

SQLite Database in Android

10:35

Java

20

8

Introduction to Java

Add Course

Read All Courses

10:37

q	
w	www
ww	
C++	
100	20
Introduction to C++	
Python	
40	4
Introduction to Python	

10:38

Php

30

3

Introduction to Php

Update Course

Delete Course

10:39

Php		
30	3	
Introduction to Php		
C++		
100	20	
Introduction to C++		
Python		
40	4	
Introduction to Python		

10:44

C++

20

100

Introduction to C++

Update Course

Delete Course

10:45

Php	
30	3
Introduction to Php	
Python	
40	4
Introduction to Python	

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:id="@+id/main"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     android:orientation="vertical"
9     tools:context=".MainActivity">
10
11     <EditText
12         android:id="@+id/idEdtCourseName"
13         android:layout_width="match_parent"
14         android:layout_height="wrap_content"
15         android:layout_margin="10dp"
16         android:hint="Enter course Name" />
17
18     <EditText
19         android:id="@+id/idEdtCourseDuration"
20         android:layout_width="match_parent"
21         android:layout_height="wrap_content"
22         android:layout_margin="10dp"
23         android:hint="Enter Course Duration" />
24
25     <EditText
26         android:id="@+id/idEdtCourseTracks"
27         android:layout_width="match_parent"
28         android:layout_height="wrap_content"
29         android:layout_margin="10dp"
30         android:hint="Enter Course Tracks" />
31
```

```
32 <EditText
33     android:id="@+id/idEdtCourseDescription"
34     android:layout_width="match_parent"
35     android:layout_height="wrap_content"
36     android:layout_margin="10dp"
37     android:hint="Enter Course Description" />
38
39 <Button
40     android:id="@+id/idBtnAddCourse"
41     android:layout_width="match_parent"
42     android:layout_height="wrap_content"
43     android:layout_margin="10dp"
44     android:text="Add Course"
45     android:textAllCaps="false" />
46
47 <Button
48     android:id="@+id/idBtnReadCourse"
49     android:layout_width="match_parent"
50     android:layout_height="wrap_content"
51     android:layout_margin="10dp"
52     android:text="Read All Courses"
53     android:textAllCaps="false" />
54
55
56 </LinearLayout>
57
```

```
1 package com.example.sqlite2;
2 //Creating a modal class for storing the data
  7 usages
3 public class CourseModal {
4
5     3 usages
    private String courseName;
    3 usages
6     private String courseDuration;
    3 usages
7     private String courseTracks;
    3 usages
8     private String courseDescription;
    2 usages
9     private int id;
10
11     //Getter: To retrieve the value of an object's attribute
    2 usages
12 > public String getCourseName() { return courseName; }
15
    2 usages
16 > public String getCourseDuration() { return courseDuration; }
19
    2 usages
20 > public String getCourseTracks() { return courseTracks; }
23
    2 usages
24 > public String getCourseDescription() { return courseDescription; }
27
28 > public int getId() { return id; }
```

```
32 //Setter: To set the value of an object's attribute
    no usages
33 > public void setCourseName(String courseName) { this.courseName = courseName; }
36
    no usages
37 > public void setCourseDuration(String courseDuration) { this.courseDuration = courseDuration; }
40
    no usages
41 > public void setCourseTracks(String courseTracks) { this.courseTracks = courseTracks; }
44
    no usages
45 public void setCourseDescription(String courseDescription) {
46     this.courseDescription = courseDescription;
47 }
48
49 > public void setId(int id) { this.id = id; }
52
53 //Constructor: To initialize the values at the time of object creation
    1 usage
54 public CourseModal(String courseName, String courseDuration, String courseTracks, String courseDescription) {
55     this.courseName = courseName;
56     this.courseDuration = courseDuration;
57     this.courseTracks = courseTracks;
58     this.courseDescription = courseDescription;
59 }
60 }
61 }//CourseModal
```

```
1 package com.example.sqlite2;
2
3 import android.content.ContentValues;
4 import android.content.Context;
5 import android.database.Cursor;
6 import android.database.sqlite.SQLiteDatabase;
7 import android.database.sqlite.SQLiteOpenHelper;
8 import java.util.ArrayList;
9
10 public class DBHandler extends SQLiteOpenHelper {
11
12     // creating a constant variables for our database.
13     private static final String DB_NAME = "coursedb";
14     private static final int DB_VERSION = 1;
15     private static final String TABLE_NAME = "mycourses";
16     private static final String ID_COL = "id";
17     private static final String NAME_COL = "name";
18     private static final String DURATION_COL = "duration";
19     private static final String DESCRIPTION_COL = "description";
20     private static final String TRACKS_COL = "tracks";
```



```
22      // Constructor for the database handler.  
    3 usages  
23      public DBHandler(Context context) {  
24  
25          super(context, DB_NAME, factory: null, DB_VERSION);  
26      }//DBHandler  
27  
28      // .....Creating a database.....  
29      @Override  
30      public void onCreate(SQLiteDatabase db) {  
31          String query = "CREATE TABLE " + TABLE_NAME + " (" +  
32              + ID_COL + " INTEGER PRIMARY KEY AUTOINCREMENT, "  
33              + NAME_COL + " TEXT, "  
34              + DURATION_COL + " TEXT, "  
35              + DESCRIPTION_COL + " TEXT, "  
36              + TRACKS_COL + " TEXT)";  
37  
38          db.execSQL(query);  
39      }//onCreate
```

```
40
41 // .....Adding a new course.....
1 usage
42 public void addNewCourse(String courseName, String courseDuration, String courseDescription, String courseTracks) {
43
44     SQLiteDatabase db = this.getWritableDatabase();
45
46     // creating a variable for content values.
47     ContentValues values = new ContentValues();
48
49     // passing all values along with its key and value pair.
50     values.put(NAME_COL, courseName);
51     values.put(DURATION_COL, courseDuration);
52     values.put(DESCRIPTION_COL, courseDescription);
53     values.put(TRACKS_COL, courseTracks);
54
55     // passing content values to the table.
56     db.insert(TABLE_NAME, nullColumnHack: null, values);
57
58     // at last we are closing our database after adding database.
59     db.close();
60 }//addNewCourse
```

```
62 // .....Reading all the courses.....
1 usage
63 public ArrayList<CourseModal> readCourses() {
64     // creating a database for reading the database.
65     SQLiteDatabase db = this.getReadableDatabase();
66
67     // creating a cursor with query to read data from database.
68     Cursor cursorCourses = db.rawQuery( sql: "SELECT * FROM " + TABLE_NAME, selectionArgs: null);
69
70     // creating a new array list.
71     ArrayList<CourseModal> courseModalArrayList = new ArrayList<>();
72
73     // moving the cursor to first position.
74     if (cursorCourses.moveToFirst()) {
75         do {
76             // adding the data from cursor to the array list.
77             courseModalArrayList.add(new CourseModal(cursorCourses.getString( columnIndex: 1),
78                 cursorCourses.getString( columnIndex: 4),
79                 cursorCourses.getString( columnIndex: 2),
80                 cursorCourses.getString( columnIndex: 3)));
81         } while (cursorCourses.moveToNext());
82         // moving our cursor to next.
83     } //if
84     // at last closing our cursor and returning our array list.
85     cursorCourses.close();
86     return courseModalArrayList;
87 } //ArrayList
```

```
89
90      // .....Updating the course.....
1 usage
91      public void updateCourse(String originalCourseName, String courseName, String courseDescription,
92                               String courseTracks, String courseDuration) {
93
94          // calling a method to get writable database.
95          SQLiteDatabase db = this.getWritableDatabase();
96          ContentValues values = new ContentValues();
97
98          // passing all values along with its key and value pair.
99          values.put(NAME_COL, courseName);
100         values.put(DURATION_COL, courseDuration);
101         values.put(DESCRIPTION_COL, courseDescription);
102         values.put(TRACKS_COL, courseTracks);
103
104         // calling a update method to update our database and passing our values.
105         // and we are comparing it with name of our course which is stored in original name variable.
106         db.update(TABLE_NAME, values, whereClause: "name=?", new String[]{originalCourseName});
107         db.close();
108     }
```

```

110 // .....Deleting the course.....
111 1 usage
112 public void deleteCourse(String courseName) {
113     // Creating a variable to write our database.
114     SQLiteDatabase db = this.getWritableDatabase();
115
116     // Calling a method to delete our course and we are comparing it with our course name.
117     db.delete(TABLE_NAME, whereClause: "name=?", new String[]{courseName});
118     db.close();
119 }
120
121 10 usages
122 @Override
123 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
124     // this method is called to check if the table exists already.
125     db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
126     onCreate(db);
127 } //onUpgrade
128 } //DBHandler

```

© MainActivity.java ×

```
3      import android.os.Bundle;
4      import android.content.Intent;
5      import android.view.View;
6      import android.widget.Button;
7      import android.widget.EditText;
8      import android.widget.Toast;
9      import androidx.appcompat.app.AppCompatActivity;
10
11      //Creating an Activity for displaying the list of courses
12      public class MainActivity extends AppCompatActivity {
13
14          3 usages
15          private EditText courseNameEdt, courseTracksEdt, courseDurationEdt, courseDescriptionEdt;
16          2 usages
17          private Button addCourseBtn, readCourseBtn, updateCourseBtn, deleteCourseBtn;
18          2 usages
19          private DBHelper dbHelper;
20          no usages
21          String courseName, courseDesc, courseDuration, courseTracks;
```

```
20      @Override
21      protected void onCreate(Bundle savedInstanceState) {
22          super.onCreate(savedInstanceState);
23          setContentView(R.layout.activity_main);
24
25          courseNameEdt = findViewById(R.id.idEdtCourseName);
26          courseTracksEdt = findViewById(R.id.idEdtCourseTracks);
27          courseDurationEdt = findViewById(R.id.idEdtCourseDuration);
28          courseDescriptionEdt = findViewById(R.id.idEdtCourseDescription);
29          addCourseBtn = findViewById(R.id.idBtnAddCourse);
30          readCourseBtn = findViewById(R.id.idBtnReadCourse);
31
32          // creating a new dbhandler class and passing the context to it.
33          dbHandler = new DBHandler( context: MainActivity.this);
34
35          addCourseBtn.setOnClickListener(new View.OnClickListener() {
36              @Override
37              public void onClick(View v) {
38
39                  String courseName = courseNameEdt.getText().toString();
40                  String courseTracks = courseTracksEdt.getText().toString();
41                  String courseDuration = courseDurationEdt.getText().toString();
42                  String courseDescription = courseDescriptionEdt.getText().toString();
43
44                  // validating if the text fields are empty or not.
45                  if (courseName.isEmpty() || courseTracks.isEmpty() || courseDuration.isEmpty() || courseDescription.isEmpty()) {
46                      Toast.makeText( context: MainActivity.this, text: "Please enter all the data..", Toast.LENGTH_SHORT).show();
47                      return;
48                  }
49              }
50          });
51      }
```

```

        // calling a method to add new course to sqLite data and pass all our values to it.
        dbHelper.addNewCourse(courseName, courseDuration, courseDescription, courseTracks);

        // after adding the data, a toast message is displayed.
        Toast.makeText(context: MainActivity.this, text: "Course has been added.", Toast.LENGTH_SHORT).show();
        courseNameEdt.setText("");
        courseDurationEdt.setText("");
        courseTracksEdt.setText("");
        courseDescriptionEdt.setText("");
    }////onClick
}); //setOnClickListener

readCourseBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // opening a new activity via a intent.
        Intent i = new Intent(packageContext: MainActivity.this, ViewCourses.class);
        startActivity(i);
    } //onClick
}); //setOnClickListener

} //onCreate
} //MainActivity

```


</> course_rv_item.xml x

```
1 <?xml version="1.0" encoding="utf-8"?>
2
3 <!--> Creating a new layout file for the item of RecyclerView <!-->
4 <androidx.cardview.widget.CardView
5     xmlns:android="http://schemas.android.com/apk/res/android"
6     xmlns:app="http://schemas.android.com/apk/res-auto"
7     android:layout_width="match_parent"
8     android:layout_height="wrap_content"
9     android:layout_margin="5dp"
10    android:elevation="8dp"
11    app:cardCornerRadius="4dp">
12
13    <LinearLayout
14        android:layout_width="match_parent"
15        android:layout_height="wrap_content"
16        android:layout_margin="3dp"
17        android:orientation="vertical">
18
19        <TextView
20            android:id="@+id/idTVCourseName"
21            android:layout_width="match_parent"
22            android:layout_height="wrap_content"
23            android:padding="3dp"
24            android:text="Course Name"
25            android:textColor="@color/black" />
```

Palette

course_rv_item.xml v



Course Name	
Course Tracks	Duration
Description	

</> course_rv_item.xml x

```
27 <LinearLayout
28     android:layout_width="match_parent"
29     android:layout_height="wrap_content"
30     android:orientation="horizontal"
31     android:weightSum="2">
32
33     <TextView
34         android:id="@+id/idTVCourseTracks"
35         android:layout_width="0dp"
36         android:layout_height="wrap_content"
37         android:layout_weight="1"
38         android:padding="3dp"
39         android:text="Course Tracks"
40         android:textColor="@color/black" />
41
42     <TextView
43         android:id="@+id/idTVCourseDuration"
44         android:layout_width="0dp"
45         android:layout_height="wrap_content"
46         android:layout_weight="1"
47         android:padding="3dp"
48         android:text="Duration"
49         android:textColor="@color/black" />
50
51 </LinearLayout>
```

```
53 <TextView
54     android:id="@+id/idTVCourseDescription"
55     android:layout_width="match_parent"
56     android:layout_height="wrap_content"
57     android:layout_weight="1"
58     android:padding="3dp"
59     android:text="Description"
60     android:textColor="@color/black" />
61
62 </LinearLayout>
63
64 </androidx.cardview.widget.CardView>
```

```
3  import android.content.Context;
4  import android.content.Intent;
5  import android.view.LayoutInflater;
6  import android.view.View;
7  import android.view.ViewGroup;
8  import android.widget.TextView;
9  import androidx.annotation.NonNull;
10 import androidx.recyclerview.widget.RecyclerView;
11 import java.util.ArrayList;
12
13 3 usages
14 public class CourseRVAdapter extends RecyclerView.Adapter<CourseRVAdapter.ViewHolder> {
15
16 3 usages
17 private ArrayList<CourseModal> courseModalArrayList;
18 3 usages
19 private Context context;
20
21 // constructor
22 1 usage
23 public CourseRVAdapter(ArrayList<CourseModal> courseModalArrayList, Context context) {
24     this.courseModalArrayList = courseModalArrayList;
25     this.context = context;
26 }
27
28 @NonNull
29 @Override
30 public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
31     // Inflating the layout file for the recycler view items.
32     View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.course_rv_item, parent, attachToRoot: false);
33     return new ViewHolder(view);
34 }
```

```
32      @Override
33      public void onBindViewHolder(@NonNull ViewHolder holder, int position) {
34          // Setting data to the views of recycler view item.
35          CourseModal modal = courseModalArrayList.get(position);
36          holder.courseNameTV.setText(modal.getCourseName());
37          holder.courseDescTV.setText(modal.getCourseDescription());
38          holder.courseDurationTV.setText(modal.getCourseDuration());
39          holder.courseTracksTV.setText(modal.getCourseTracks());
40
41          // to add on click listener for the recycler view item.
42          holder.itemView.setOnClickListener(new View.OnClickListener() {
43              @Override
44              public void onClick(View v) {
45
46                  // Calling an intent.
47                  Intent i = new Intent(context, UpdateCourseActivity.class);
48
49                  // Passing all our values.
50                  i.putExtra(name: "name", modal.getCourseName());
51                  i.putExtra(name: "description", modal.getCourseDescription());
52                  i.putExtra(name: "duration", modal.getCourseDuration());
53                  i.putExtra(name: "tracks", modal.getCourseTracks());
54
55                  context.startActivity(i);
56              }
57          });
58      }
```

```

60      @Override
61      public int getItemCount() {
62          // returning the size of our array list
63          return courseModalArrayList.size();
64      }
65
66      4 usages
67      public class ViewHolder extends RecyclerView.ViewHolder {
68          // creating variables for our text views.
69          2 usages
70          private TextView courseNameTV, courseDescTV, courseDurationTV, courseTracksTV;
71
72          1 usage
73          public ViewHolder(@NonNull View itemView) {
74              super(itemView);
75              // initializing our text views
76              courseNameTV = itemView.findViewById(R.id.idTVCourseName);
77              courseDescTV = itemView.findViewById(R.id.idTVCourseDescription);
78              courseDurationTV = itemView.findViewById(R.id.idTVCourseDuration);
79              courseTracksTV = itemView.findViewById(R.id.idTVCourseTracks);
80          }
81      } //ViewHolder
82  } //CourseRVAdapter

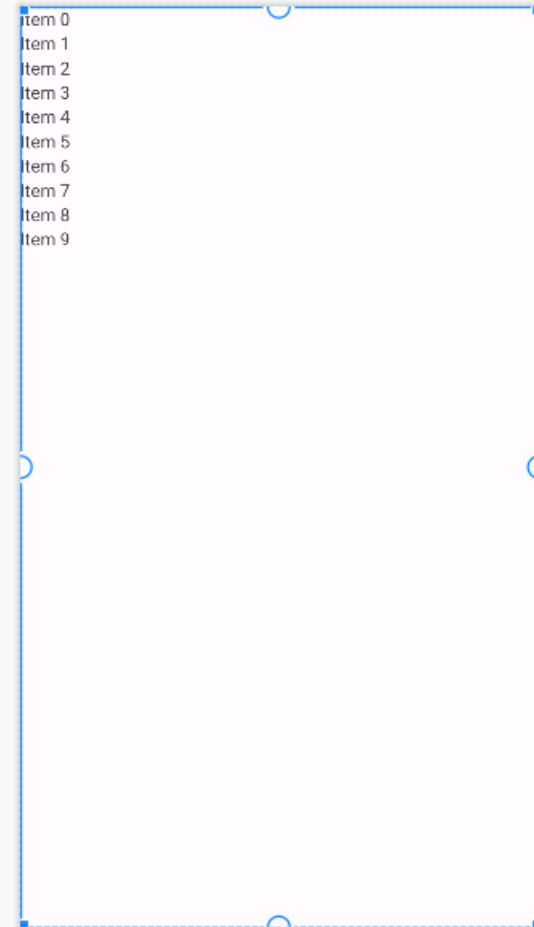
```

</> activity_view_courses.xml x

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".ViewCourses">
8
9     <!--recycler view for displaying our courses-->
10    <androidx.recyclerview.widget.RecyclerView
11        android:id="@+id/idRVCourses"
12        android:layout_width="match_parent"
13        android:layout_height="match_parent" />
14
15 </RelativeLayout>
16
```

Palette

activity_view_courses.xml v



Component Tree

```
3      import android.os.Bundle;
4      import androidx.appcompat.app.AppCompatActivity;
5      import androidx.recyclerview.widget.LinearLayoutManager;
6      import androidx.recyclerview.widget.RecyclerView;
7      import java.util.ArrayList;
8
9      public class ViewCourses extends AppCompatActivity {
10
11          // creating variables for the array list, dbhandler, adapter and recycler view.
12          // 3 usages
13          private ArrayList<CourseModal> courseModalArrayList;
14          // 2 usages
15          private DBHandler dbHandler;
16          // 2 usages
17          private CourseRVAdapter courseRVAdapter;
18          // 3 usages
19          private RecyclerView coursesRV;
20
21          @Override
22          protected void onCreate(Bundle savedInstanceState) {
23              super.onCreate(savedInstanceState);
24              setContentView(R.layout.activity_view_courses);
25
26              // initializing all variables.
27              courseModalArrayList = new ArrayList<>();
28              dbHandler = new DBHandler(context, ViewCourses.this);
29
30              // getting our course array list from dbhandler class.
31              courseModalArrayList = dbHandler.readCourses();
```

```
28
29      // passing the array list to the adapter class.
30      courseRVAdapter = new CourseRVAdapter(courseModalArrayList, context: ViewCourses.this);
31      coursesRV = findViewById(R.id.idRVCourses);
32
33      // setting layout manager for the recycler view.
34      LinearLayoutManager layoutManager = new LinearLayoutManager(context: ViewCourses.this, RecyclerView.VERTICAL, reverseLayout: false);
35      coursesRV.setLayoutManager(layoutManager);
36
37      // setting the adapter to recycler view.
38      coursesRV.setAdapter(courseRVAdapter);
39  }//onCreate
40 }//ViewCourses
```


</> activity_update_course.xml x

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:orientation="vertical"
8     tools:context=".UpdateCourseActivity">
9
10    <!--Edit text to enter course name-->
11    <EditText
12        android:id="@+id/idEdtCourseName"
13        android:layout_width="match_parent"
14        android:layout_height="wrap_content"
15        android:layout_margin="10dp"
16        android:hint="Enter course Name" />
17
18    <!--edit text to enter course duration-->
19    <EditText
20        android:id="@+id/idEdtCourseDuration"
21        android:layout_width="match_parent"
22        android:layout_height="wrap_content"
23        android:layout_margin="10dp"
24        android:hint="Enter Course Duration" />
```

Palette

activity_update_course.xml



Enter course Name

Enter Course Duration

Enter Course Tracks

Enter Course Description

Update Course

Delete Course

</> activity_update_course.xml x

```
25
26      <!--edit text to display course tracks-->
27      <EditText
28          android:id="@+id/idEdtCourseTracks"
29          android:layout_width="match_parent"
30          android:layout_height="wrap_content"
31          android:layout_margin="10dp"
32          android:hint="Enter Course Tracks" />
33
34      <!--edit text for course description-->
35      <EditText
36          android:id="@+id/idEdtCourseDescription"
37          android:layout_width="match_parent"
38          android:layout_height="wrap_content"
39          android:layout_margin="10dp"
40          android:hint="Enter Course Description" />
41
42      <!--button for updating our course-->
43      <Button
44          android:id="@+id/idBtnUpdateCourse"
45          android:layout_width="match_parent"
46          android:layout_height="wrap_content"
47          android:layout_margin="10dp"
48          android:text="Update Course"
49          android:textAllCaps="false" />
```

```
51      <!--button for deleting our course-->
52      <Button
53          android:id="@+id/idBtnDelete"
54          android:layout_width="match_parent"
55          android:layout_height="wrap_content"
56          android:layout_margin="10dp"
57          android:text="Delete Course"
58          android:textAllCaps="false" />
59
60  </LinearLayout>
```

```
3      import android.content.Intent;
4      import android.os.Bundle;
5      import android.view.View;
6      import android.widget.Button;
7      import android.widget.EditText;
8      import android.widget.Toast;
9      import androidx.appcompat.app.AppCompatActivity;
10
11  public class UpdateCourseActivity extends AppCompatActivity {
12
13      3 usages
14      private EditText courseNameEdt, courseTracksEdt, courseDurationEdt, courseDescriptionEdt;
15      2 usages
16      private Button updateCourseBtn, deleteCourseBtn;
17      3 usages
18      private DBHandler dbHandler;
19      4 usages
20      String courseName, courseDesc, courseDuration, courseTracks;
21
22      @Override
23      protected void onCreate(Bundle savedInstanceState) {
24          super.onCreate(savedInstanceState);
25          setContentView(R.layout.activity_update_course);
26
27          courseNameEdt = findViewById(R.id.idEdtCourseName);
28          courseTracksEdt = findViewById(R.id.idEdtCourseTracks);
29          courseDurationEdt = findViewById(R.id.idEdtCourseDuration);
30          courseDescriptionEdt = findViewById(R.id.idEdtCourseDescription);
31          updateCourseBtn = findViewById(R.id.idBtnUpdateCourse);
32          deleteCourseBtn = findViewById(R.id.idBtnDelete);
```

```
29
30 // Initializing our dbhandler class.
31 dbHandler = new DBHandler( context: UpdateCourseActivity.this);
32
33 // Getting data
34 courseName = getIntent().getStringExtra( name: "name");
35 courseDesc = getIntent().getStringExtra( name: "description");
36 courseDuration = getIntent().getStringExtra( name: "duration");
37 courseTracks = getIntent().getStringExtra( name: "tracks");
38
39 // setting data
40 courseNameEdt.setText(courseName);
41 courseDescriptionEdt.setText(courseDesc);
42 courseTracksEdt.setText(courseTracks);
43 courseDurationEdt.setText(courseDuration);
44
45 // adding on click listener to our update course button.
46 updateCourseBtn.setOnClickListener(new View.OnClickListener() {
47     @Override
48     public void onClick(View v) {
49
50         // Calling an update course method and passing all the edit text values.
51         dbHandler.updateCourse(courseName, courseNameEdt.getText().toString(), courseDescriptionEdt.getText().toString(),
52             courseTracksEdt.getText().toString(), courseDurationEdt.getText().toString());
53
54         Toast.makeText( context: UpdateCourseActivity.this, text: "Course Updated..", Toast.LENGTH_SHORT).show();
55
56         // launching our main activity.
57         Intent i = new Intent( packageContext: UpdateCourseActivity.this, MainActivity.class);
58         startActivity(i);
59     } //onClick
60 }); //setOnClickListener
```

```
62      // adding on click listener for delete button to delete our course.
63      deleteCourseBtn.setOnClickListener(new View.OnClickListener() {
64          @Override
65          public void onClick(View v) {
66              // calling a method to delete the course.
67              dbHelper.deleteCourse(courseName);
68              Toast.makeText(context: UpdateCourseActivity.this, text: "Deleted the course", Toast.LENGTH_SHORT).show();
69              Intent i = new Intent(packageContext: UpdateCourseActivity.this, MainActivity.class);
70              startActivity(i);
71          } //onClick
72      }); //setOnClickListener
73  } //onCreate
74 } //UpdateCourseActivity
```