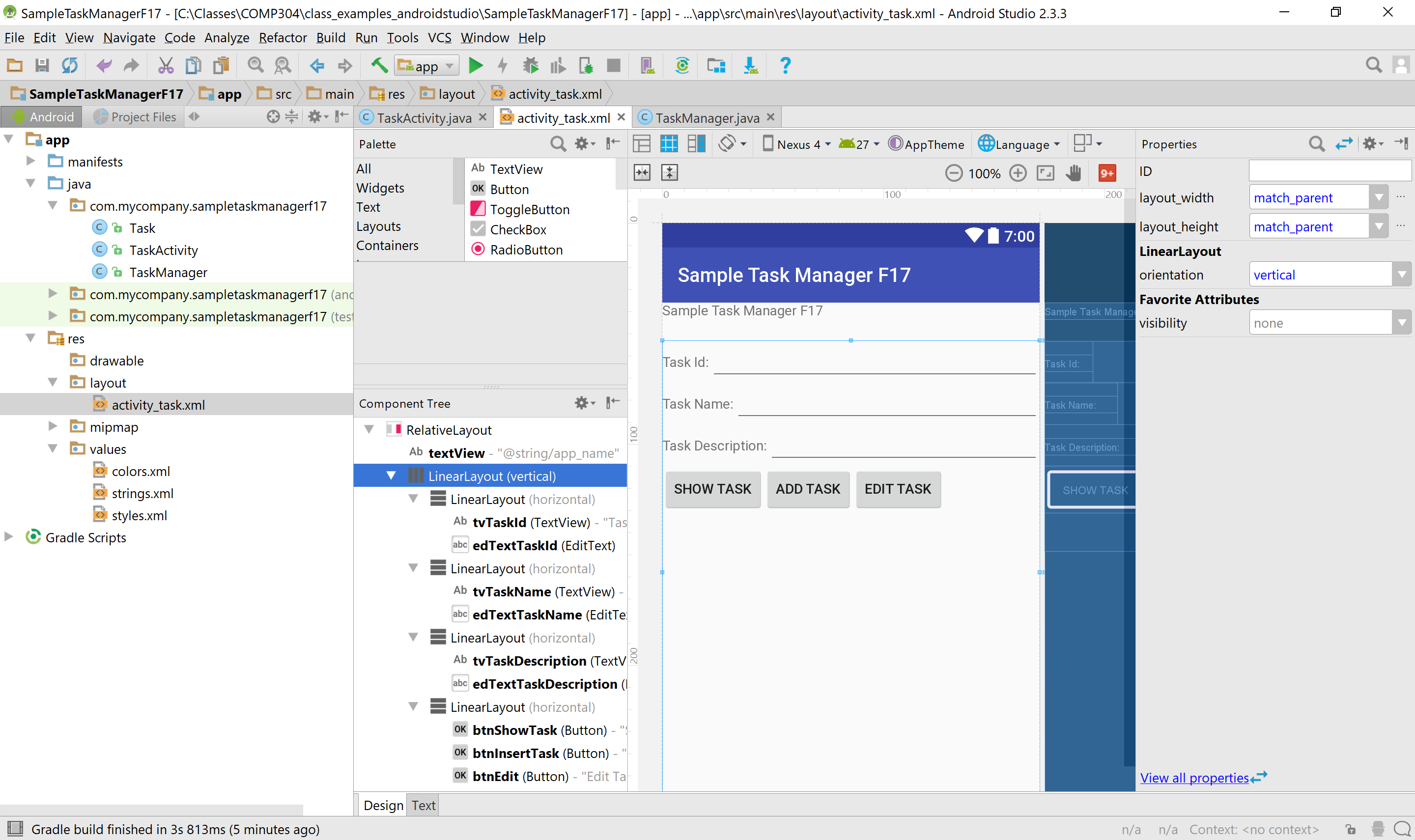
****

Create a new project using usual notations (John\_Smith\_PracticeTest). Name the blank activity TaskActivity. Create the following UI:



Use a LinearLayout with vertical orientation as the outer container.

Use four LinearLayouts with horizontal orientations to hold the UI elements as shown in the picture above (one LinearLayout per each line). Use the drag and drop from the Android Studio palette to create the UI. Use the notation shown in the picture below:



Create a new class named Task. Declare the following instance variables:

*//private fields***private int taskId**;  
**private** String **taskName**, **taskDescription**;

Generate constructors, getter & setters methods for this class. Use Code menu/generate to generate the code. Make sure you have a default constructor and a constructor with three arguments.

Create a new class named TaskManager. This class will contain SQLite functionalities, including database and table creation, getting task information, as well as adding and editing tasks.

Use the following code:

**package** com.mycompany.sampletaskmanagerf17;  
  
**import** android.content.ContentValues;  
**import** android.content.Context;  
**import** android.database.Cursor;  
**import** android.database.DatabaseUtils;  
**import** android.database.sqlite.SQLiteDatabase;  
**import** android.database.sqlite.SQLiteOpenHelper;  
  
*/\*\*  
 \* Created by inika on 1/1/2018.  
 \*/***public class** TaskManager **extends** SQLiteOpenHelper {  
 *//database name and version* **private static final** String ***DATABASE\_NAME*** = **"TaskDB"**;  
 **private static final int *DATABASE\_VERSION*** = 1;  
 *// table name and table creator string (SQL statement to create the table)  
 // should be set from within main activity* **private static** String *tableName*;  
 **private static** String *tableCreatorString*;  
 *//  
 // no-argument constructor* **public** TaskManager(Context context)  
 {  
 **super**(context, ***DATABASE\_NAME*** , **null**, ***DATABASE\_VERSION***);  
 }  
 *// Called when the database is created for the first time.  
 // This is where the creation of tables and the initial population  
 // of the tables should happen.* @Override  
 **public void** onCreate(SQLiteDatabase db) {  
 *//create the table* db.execSQL(*tableCreatorString*);  
 }  
 *//  
 // Called when the database needs to be upgraded.  
 // The implementation should use this method to drop tables,  
 // add tables, or do anything else it needs to upgrade  
 // to the new schema version.* @Override  
 **public void** onUpgrade(SQLiteDatabase db, **int** oldVersion, **int** newVersion) {  
 *//drop table if existed* db.execSQL(**"DROP TABLE IF EXISTS "** + *tableName*);  
 *//recreate the table* onCreate(db);  
 }  
 *//  
 //  
 //  
 // initialize table names and CREATE TABLE statement  
 // called by activity to create a table in the database  
 // The following arguments should be passed:  
 // tableName - a String variable which holds the table name  
 // tableCreatorString - a String variable which holds the CREATE Table statement* **public void** dbInitialize(String tableName, String tableCreatorString)  
 {  
 **this**.*tableName*=tableName;  
 **this**.*tableCreatorString*=tableCreatorString;  
  
 }  
 *//  
 // CRUD Operations  
 //  
 //This method is called by the activity to add a row in the table  
 // The following arguments should be passed:  
 // values - a ContentValues object that holds row values* **public boolean** addRow (ContentValues values) **throws** Exception {  
 SQLiteDatabase db = **this**.getWritableDatabase();  
 *// Insert the row* **long** nr= db.insert(*tableName*, **null**, values);  
 db.close(); *//close database connection* **return** nr> -1;  
 }  
  
 *// This method returns a task object which holds the table row with the given id  
 // The following argument should be passed:  
 // id - an Object which holds the primary key value  
 // fieldName - the name of the primary key field* **public** Task getTaskById(Object id, String fieldName) **throws** Exception{  
 SQLiteDatabase db = **this**.getReadableDatabase();  
 Cursor cursor = db.rawQuery( **"select \* from "** + *tableName* + **" where "**+ fieldName + **"='"**+String.*valueOf*(id)+**"'"**, **null** );  
 Task task = **new** Task(); *//create a new Task object* **if** (cursor.moveToFirst()) { *//if row exists* cursor.moveToFirst(); *//move to first row  
 //initialize the instance variables of task object* task.setTaskId(cursor.getInt(0));  
 task.setTaskName(cursor.getString(1));  
 task.setTaskDescription(cursor.getString(2));  
 cursor.close();  
  
 } **else** {  
 task = **null**;  
 }  
 db.close();  
 **return** task;  
  
 }  
 *//  
 //  
 // The following argument should be passed:  
 // id - an Object which holds the primary key value  
 // fieldName - the name of the primary key field  
 // values - a ContentValues object that holds row values* **public boolean** editRow (Object id, String fieldName, ContentValues values) **throws** Exception {  
 SQLiteDatabase db = **this**.getWritableDatabase();  
 *//* **int** nr = db.update(*tableName*, values, fieldName + **" = ? "**, **new** String[]{String.*valueOf*(id)});  
 **return** nr > 0;  
 }  
  
}

Make sure to change the package name to your package.

Now you have to write the code for TaskActivity class. Here is my code:

**package** com.mycompany.sampletaskmanagerf17;  
  
**import** android.content.ContentValues;  
**import** android.database.Cursor;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.util.Log;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.Toast;  
  
**public class** TaskActivity **extends** AppCompatActivity {  
  
 **private** TaskManager **taskManager**;  
 **private** EditText **txtId**, **txtTaskName**, **txtTaskDescription** ;  
 **private** Button **btnAdd**, **btnShow**, **btnEdit**;  
 **private final static** String ***TABLE\_NAME*** = **"AndroidTask"**;  
  
 **private static final** String ***tableCreatorString*** =  
 **"CREATE TABLE "**+ ***TABLE\_NAME*** + **" (taskId integer primary key, taskName text,taskDescription text);"**;  
 *//  
  
 //* @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_task***);  
 *//* **txtId** = (EditText) findViewById(R.id.***edTextTaskId***);  
 **txtTaskName** = (EditText) findViewById(R.id.***edTextTaskName***);  
 **txtTaskDescription** = (EditText) findViewById(R.id.***edTextTaskDescription***);  
 *//* **btnAdd** = (Button)findViewById(R.id.***btnInsertTask***);  
 **btnShow** = (Button)findViewById(R.id.***btnShowTask***);  
 **btnEdit** = (Button)findViewById(R.id.***btnEdit***);  
 *//  
 // initialize the tables* **try** {  
 **taskManager** = **new** TaskManager(**this**);  
 *//create the table* **taskManager**.dbInitialize(***TABLE\_NAME***, ***tableCreatorString***);  
 }  
 **catch**(Exception exception)  
 {  
 Toast.*makeText*(TaskActivity.**this**,  
 exception.getMessage(), Toast.***LENGTH\_SHORT***).show();  
 Log.*i*(**"Error: "**,exception.getMessage());  
 } }  
  
 **public void** showTask(View v)  
 {  
 **try** {  
 Task task = **taskManager**.getTaskById(**txtId**.getText().toString(), **"taskId"**);  
 **txtTaskName**.setText(task.getTaskName());  
 **txtTaskDescription**.setText(task.getTaskDescription());  
 }  
 **catch** (Exception exception)  
 {  
 Toast.*makeText*(TaskActivity.**this**,  
 exception.getMessage(), Toast.***LENGTH\_SHORT***).show();  
 Log.*i*(**"Error: "**,exception.getMessage());  
  
 }  
 }  
 *//* **public void** addTask(View v)  
 {  
 *//read values* **int** taskId = Integer.*parseInt*(**txtId**.getText().toString());  
 String taskName = **txtTaskName**.getText().toString();  
 String taskDescription = **txtTaskDescription**.getText().toString();  
 *//initialize ContentValues object with the new task* ContentValues contentValues = **new** ContentValues();  
 contentValues.put(**"taskId"**,taskId);  
 contentValues.put(**"taskName"**,taskName);  
 contentValues.put(**"taskDescription"**,taskDescription);  
 *//* **try** {  
 **taskManager**.addRow(contentValues);  
 }  
 **catch**(Exception exception)  
 {  
 *//* Toast.*makeText*(TaskActivity.**this**,  
 exception.getMessage(), Toast.***LENGTH\_SHORT***).show();  
 Log.*i*(**"Error: "**,exception.getMessage());  
  
 }  
 }  
  
 **public void** editTask(View v)  
 {  
 **int** taskId = Integer.*parseInt*(**txtId**.getText().toString());  
  
 String taskName = **txtTaskName**.getText().toString();  
 String taskDescription = **txtTaskDescription**.getText().toString();  
 **try**{  
 ContentValues contentValues = **new** ContentValues();  
 contentValues.put(**"taskId"**,taskId);  
 contentValues.put(**"taskName"**,taskName);  
 contentValues.put(**"taskDescription"**,taskDescription);  
 *//edit the row* **boolean** b=**taskManager**.editRow(taskId, **"taskId"**, contentValues);  
  
  
 }  
 **catch**(Exception exception)  
 {  
 Toast.*makeText*(TaskActivity.**this**,  
 exception.getMessage(), Toast.***LENGTH\_SHORT***).show();  
 Log.*i*(**"Error: "**,exception.getMessage());  
  
 }  
  
  
 }  
  
}

Run the application to test the database operations.



After testing the application, apply the following changes:

1. Add a delete task functionality to TaskManager class:
   1. First create a deleteTask method in TaskManager class:

*// delete a row***public void** deleteRow(Object id, String fieldName) **throws** Exception {  
 SQLiteDatabase db = **this**.getWritableDatabase();  
 db.delete(*tableName*, fieldName + **" = ?"**,  
 **new** String[] { String.*valueOf*(id) });  
 db.close();  
}

* 1. Add a Delete button in TaskActivity
  2. Add the proper deleteTask event handler in TaskActivity

1. Test the application by deleting an existing task.

Demonstrate your solution.