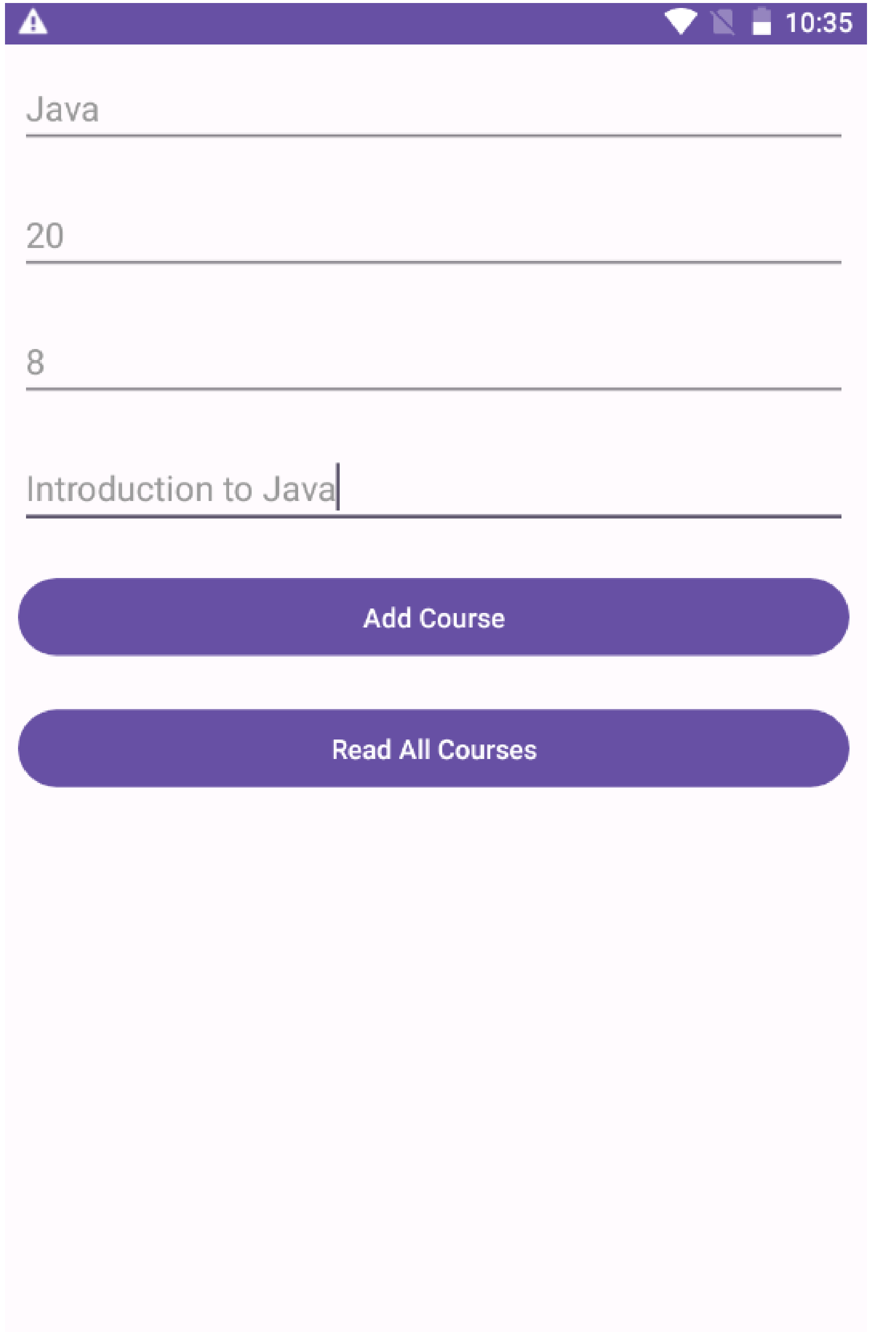
SQLite Database in Android

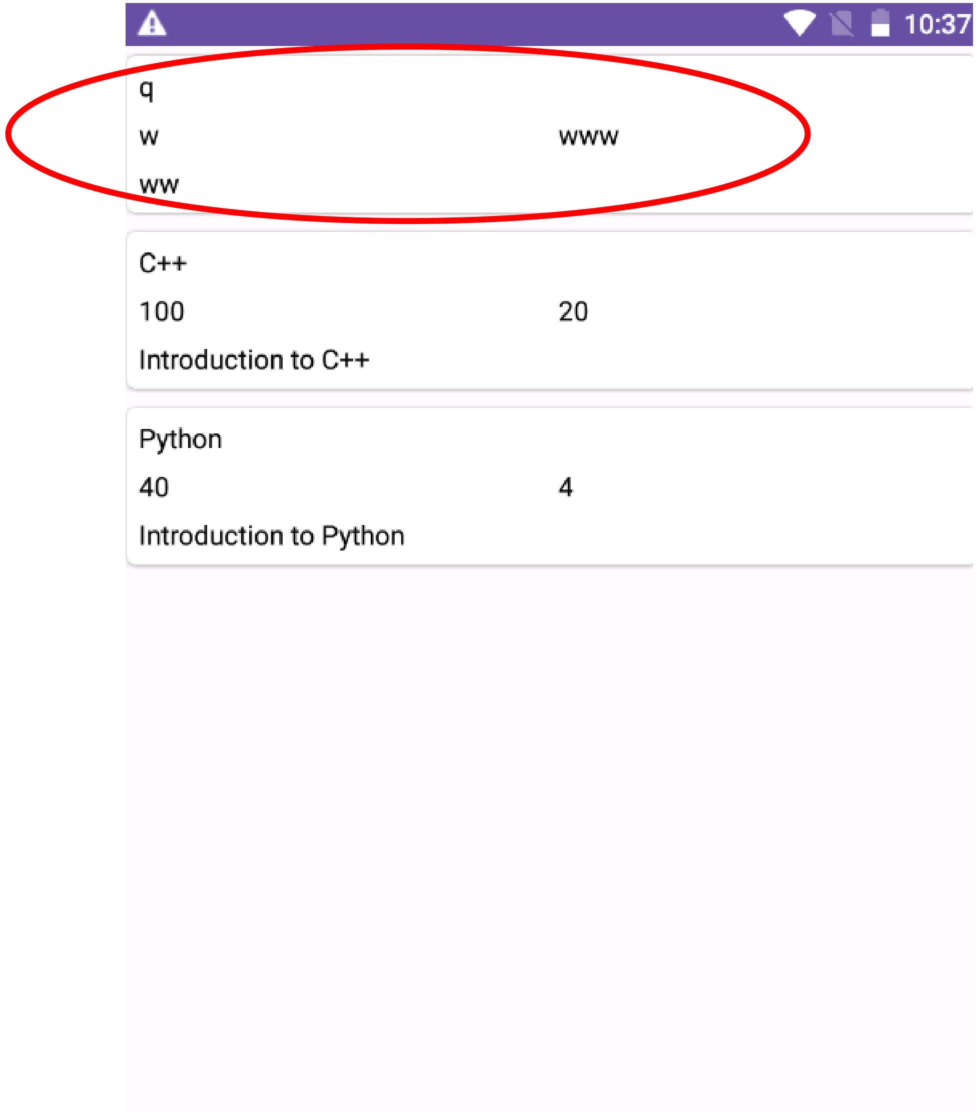


Java

20

8

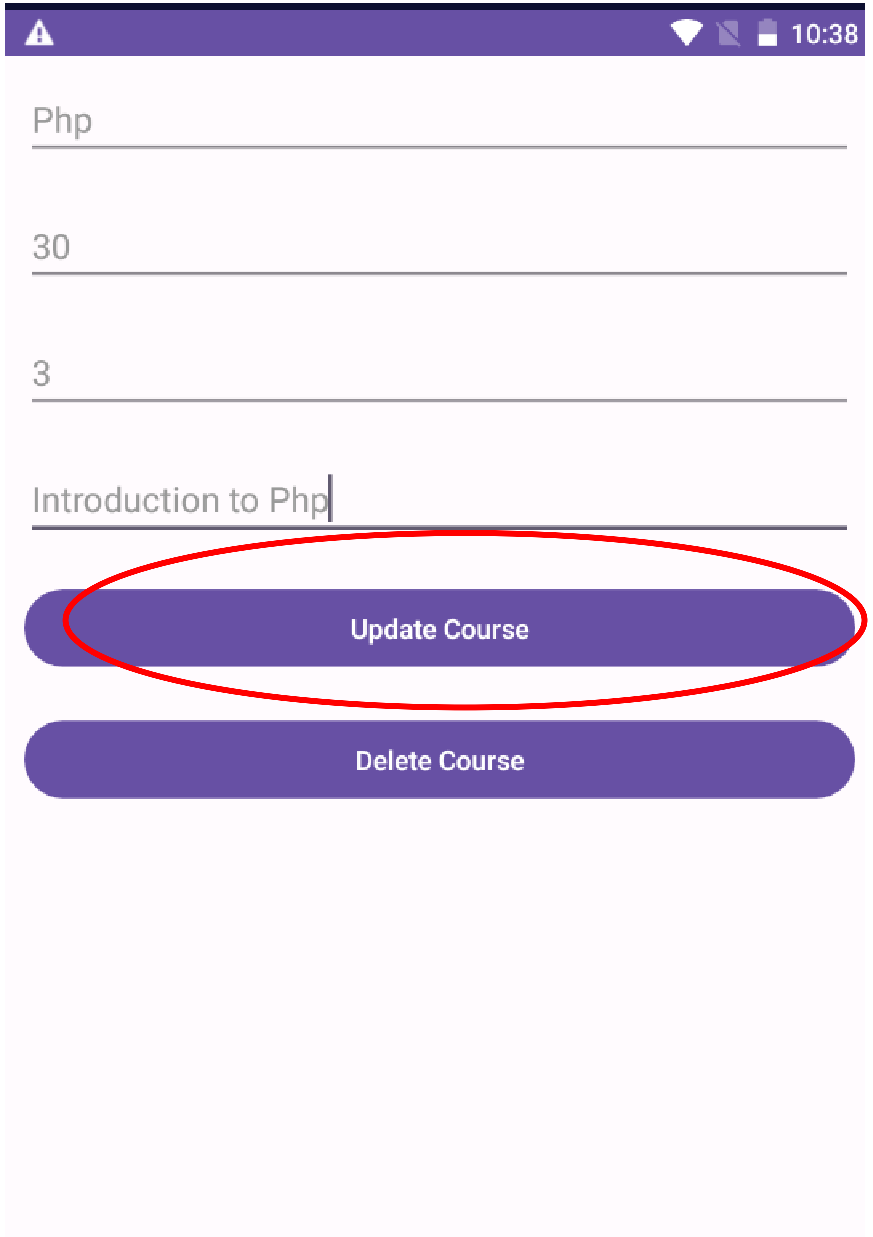
Introduction to Java



q
w www
ww

C++
100 20
Introduction to C++

Python
40 4
Introduction to Python

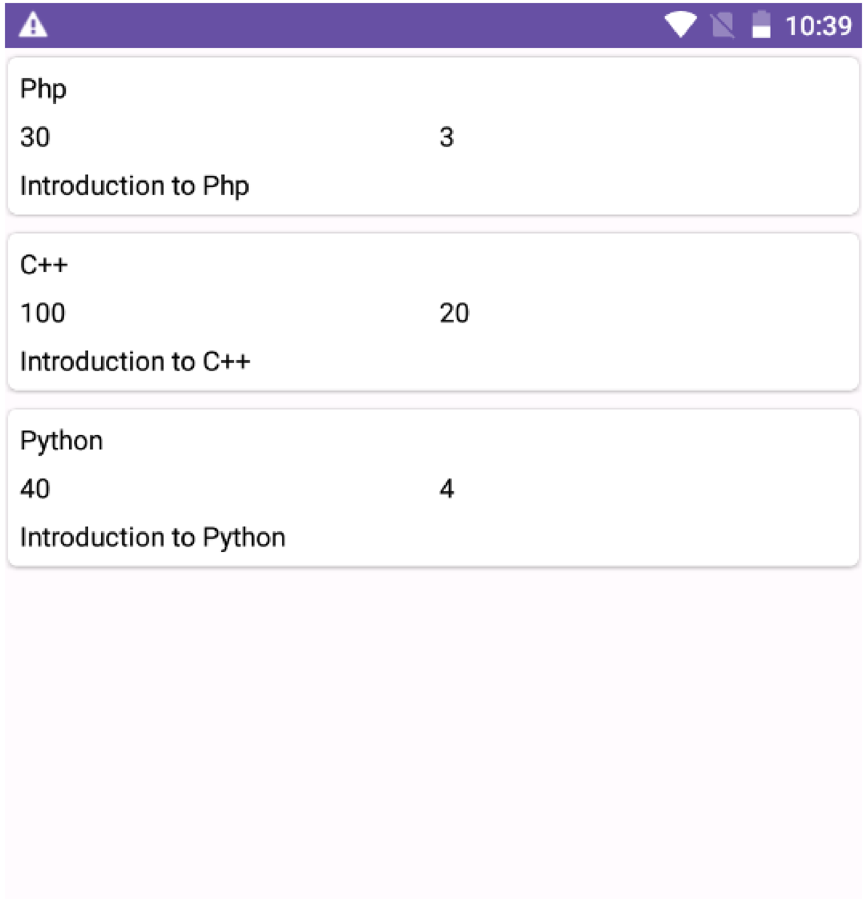


Php

30

3

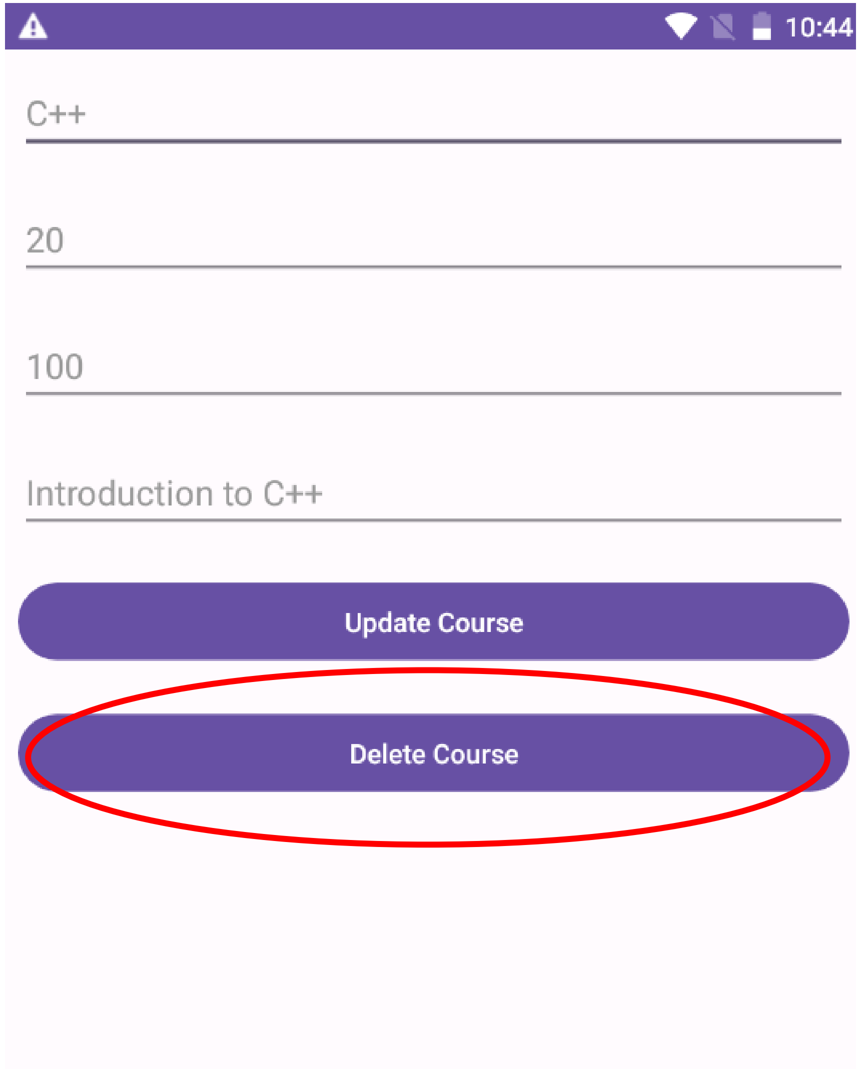
Introduction to Php



Php
30 3
Introduction to Php

C++
100 20
Introduction to C++

Python
40 4
Introduction to Python

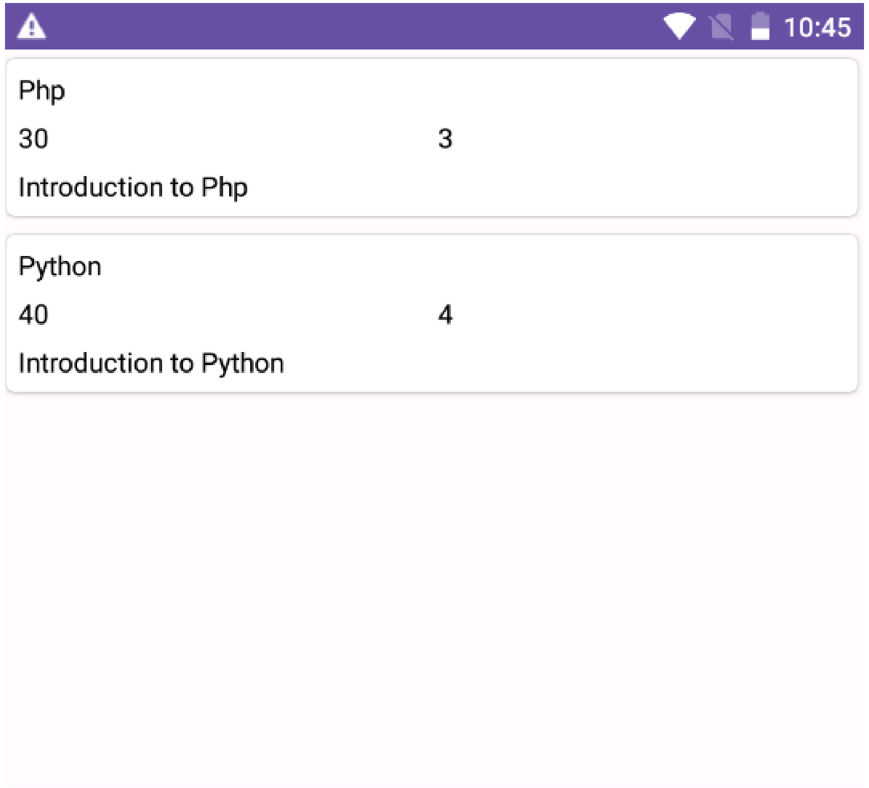


C++

20

100

Introduction to C++



Php

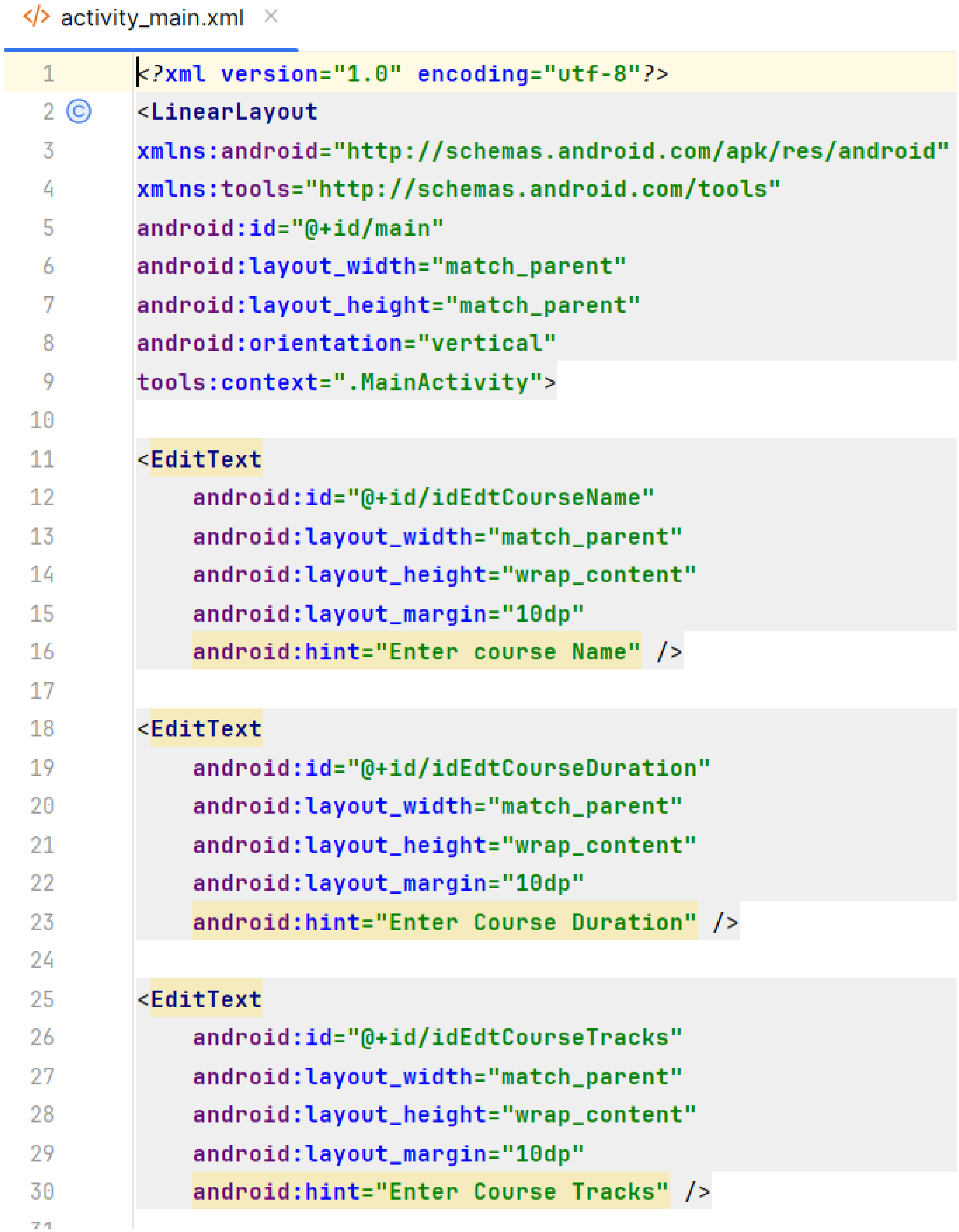
30 3

Introduction to Php

Python

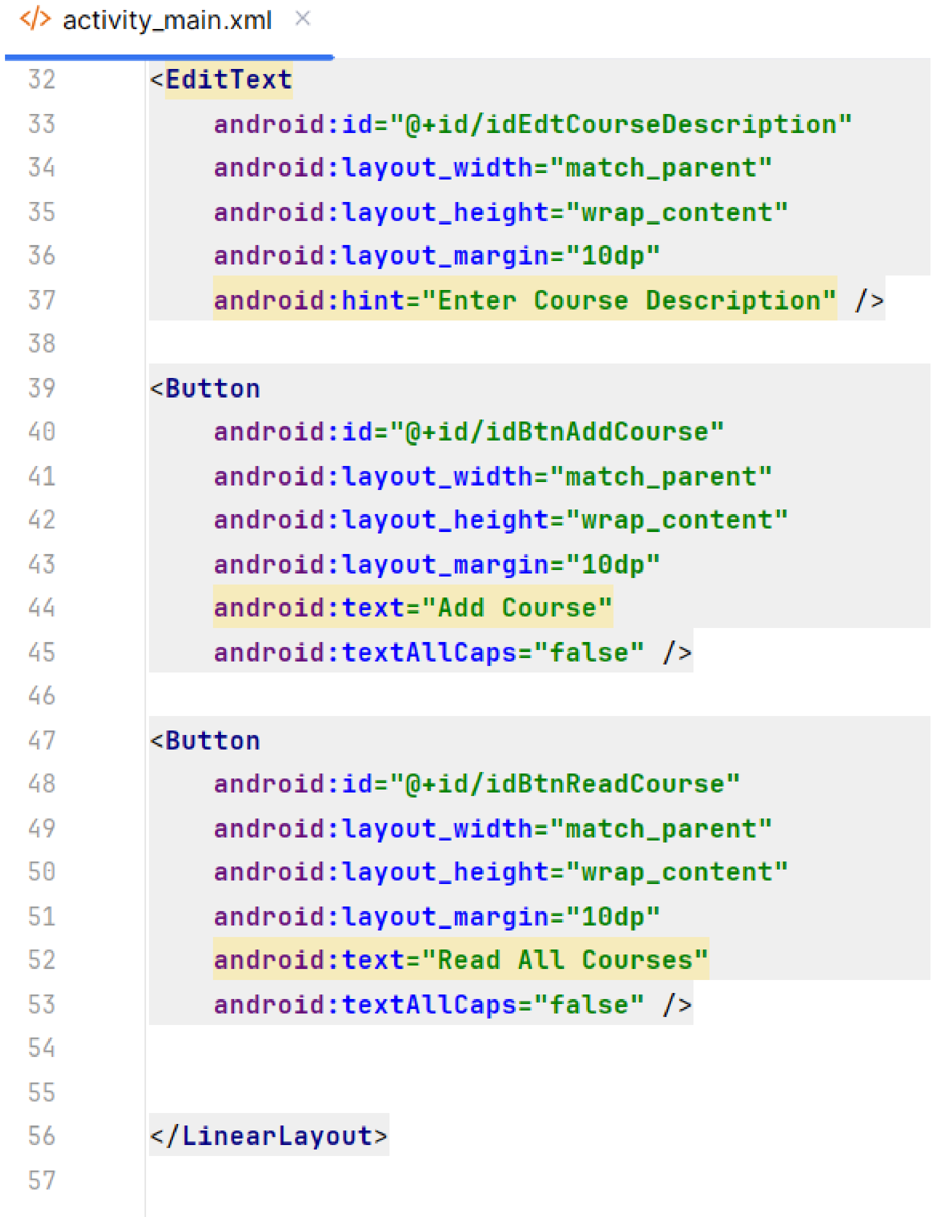
40 4

Introduction to Python



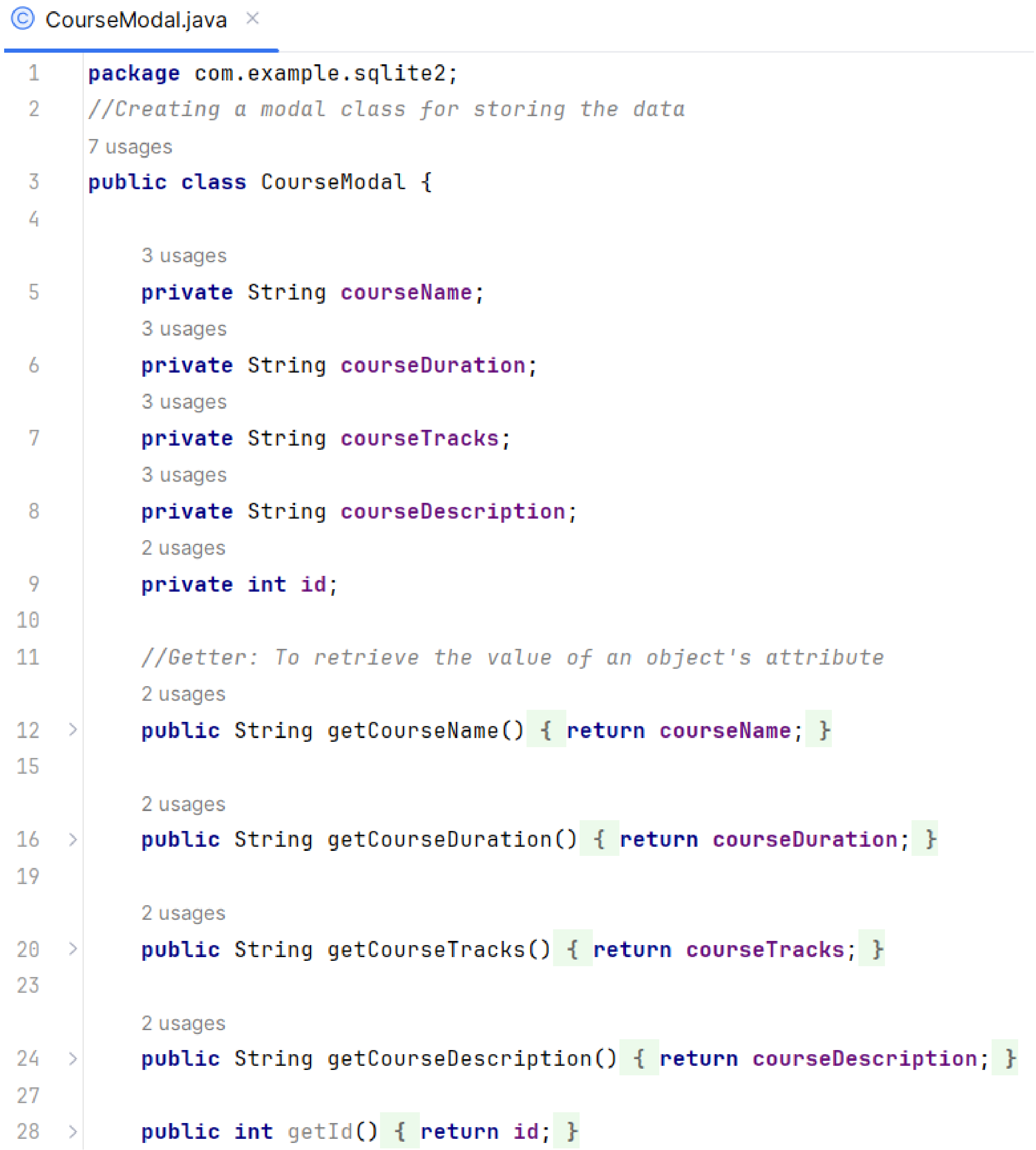
activity_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/main"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
tools:context=".MainActivity">

<EditText
    android:id="@+id/idEdtCourseName"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:hint="Enter course Name" />

<EditText
    android:id="@+id/idEdtCourseDuration"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:hint="Enter Course Duration" />

<EditText
    android:id="@+id/idEdtCourseTracks"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:hint="Enter Course Tracks" />
```



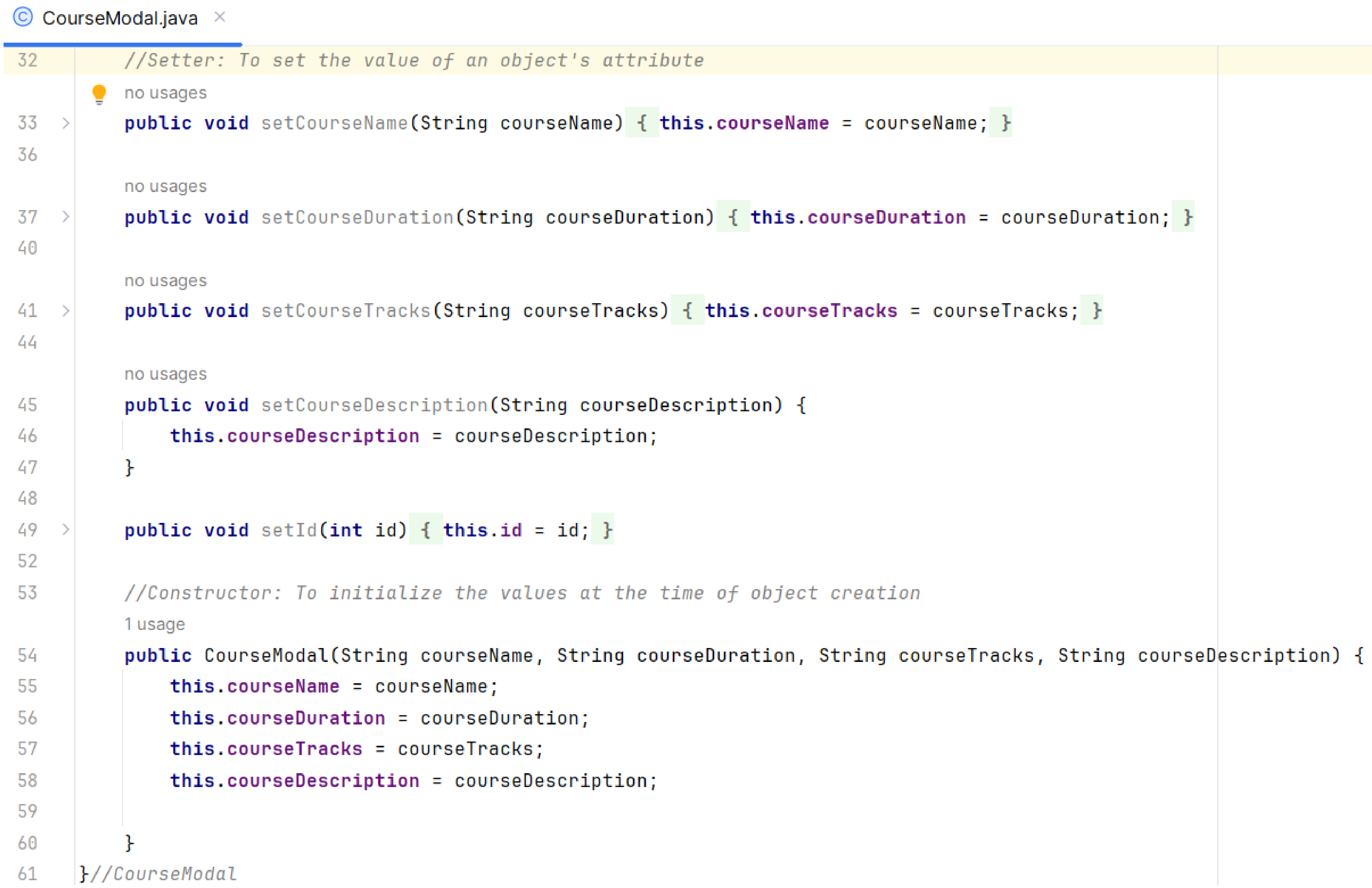
activity_main.xml

```xml
<EditText
    android:id="@+id/idEdtCourseDescription"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:hint="Enter Course Description" />

<Button
    android:id="@+id/idBtnAddCourse"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:text="Add Course"
    android:textAllCaps="false" />

<Button
    android:id="@+id/idBtnReadCourse"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:text="Read All Courses"
    android:textAllCaps="false" />


</LinearLayout>
```

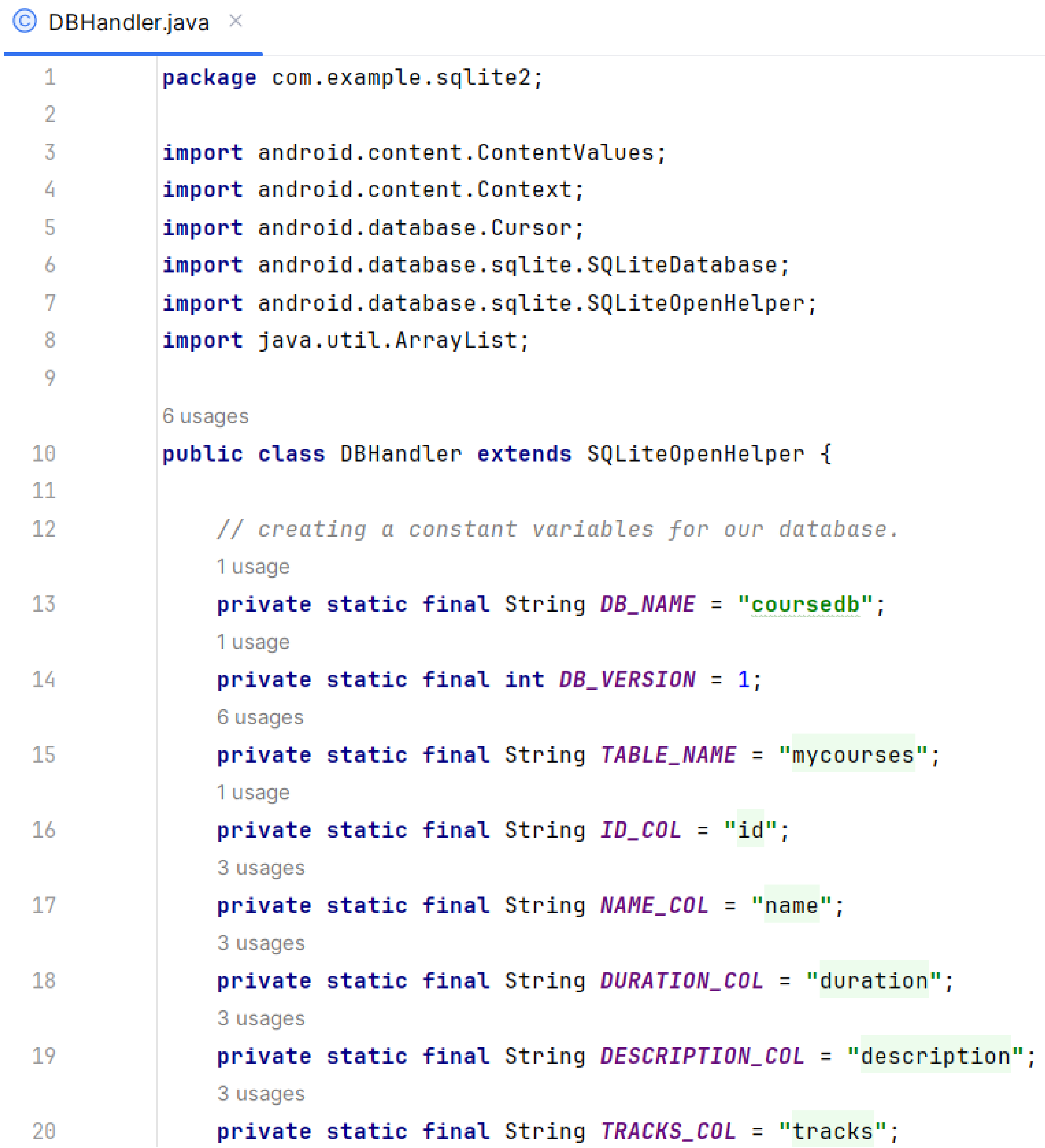


CourseModal.java

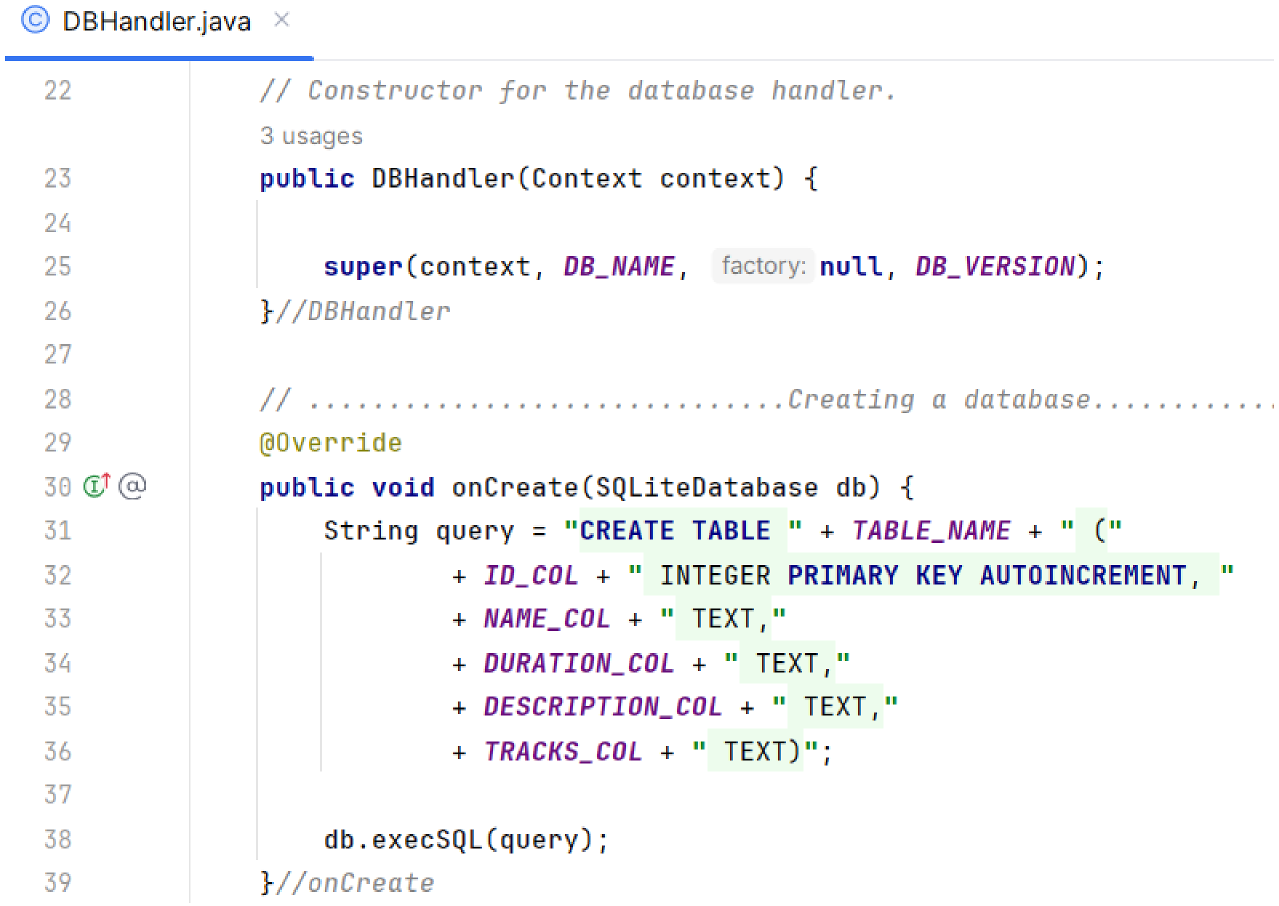
```java
package com.example.sqlite2;
//Creating a modal class for storing the data
7 usages
public class CourseModal {

    3 usages
    private String courseName;
    3 usages
    private String courseDuration;
    3 usages
    private String courseTracks;
    3 usages
    private String courseDescription;
    2 usages
    private int id;

    //Getter: To retrieve the value of an object's attribute
    2 usages
    public String getCourseName() { return courseName; }

    2 usages
    public String getCourseDuration() { return courseDuration; }

    2 usages
    public String getCourseTracks() { return courseTracks; }

    2 usages
    public String getCourseDescription() { return courseDescription; }

    public int getId() { return id; }
```



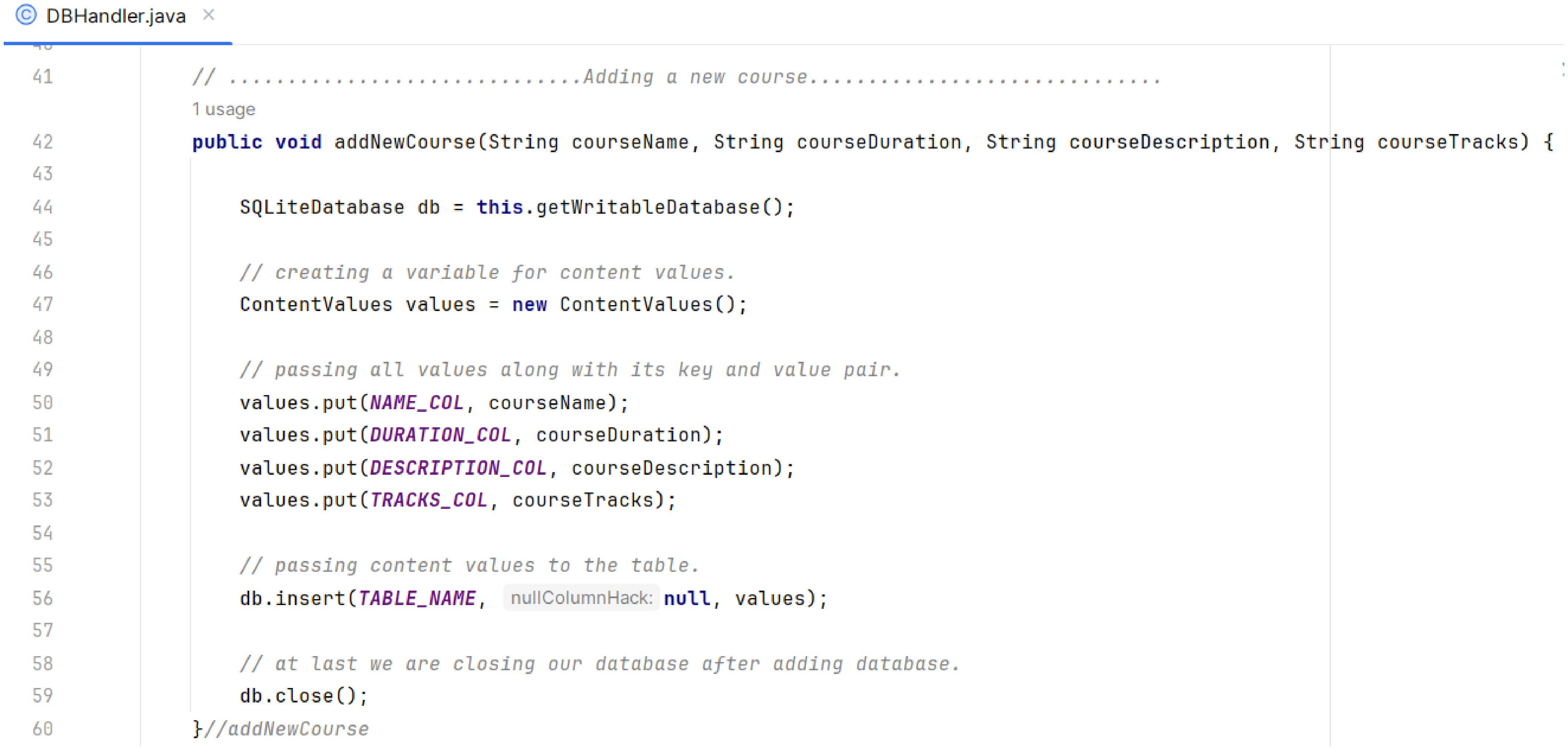
```java
    //Setter: To set the value of an object's attribute
    public void setCourseName(String courseName) { this.courseName = courseName; }

    public void setCourseDuration(String courseDuration) { this.courseDuration = courseDuration; }

    public void setCourseTracks(String courseTracks) { this.courseTracks = courseTracks; }

    public void setCourseDescription(String courseDescription) {
        this.courseDescription = courseDescription;
    }

    public void setId(int id) { this.id = id; }

    //Constructor: To initialize the values at the time of object creation
    public CourseModal(String courseName, String courseDuration, String courseTracks, String courseDescription) {
        this.courseName = courseName;
        this.courseDuration = courseDuration;
        this.courseTracks = courseTracks;
        this.courseDescription = courseDescription;

    }
}//CourseModal
```



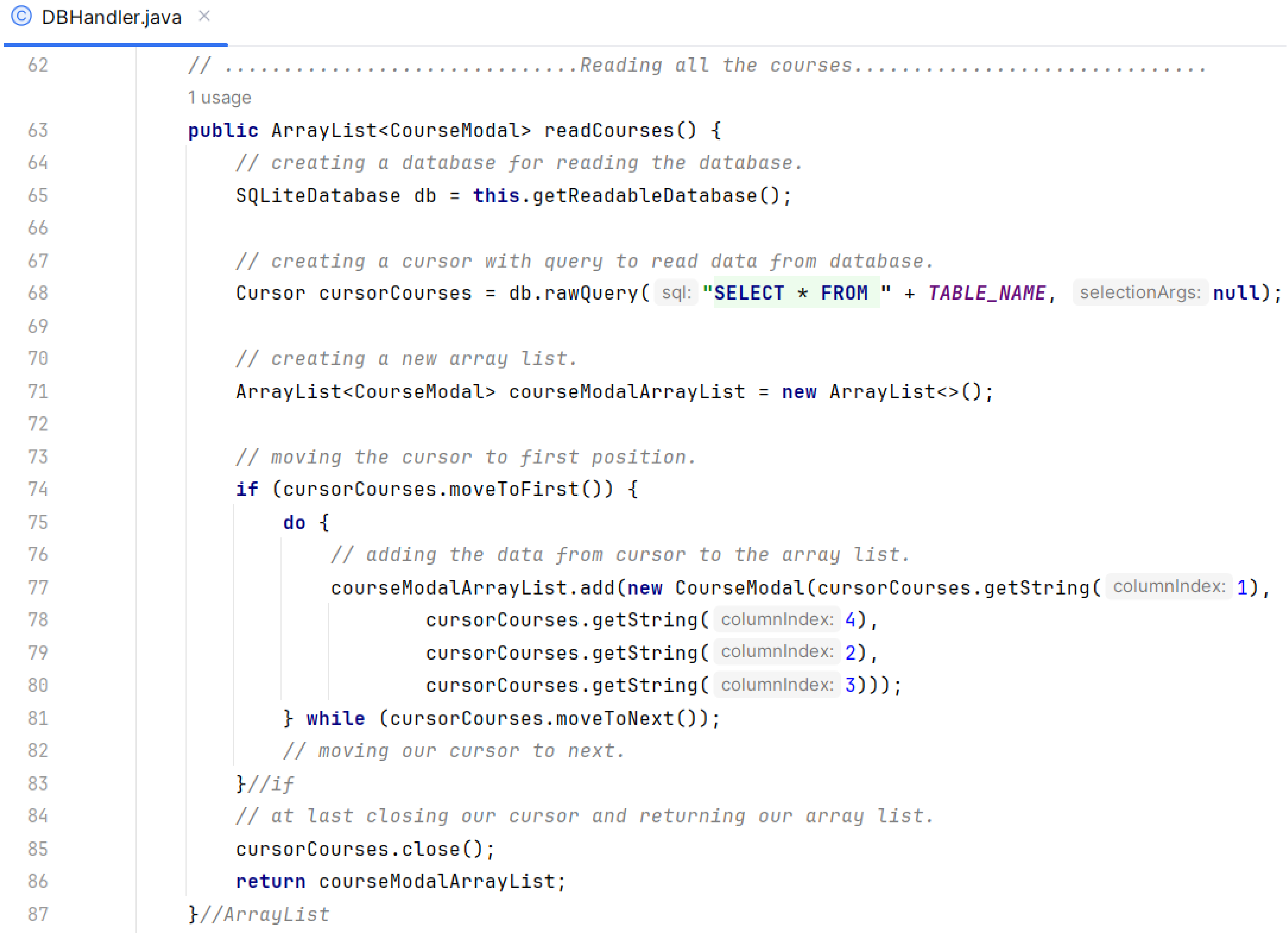
```
package com.example.sqlite2;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import java.util.ArrayList;

6 usages
public class DBHandler extends SQLiteOpenHelper {

    // creating a constant variables for our database.
    1 usage
    private static final String DB_NAME = "coursedb";
    1 usage
    private static final int DB_VERSION = 1;
    6 usages
    private static final String TABLE_NAME = "mycourses";
    1 usage
    private static final String ID_COL = "id";
    3 usages
    private static final String NAME_COL = "name";
    3 usages
    private static final String DURATION_COL = "duration";
    3 usages
    private static final String DESCRIPTION_COL = "description";
    3 usages
    private static final String TRACKS_COL = "tracks";
```



```java
// Constructor for the database handler.
3 usages
public DBHandler(Context context) {

    super(context, DB_NAME, factory: null, DB_VERSION);
}//DBHandler

// ..............................Creating a database..........
@Override
public void onCreate(SQLiteDatabase db) {
    String query = "CREATE TABLE " + TABLE_NAME + " ("
            + ID_COL + " INTEGER PRIMARY KEY AUTOINCREMENT, "
            + NAME_COL + " TEXT,"
            + DURATION_COL + " TEXT,"
            + DESCRIPTION_COL + " TEXT,"
            + TRACKS_COL + " TEXT)";

    db.execSQL(query);
}//onCreate
```

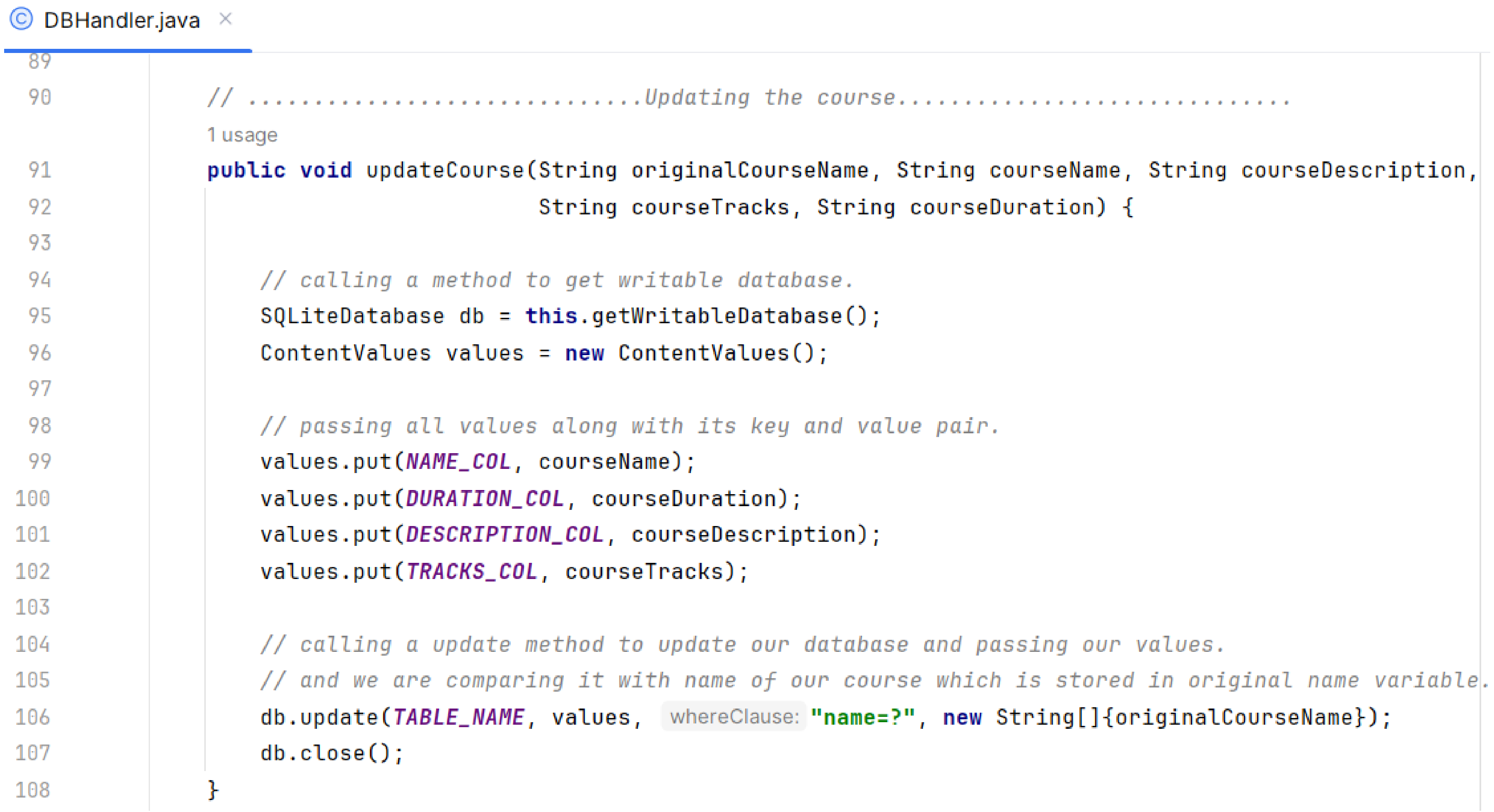


```java
// ..............................Adding a new course..............................
1 usage
public void addNewCourse(String courseName, String courseDuration, String courseDescription, String courseTracks) {

    SQLiteDatabase db = this.getWritableDatabase();

    // creating a variable for content values.
    ContentValues values = new ContentValues();

    // passing all values along with its key and value pair.
    values.put(NAME_COL, courseName);
    values.put(DURATION_COL, courseDuration);
    values.put(DESCRIPTION_COL, courseDescription);
    values.put(TRACKS_COL, courseTracks);

    // passing content values to the table.
    db.insert(TABLE_NAME,  nullColumnHack: null, values);

    // at last we are closing our database after adding database.
    db.close();
}//addNewCourse
```

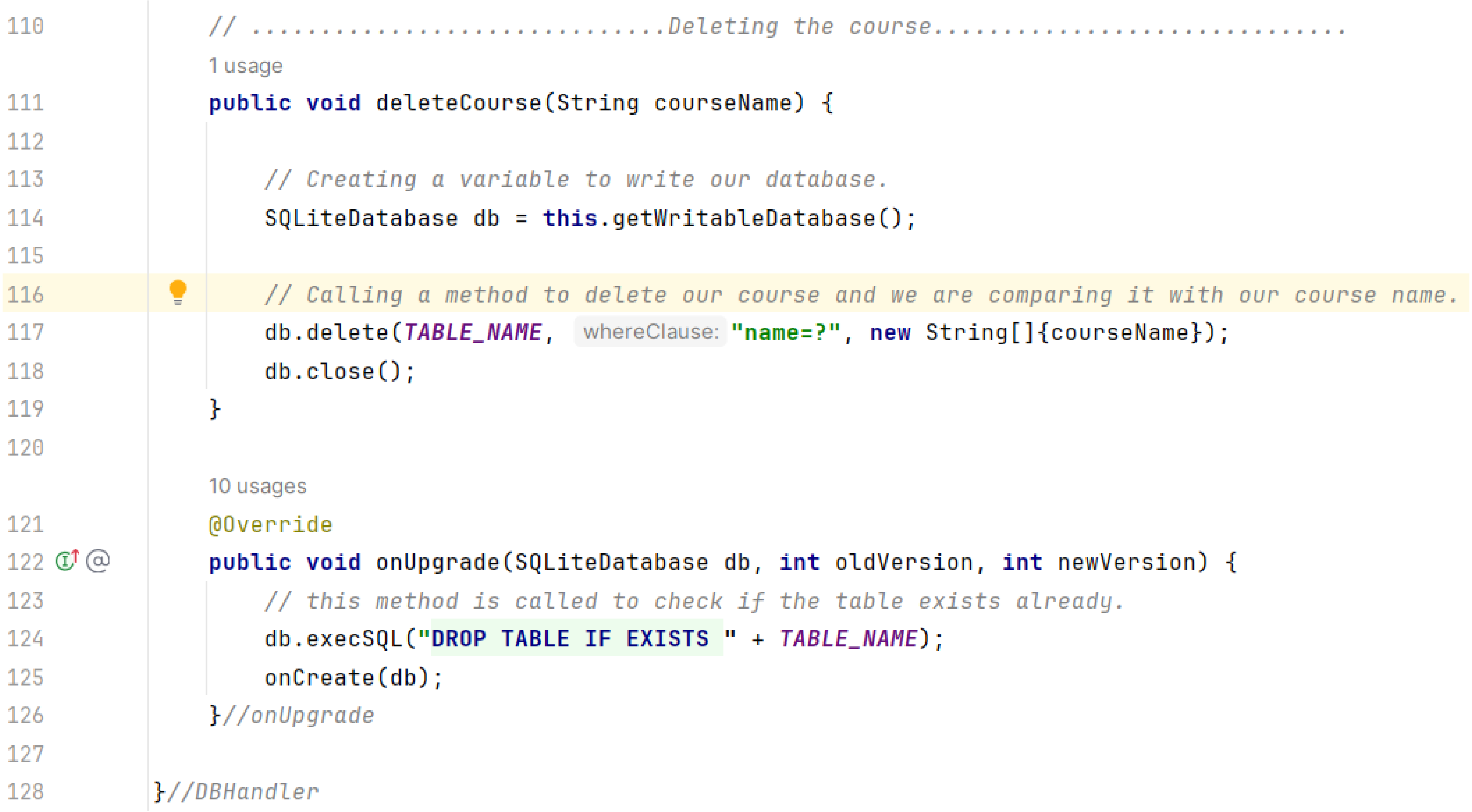


DBHandler.java

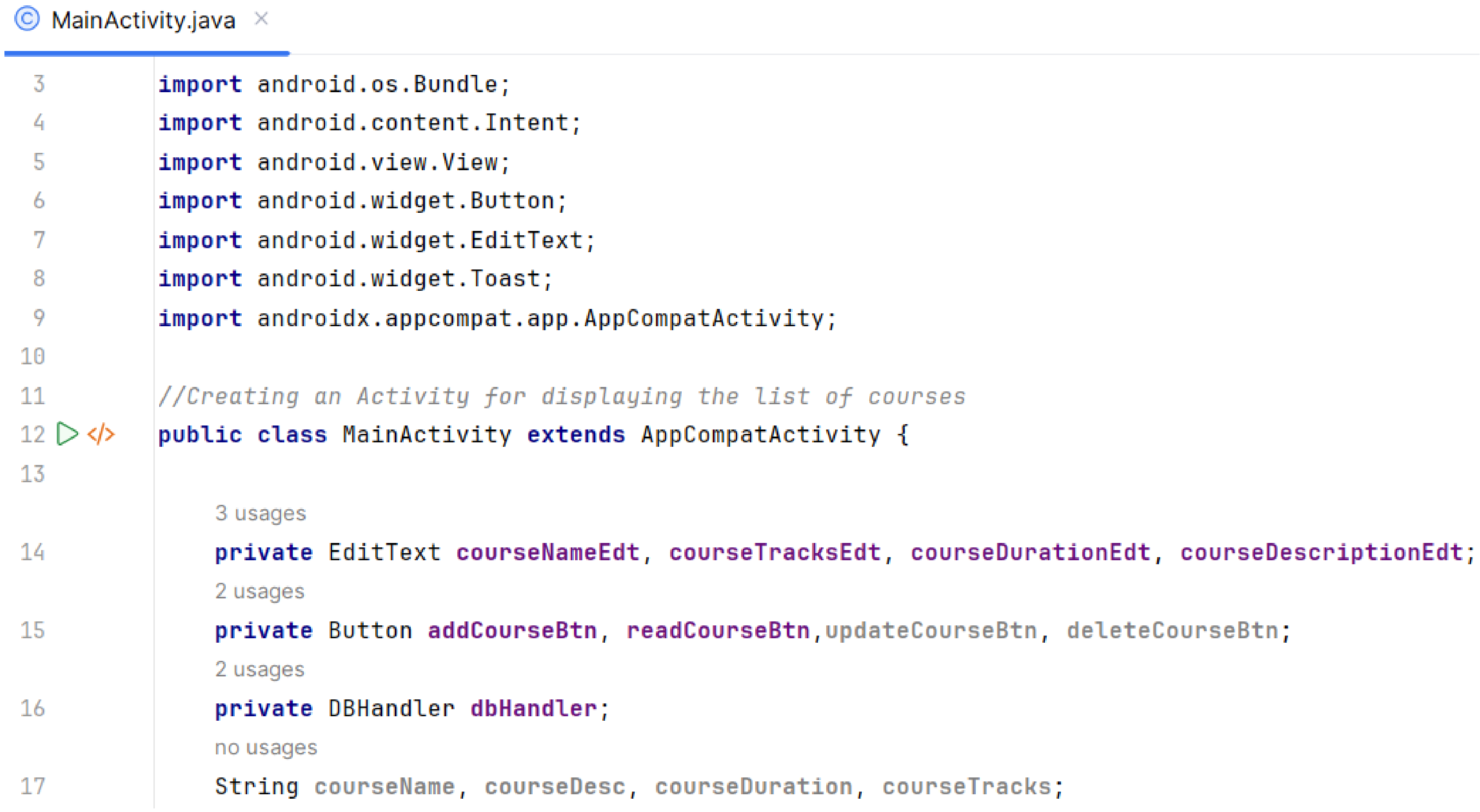
```
// ..............................Reading all the courses..............................
1 usage
public ArrayList<CourseModal> readCourses() {
    // creating a database for reading the database.
    SQLiteDatabase db = this.getReadableDatabase();

    // creating a cursor with query to read data from database.
    Cursor cursorCourses = db.rawQuery( sql: "SELECT * FROM " + TABLE_NAME,  selectionArgs: null);

    // creating a new array list.
    ArrayList<CourseModal> courseModalArrayList = new ArrayList<>();

    // moving the cursor to first position.
    if (cursorCourses.moveToFirst()) {
        do {
            // adding the data from cursor to the array list.
            courseModalArrayList.add(new CourseModal(cursorCourses.getString( columnIndex: 1),
                    cursorCourses.getString( columnIndex: 4),
                    cursorCourses.getString( columnIndex: 2),
                    cursorCourses.getString( columnIndex: 3)));
        } while (cursorCourses.moveToNext());
        // moving our cursor to next.
    }//if
    // at last closing our cursor and returning our array list.
    cursorCourses.close();
    return courseModalArrayList;
}//ArrayList
```



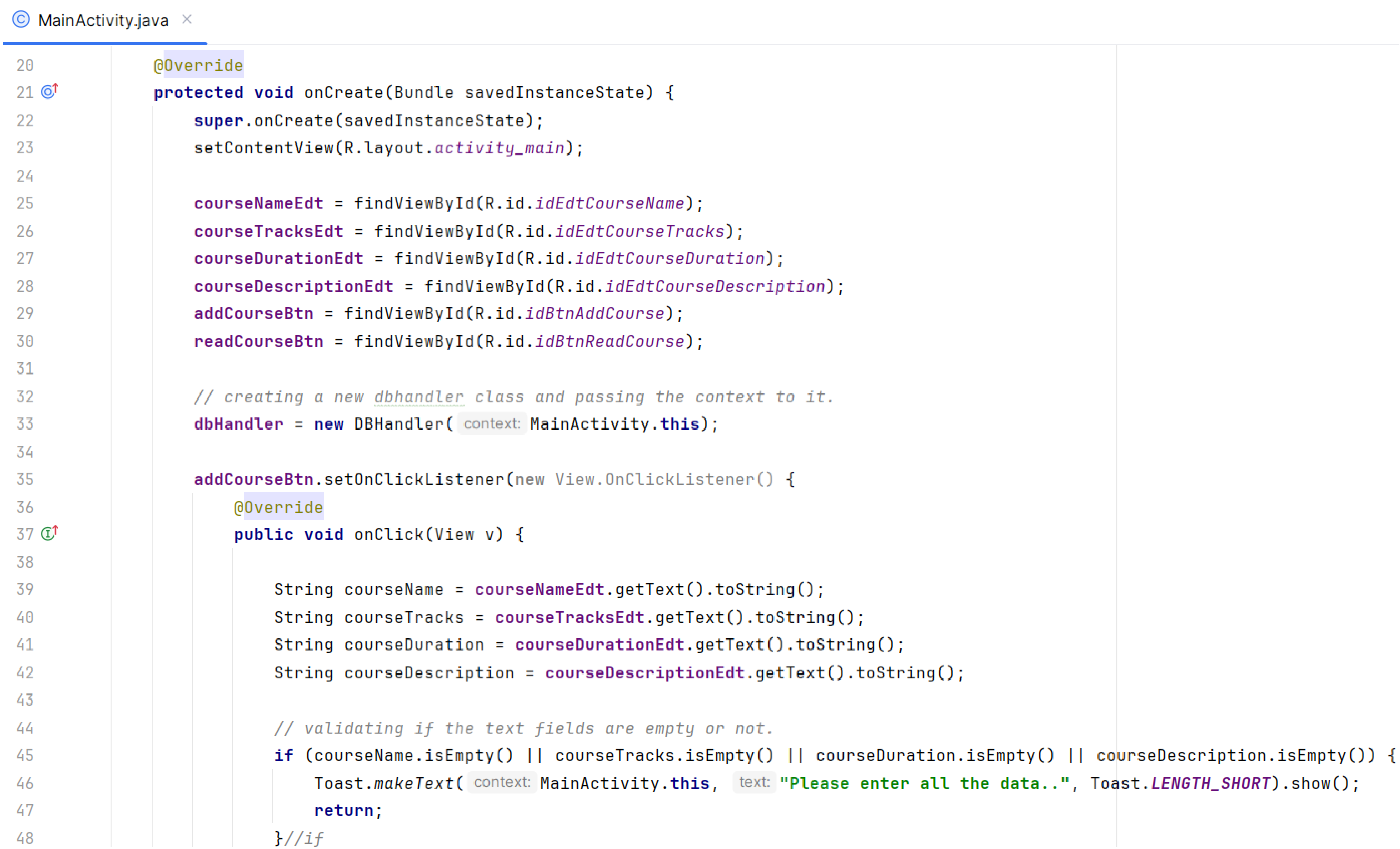
```
    // ..............................Updating the course..............................
    1 usage
    public void updateCourse(String originalCourseName, String courseName, String courseDescription,
                             String courseTracks, String courseDuration) {

        // calling a method to get writable database.
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues values = new ContentValues();

        // passing all values along with its key and value pair.
        values.put(NAME_COL, courseName);
        values.put(DURATION_COL, courseDuration);
        values.put(DESCRIPTION_COL, courseDescription);
        values.put(TRACKS_COL, courseTracks);

        // calling a update method to update our database and passing our values.
        // and we are comparing it with name of our course which is stored in original name variable.
        db.update(TABLE_NAME, values, whereClause: "name=?", new String[]{originalCourseName});
        db.close();
    }
```



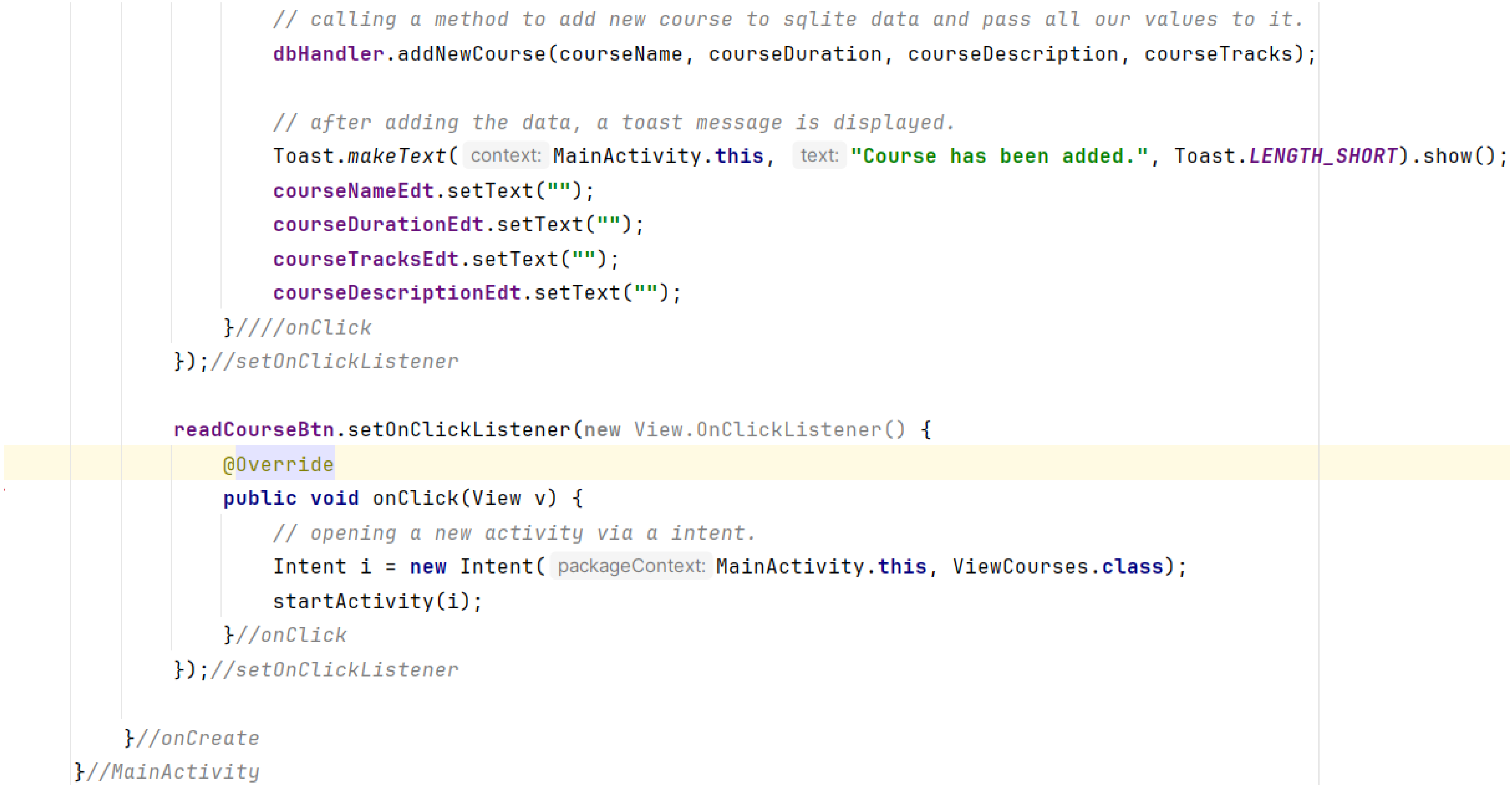
```java
    // .............................Deleting the course.............................
    public void deleteCourse(String courseName) {

        // Creating a variable to write our database.
        SQLiteDatabase db = this.getWritableDatabase();

        // Calling a method to delete our course and we are comparing it with our course name.
        db.delete(TABLE_NAME, "name=?", new String[]{courseName});
        db.close();
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        // this method is called to check if the table exists already.
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate(db);
    }//onUpgrade

}//DBHandler
```



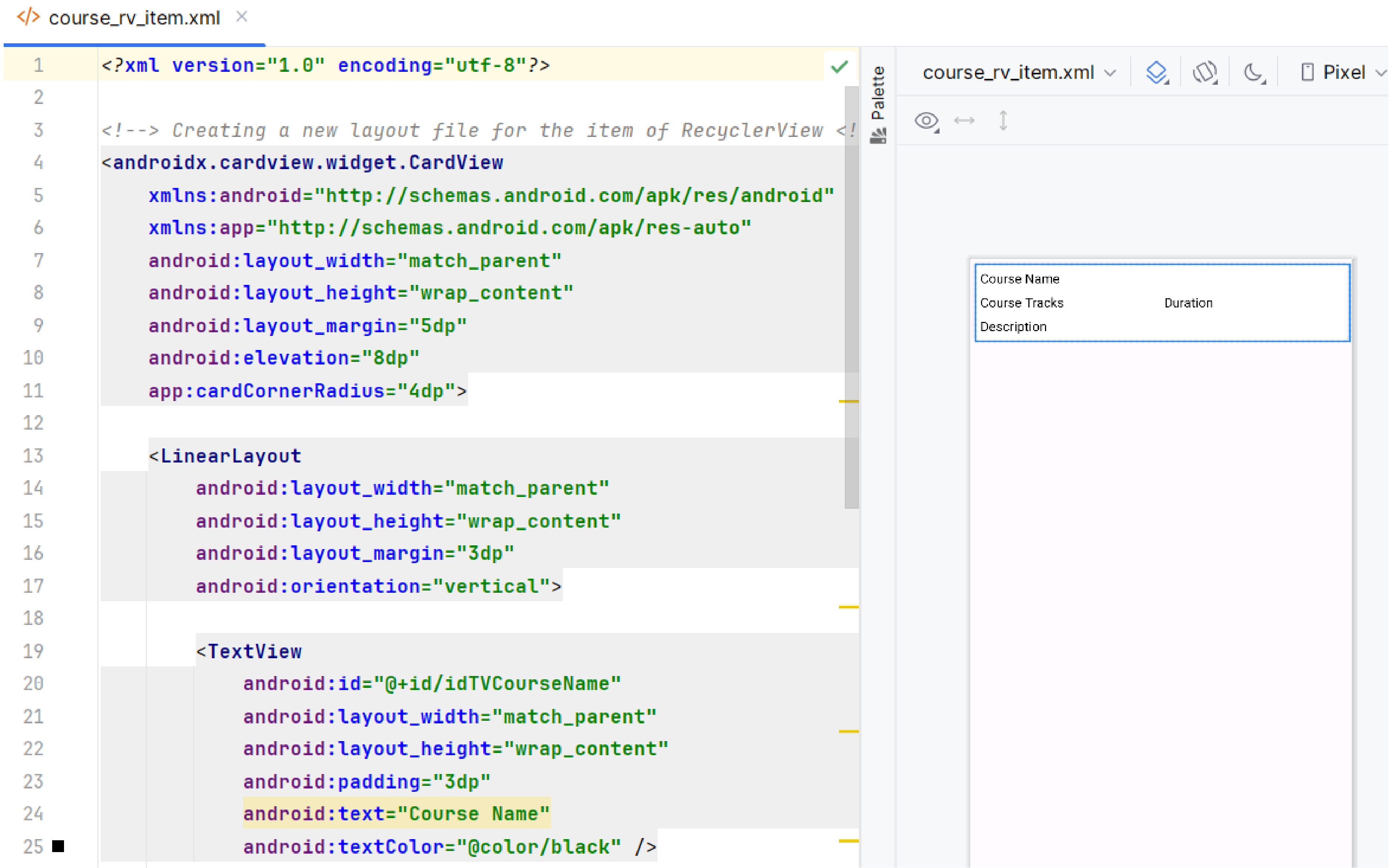
```java
import android.os.Bundle;
import android.content.Intent;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

//Creating an Activity for displaying the list of courses
public class MainActivity extends AppCompatActivity {

    private EditText courseNameEdt, courseTracksEdt, courseDurationEdt, courseDescriptionEdt;
    private Button addCourseBtn, readCourseBtn,updateCourseBtn, deleteCourseBtn;
    private DBHandler dbHandler;
    String courseName, courseDesc, courseDuration, courseTracks;
```



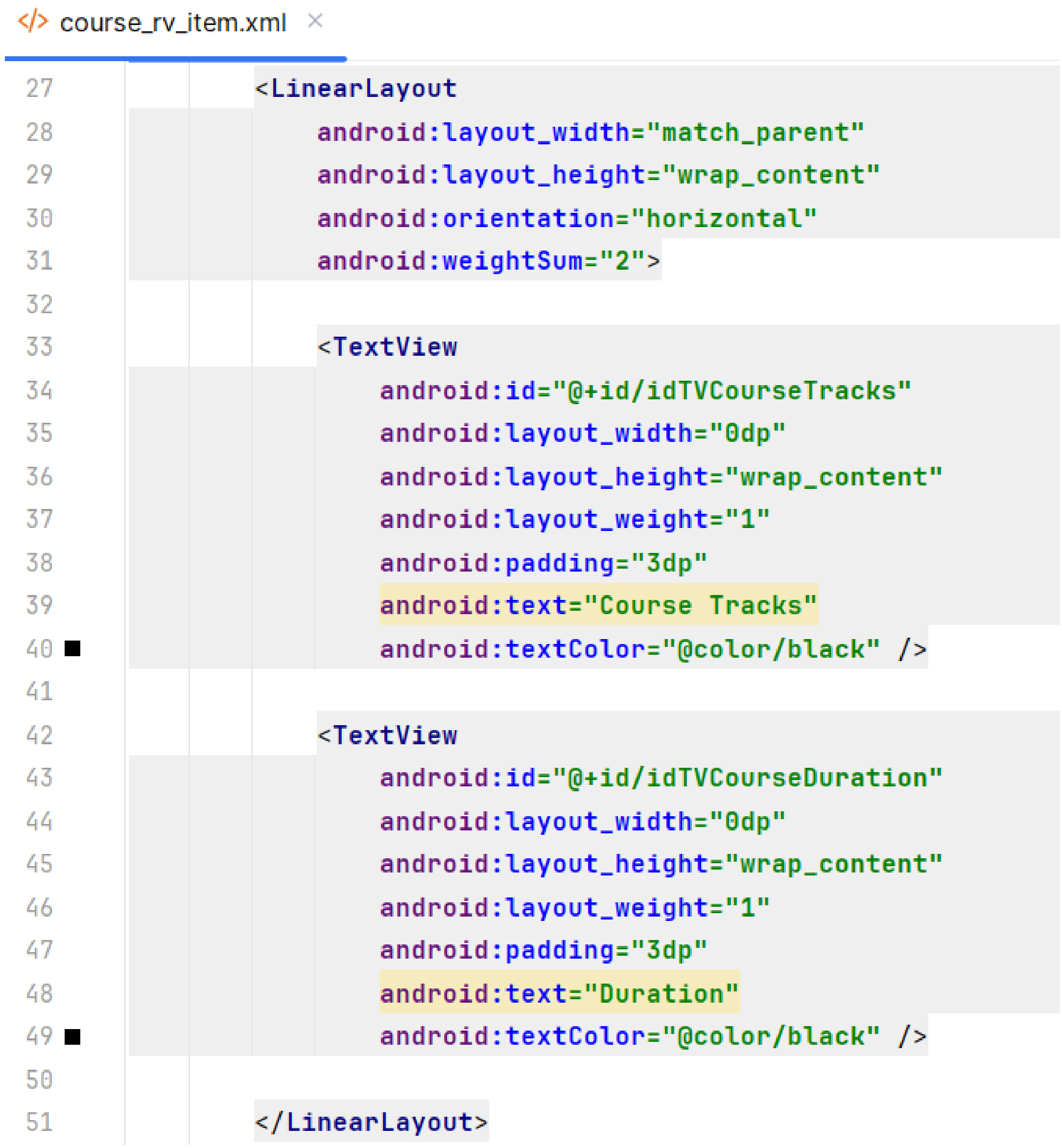
```java
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        courseNameEdt = findViewById(R.id.idEdtCourseName);
        courseTracksEdt = findViewById(R.id.idEdtCourseTracks);
        courseDurationEdt = findViewById(R.id.idEdtCourseDuration);
        courseDescriptionEdt = findViewById(R.id.idEdtCourseDescription);
        addCourseBtn = findViewById(R.id.idBtnAddCourse);
        readCourseBtn = findViewById(R.id.idBtnReadCourse);

        // creating a new dbhandler class and passing the context to it.
        dbHandler = new DBHandler( context: MainActivity.this);

        addCourseBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                String courseName = courseNameEdt.getText().toString();
                String courseTracks = courseTracksEdt.getText().toString();
                String courseDuration = courseDurationEdt.getText().toString();
                String courseDescription = courseDescriptionEdt.getText().toString();

                // validating if the text fields are empty or not.
                if (courseName.isEmpty() || courseTracks.isEmpty() || courseDuration.isEmpty() || courseDescription.isEmpty()) {
                    Toast.makeText( context: MainActivity.this,  text: "Please enter all the data..", Toast.LENGTH_SHORT).show();
                    return;
                }//if
```



```
                // calling a method to add new course to sqlite data and pass all our values to it.
                dbHandler.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
```

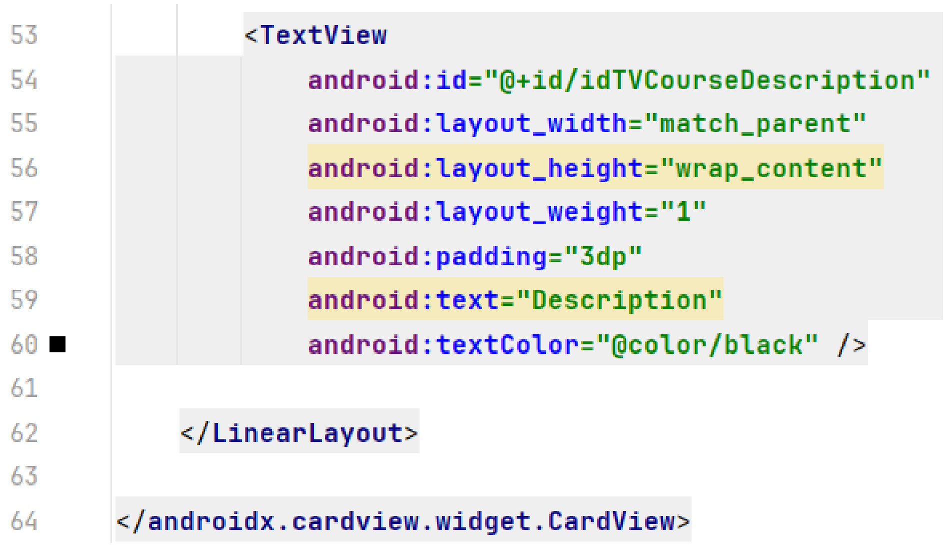


```
<?xml version="1.0" encoding="utf-8"?>

<!--> Creating a new layout file for the item of RecyclerView <!
<androidx.cardview.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:elevation="8dp"
    app:cardCornerRadius="4dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="3dp"
        android:orientation="vertical">

        <TextView
            android:id="@+id/idTVCourseName"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:padding="3dp"
            android:text="Course Name"
            android:textColor="@color/black" />
```

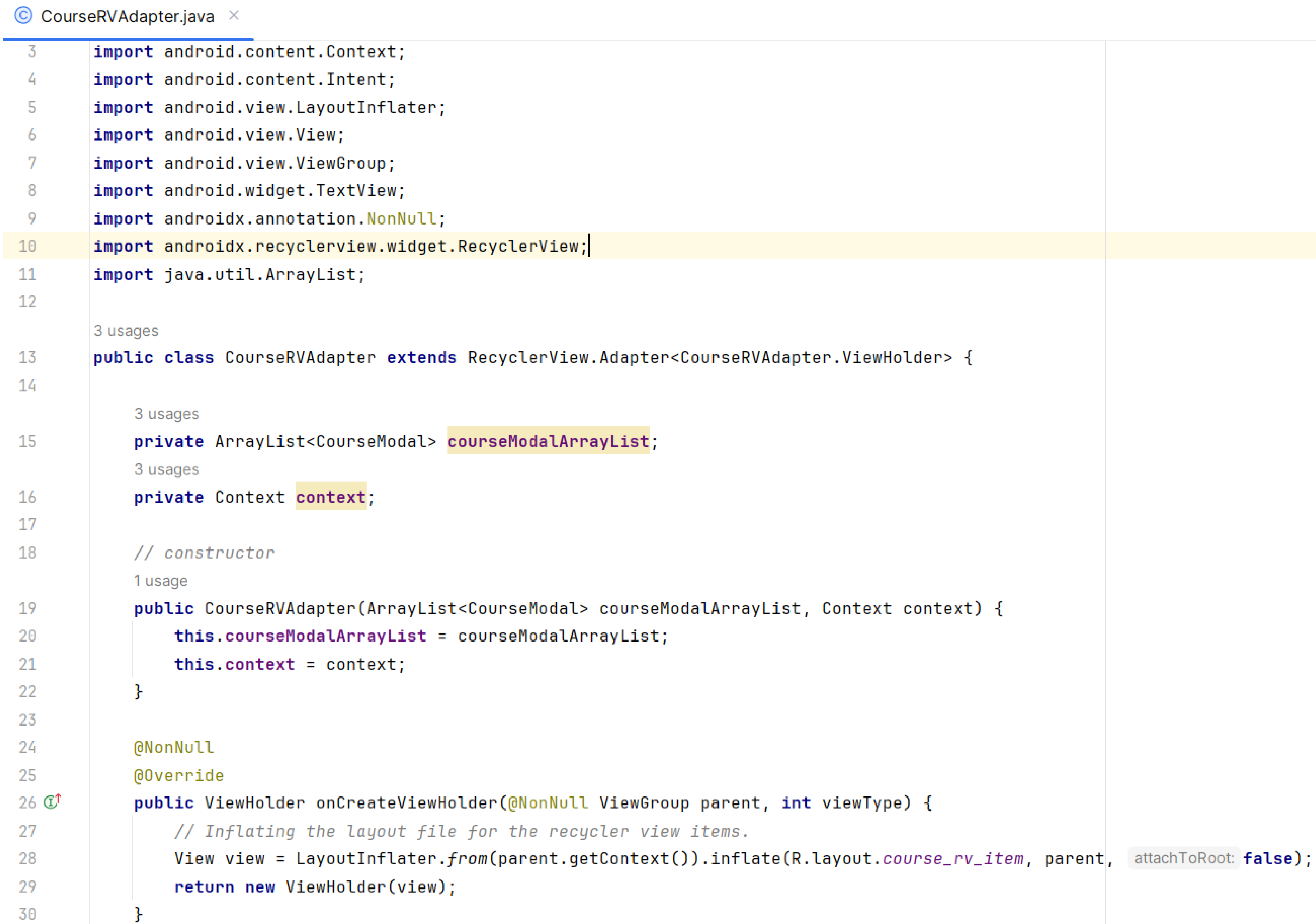


course_rv_item.xml

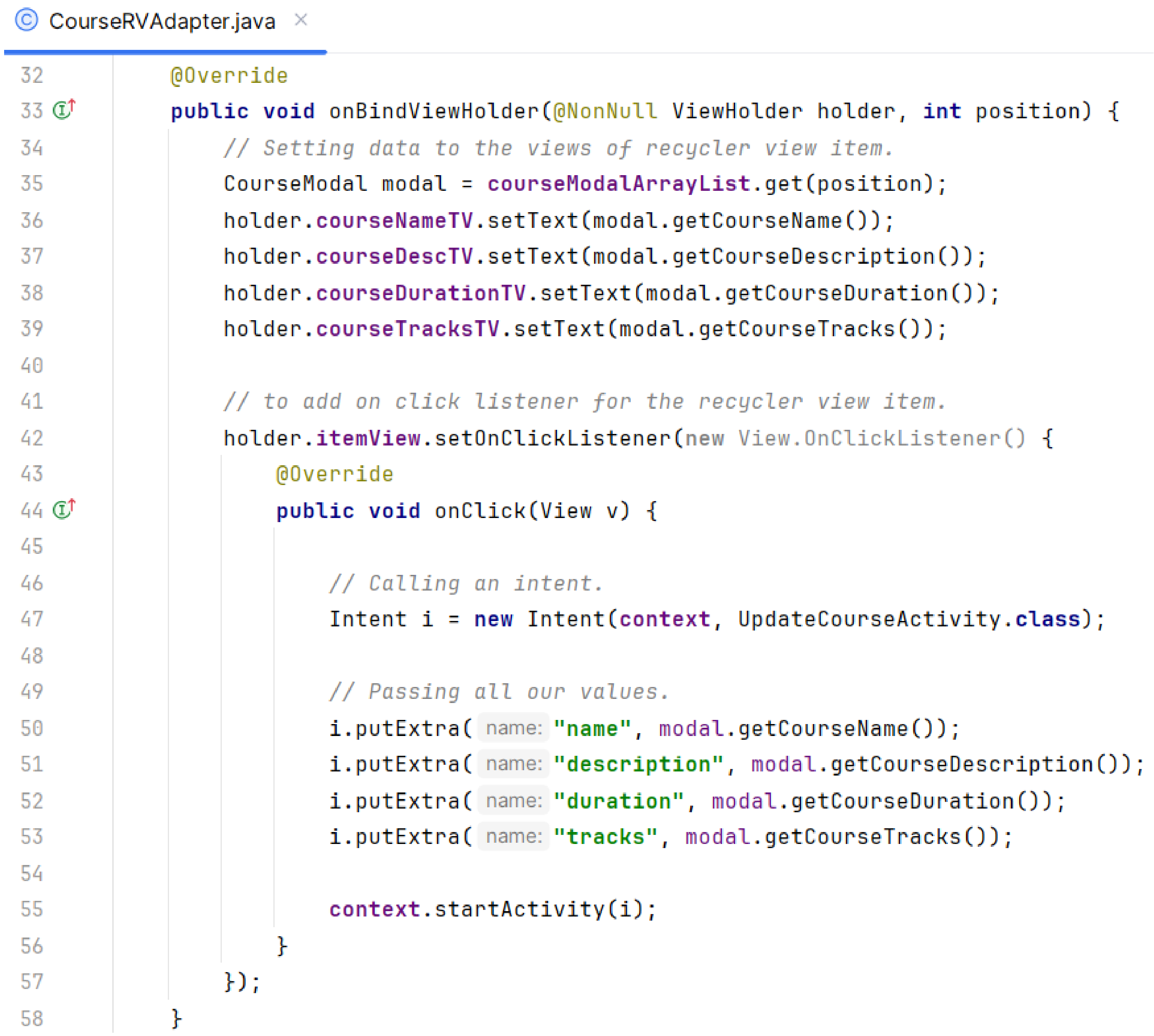
```xml
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:weightSum="2">

            <TextView
                android:id="@+id/idTVCourseTracks"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:padding="3dp"
                android:text="Course Tracks"
                android:textColor="@color/black" />

            <TextView
                android:id="@+id/idTVCourseDuration"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:padding="3dp"
                android:text="Duration"
                android:textColor="@color/black" />

        </LinearLayout>
```



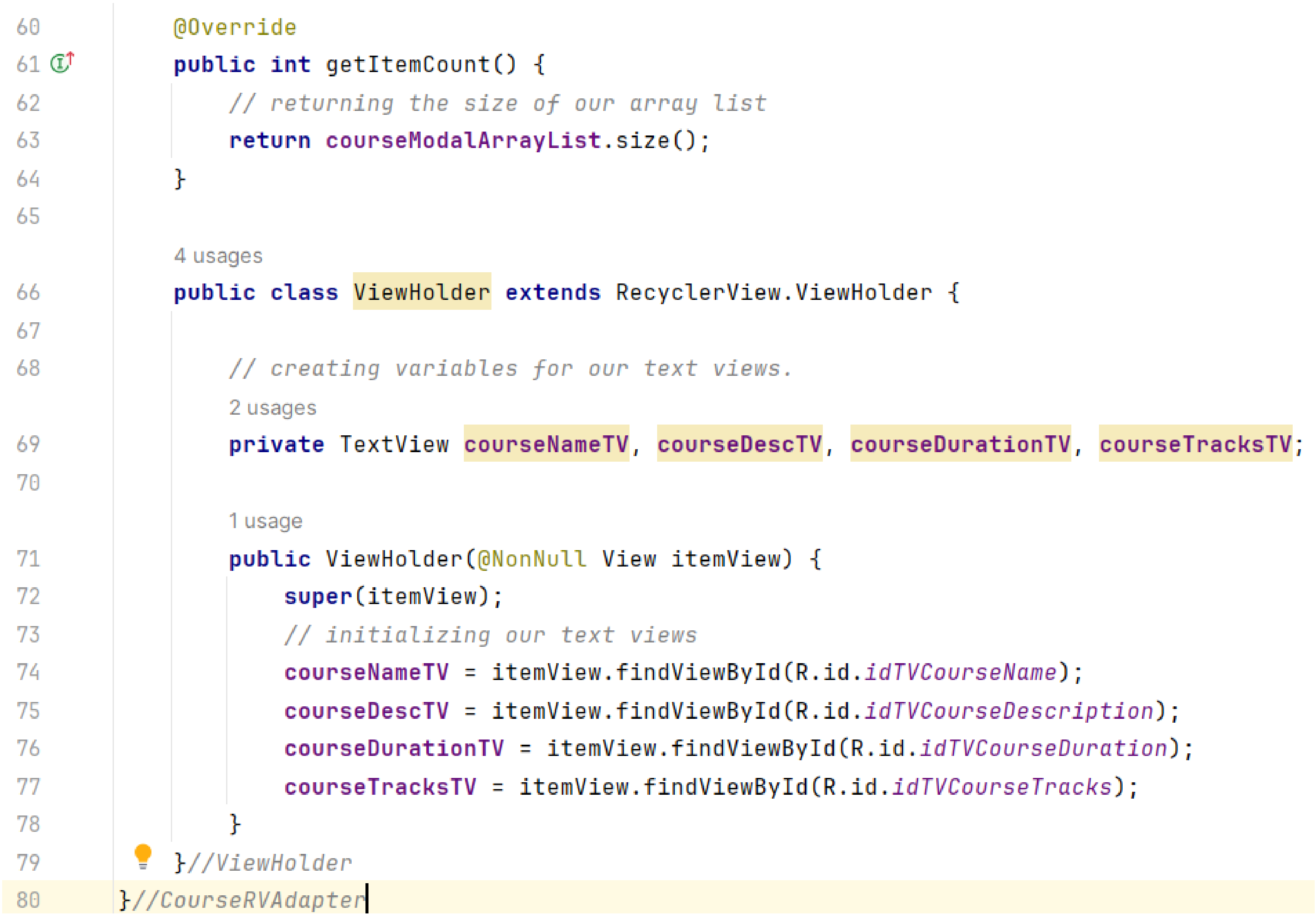
```xml
        <TextView
            android:id="@+id/idTVCourseDescription"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:padding="3dp"
            android:text="Description"
            android:textColor="@color/black" />

    </LinearLayout>

</androidx.cardview.widget.CardView>
```



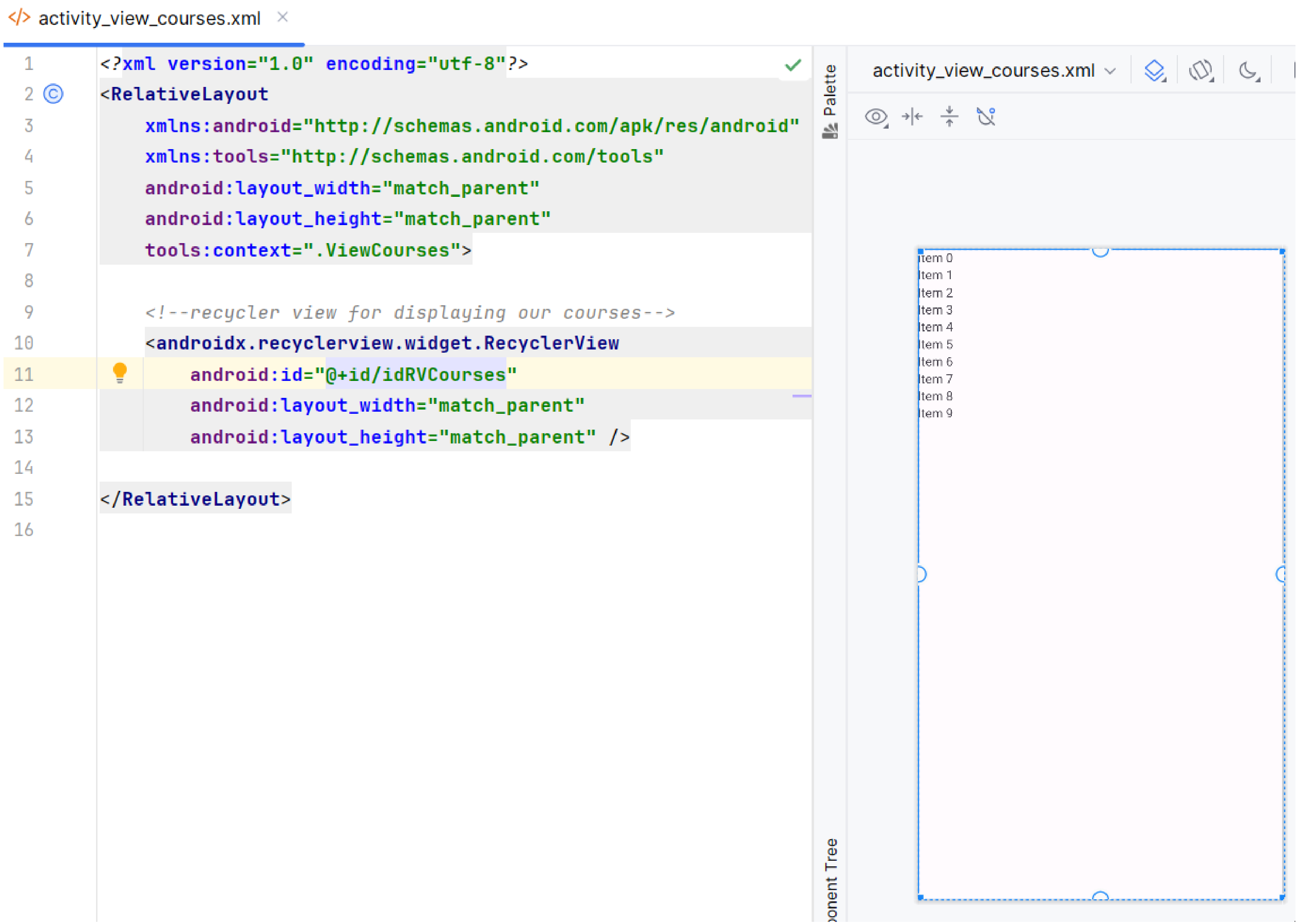
```java
import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;
import java.util.ArrayList;

3 usages
public class CourseRVAdapter extends RecyclerView.Adapter<CourseRVAdapter.ViewHolder> {

    3 usages
    private ArrayList<CourseModal> courseModalArrayList;
    3 usages
    private Context context;

    // constructor
    1 usage
    public CourseRVAdapter(ArrayList<CourseModal> courseModalArrayList, Context context) {
        this.courseModalArrayList = courseModalArrayList;
        this.context = context;
    }

    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        // Inflating the layout file for the recycler view items.
        View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.course_rv_item, parent,  attachToRoot: false);
        return new ViewHolder(view);
    }
```



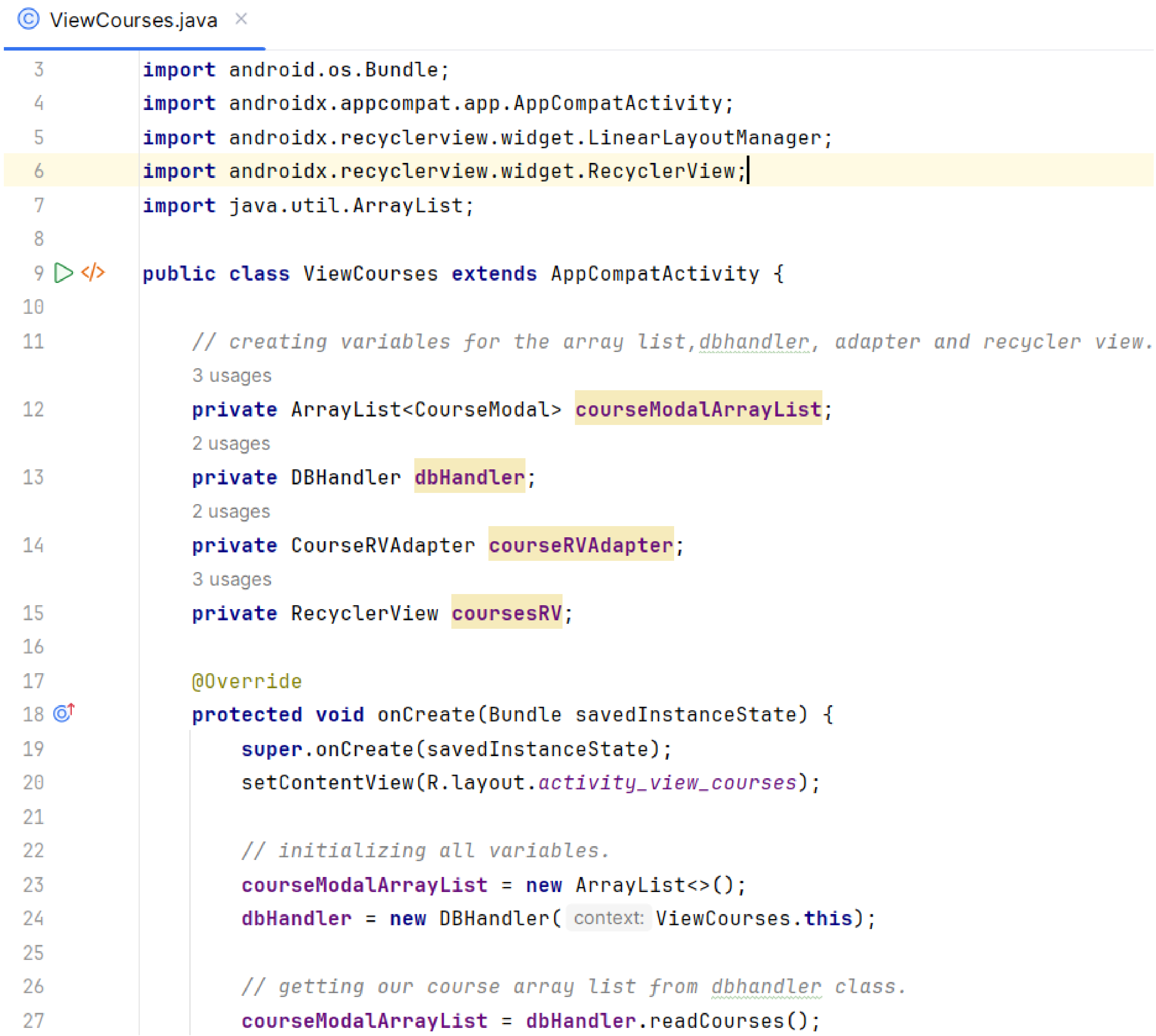
```java
    @Override
    public void onBindViewHolder(@NonNull ViewHolder holder, int position) {
        // Setting data to the views of recycler view item.
        CourseModal modal = courseModalArrayList.get(position);
        holder.courseNameTV.setText(modal.getCourseName());
        holder.courseDescTV.setText(modal.getCourseDescription());
        holder.courseDurationTV.setText(modal.getCourseDuration());
        holder.courseTracksTV.setText(modal.getCourseTracks());

        // to add on click listener for the recycler view item.
        holder.itemView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                // Calling an intent.
                Intent i = new Intent(context, UpdateCourseActivity.class);

                // Passing all our values.
                i.putExtra( name: "name", modal.getCourseName());
                i.putExtra( name: "description", modal.getCourseDescription());
                i.putExtra( name: "duration", modal.getCourseDuration());
                i.putExtra( name: "tracks", modal.getCourseTracks());

                context.startActivity(i);
            }
        });
    }
```



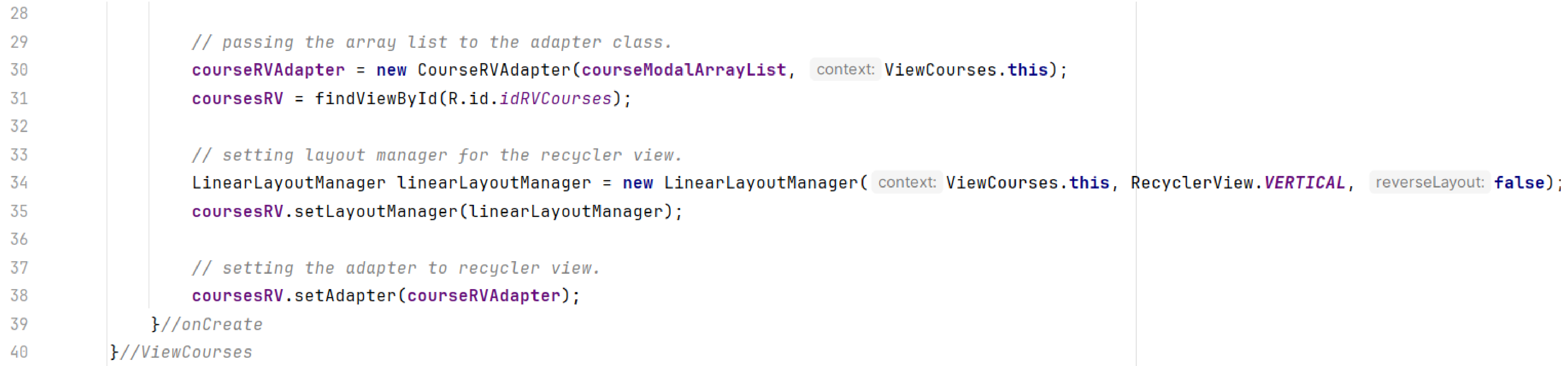
```java
    @Override
    public int getItemCount() {
        // returning the size of our array list
        return courseModalArrayList.size();
    }

    4 usages
    public class ViewHolder extends RecyclerView.ViewHolder {

        // creating variables for our text views.
        2 usages
        private TextView courseNameTV, courseDescTV, courseDurationTV, courseTracksTV;

        1 usage
        public ViewHolder(@NonNull View itemView) {
            super(itemView);
            // initializing our text views
            courseNameTV = itemView.findViewById(R.id.idTVCourseName);
            courseDescTV = itemView.findViewById(R.id.idTVCourseDescription);
            courseDurationTV = itemView.findViewById(R.id.idTVCourseDuration);
            courseTracksTV = itemView.findViewById(R.id.idTVCourseTracks);
        }
    }//ViewHolder
}//CourseRVAdapter
```



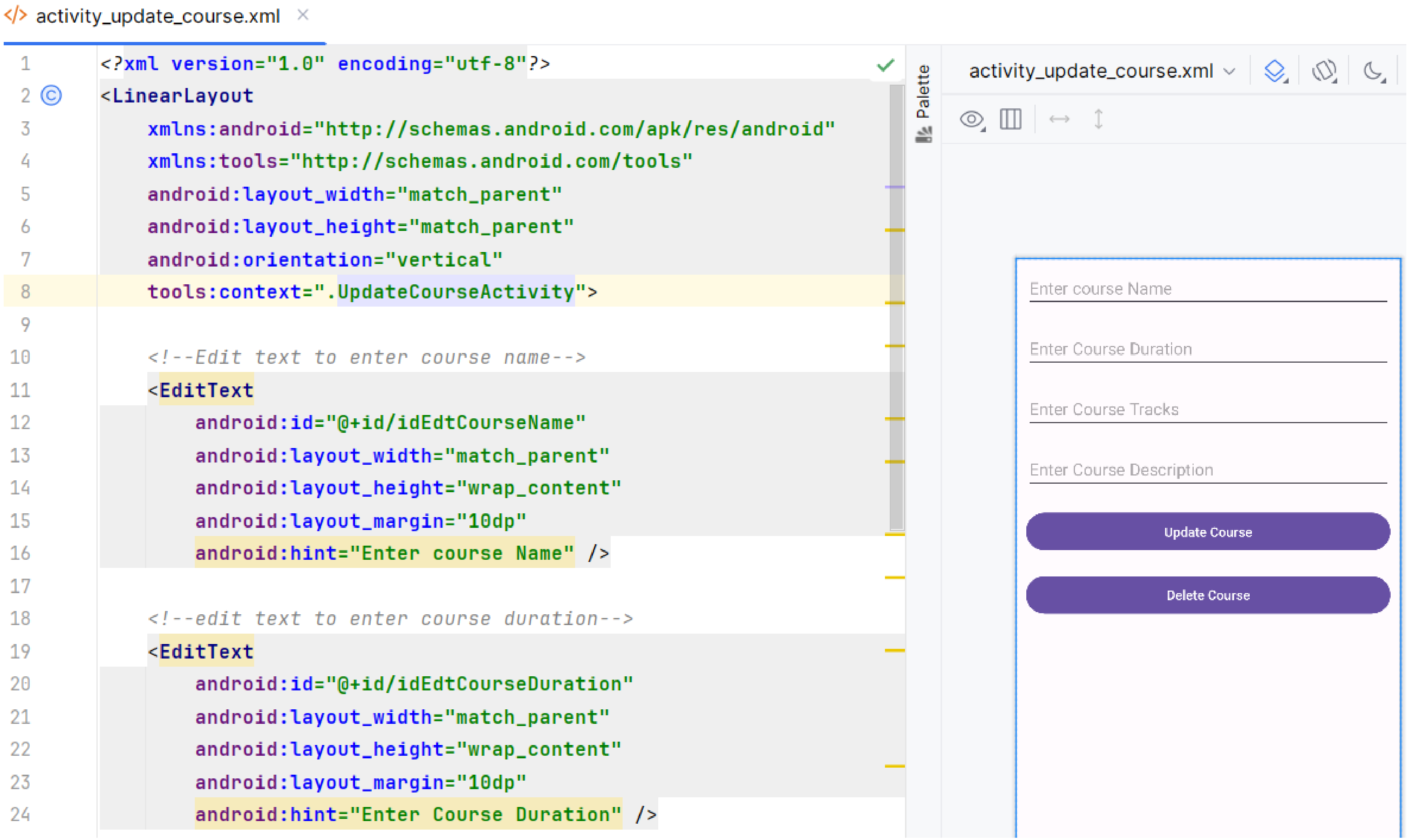
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ViewCourses">

    <!--recycler view for displaying our courses-->
    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/idRVCourses"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</RelativeLayout>
```



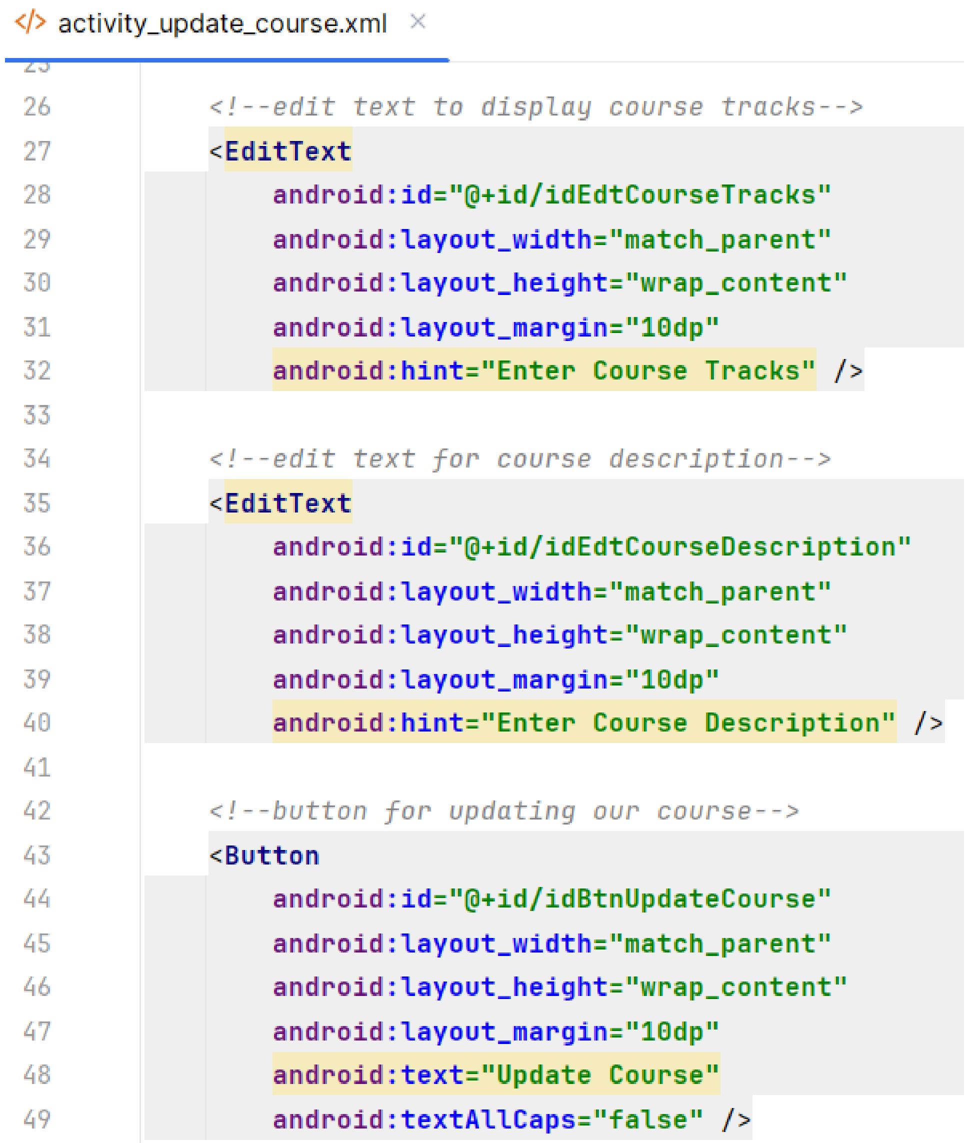
```java
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import java.util.ArrayList;

public class ViewCourses extends AppCompatActivity {

    // creating variables for the array list,dbhandler, adapter and recycler view.
    3 usages
    private ArrayList<CourseModal> courseModalArrayList;
    2 usages
    private DBHandler dbHandler;
    2 usages
    private CourseRVAdapter courseRVAdapter;
    3 usages
    private RecyclerView coursesRV;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view_courses);

        // initializing all variables.
        courseModalArrayList = new ArrayList<>();
        dbHandler = new DBHandler( context: ViewCourses.this);

        // getting our course array list from dbhandler class.
        courseModalArrayList = dbHandler.readCourses();
```



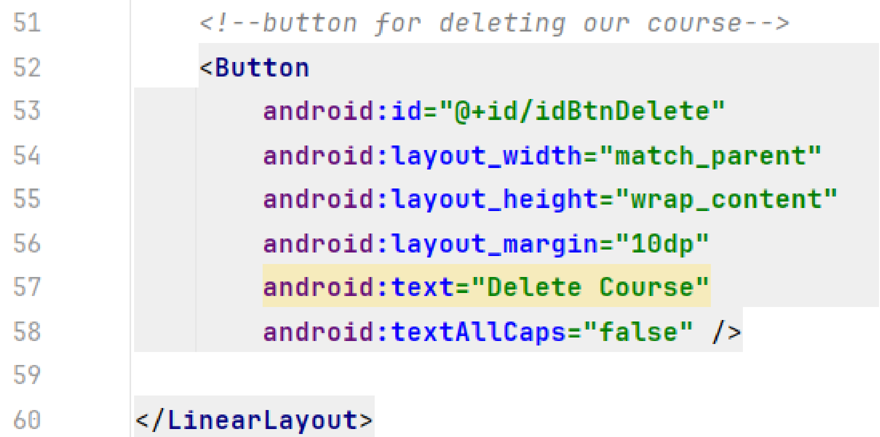
```java

        // passing the array list to the adapter class.
        courseRVAdapter = new CourseRVAdapter(courseModalArrayList,  context: ViewCourses.this);
        coursesRV = findViewById(R.id.idRVCourses);

        // setting layout manager for the recycler view.
        LinearLayoutManager linearLayoutManager = new LinearLayoutManager( context: ViewCourses.this, RecyclerView.VERTICAL,  reverseLayout: false);
        coursesRV.setLayoutManager(linearLayoutManager);

        // setting the adapter to recycler view.
        coursesRV.setAdapter(courseRVAdapter);
    }//onCreate
}//ViewCourses
```



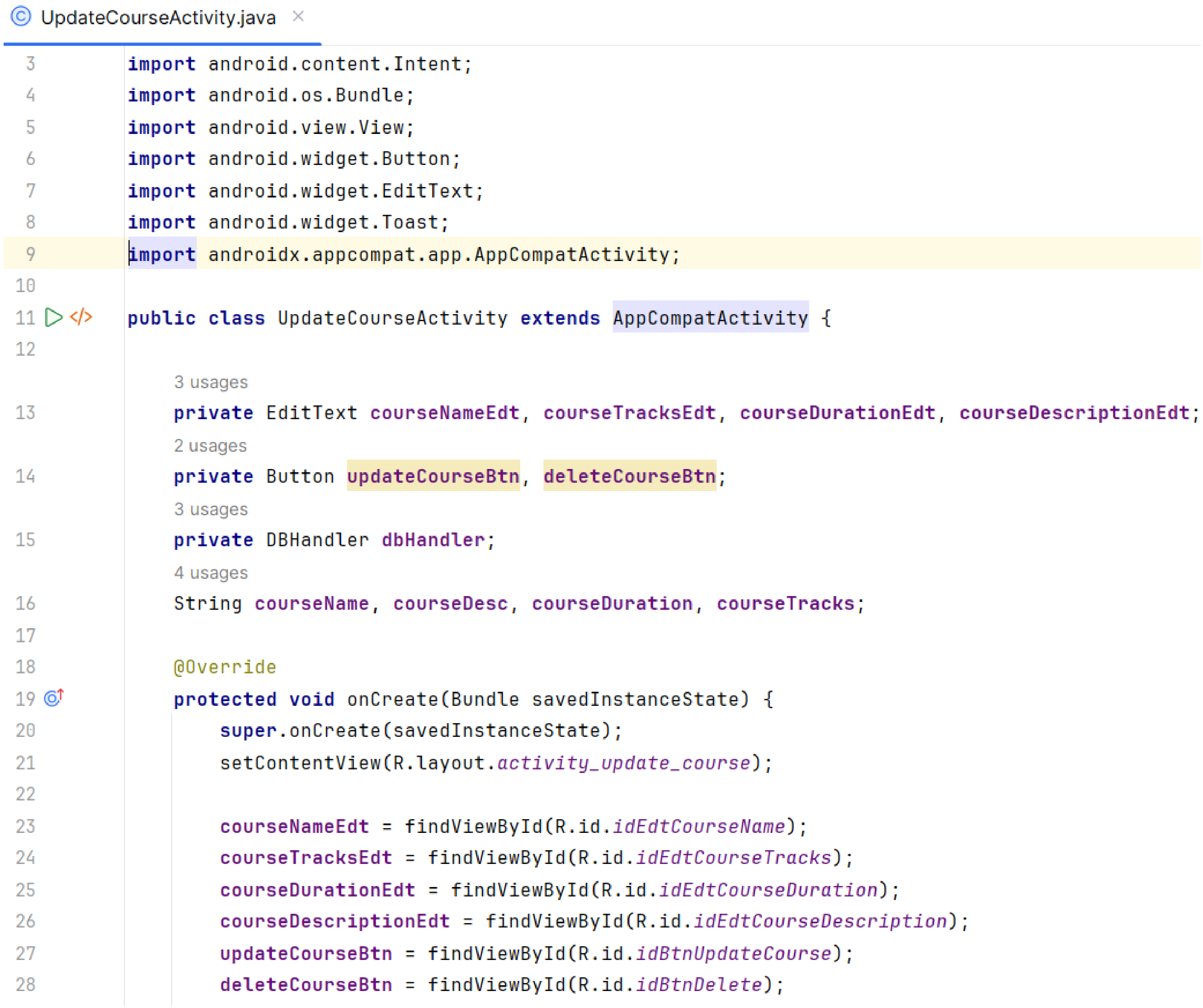
```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".UpdateCourseActivity">

    <!--Edit text to enter course name-->
    <EditText
        android:id="@+id/idEdtCourseName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:hint="Enter course Name" />

    <!--edit text to enter course duration-->
    <EditText
        android:id="@+id/idEdtCourseDuration"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:hint="Enter Course Duration" />
```



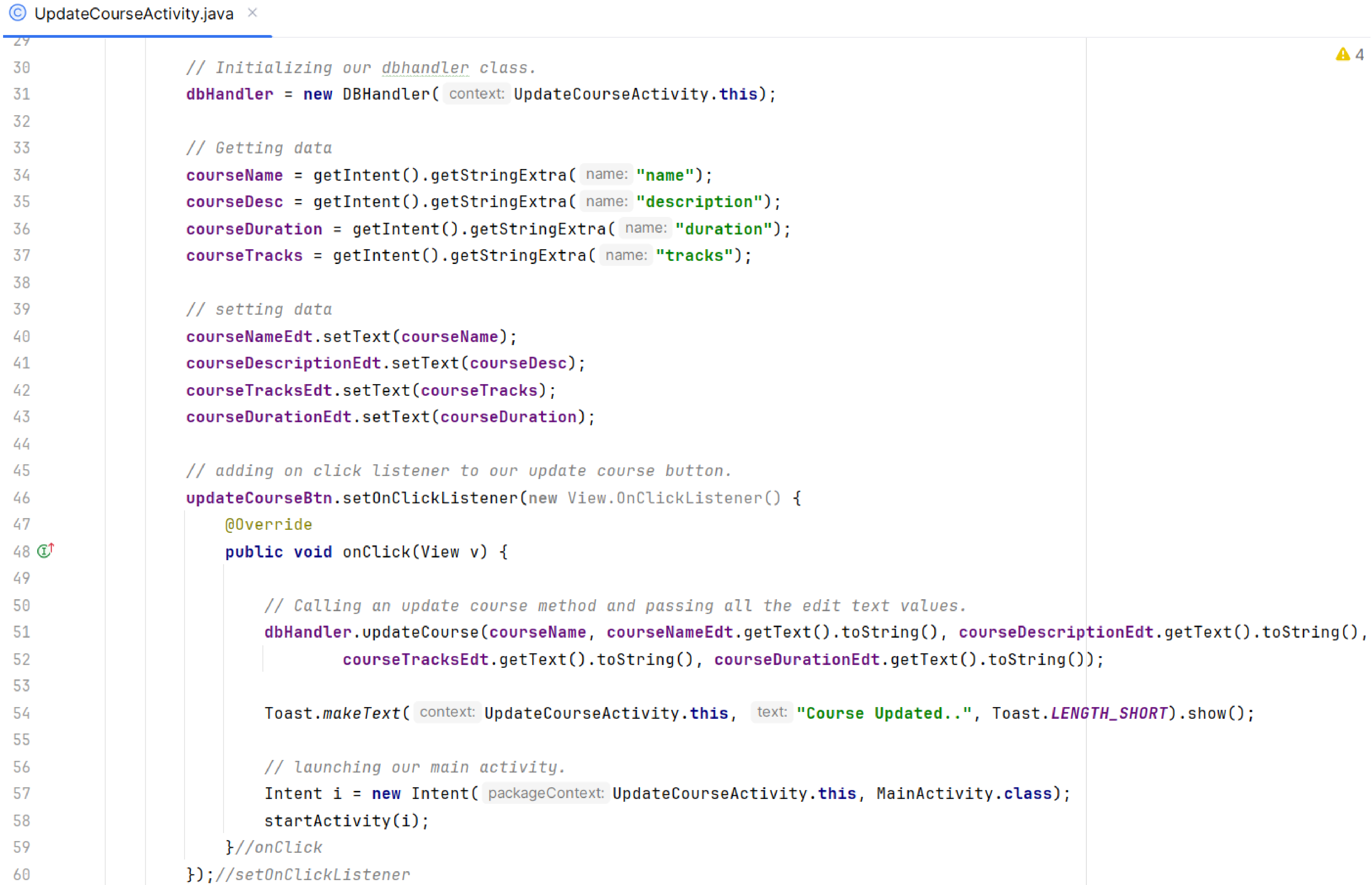
```xml
    <!--edit text to display course tracks-->
    <EditText
        android:id="@+id/idEdtCourseTracks"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:hint="Enter Course Tracks" />

    <!--edit text for course description-->
    <EditText
        android:id="@+id/idEdtCourseDescription"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:hint="Enter Course Description" />

    <!--button for updating our course-->
    <Button
        android:id="@+id/idBtnUpdateCourse"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:text="Update Course"
        android:textAllCaps="false" />
```



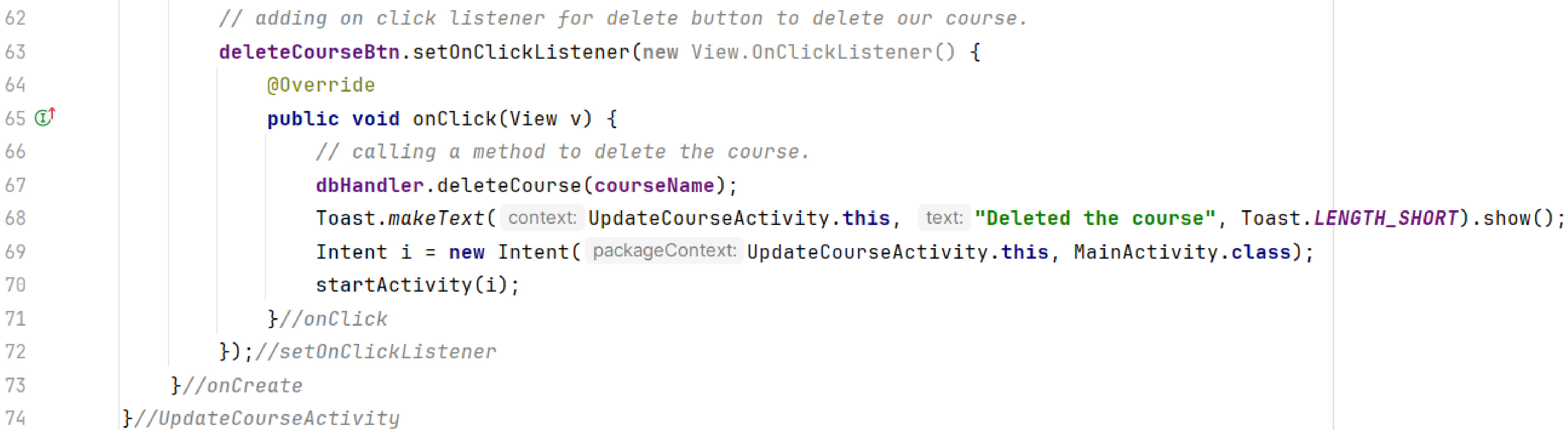
```xml
    <!--button for deleting our course-->
    <Button
        android:id="@+id/idBtnDelete"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:text="Delete Course"
        android:textAllCaps="false" />

</LinearLayout>
```



```
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class UpdateCourseActivity extends AppCompatActivity {

    3 usages
    private EditText courseNameEdt, courseTracksEdt, courseDurationEdt, courseDescriptionEdt;
    2 usages
    private Button updateCourseBtn, deleteCourseBtn;
    3 usages
    private DBHandler dbHandler;
    4 usages
    String courseName, courseDesc, courseDuration, courseTracks;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_update_course);

        courseNameEdt = findViewById(R.id.idEdtCourseName);
        courseTracksEdt = findViewById(R.id.idEdtCourseTracks);
        courseDurationEdt = findViewById(R.id.idEdtCourseDuration);
        courseDescriptionEdt = findViewById(R.id.idEdtCourseDescription);
        updateCourseBtn = findViewById(R.id.idBtnUpdateCourse);
        deleteCourseBtn = findViewById(R.id.idBtnDelete);
```



```java
        // Initializing our dbhandler class.
        dbHandler = new DBHandler( context: UpdateCourseActivity.this);

        // Getting data
        courseName = getIntent().getStringExtra( name: "name");
        courseDesc = getIntent().getStringExtra( name: "description");
        courseDuration = getIntent().getStringExtra( name: "duration");
        courseTracks = getIntent().getStringExtra( name: "tracks");

        // setting data
        courseNameEdt.setText(courseName);
        courseDescriptionEdt.setText(courseDesc);
        courseTracksEdt.setText(courseTracks);
        courseDurationEdt.setText(courseDuration);

        // adding on click listener to our update course button.
        updateCourseBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                // Calling an update course method and passing all the edit text values.
                dbHandler.updateCourse(courseName, courseNameEdt.getText().toString(), courseDescriptionEdt.getText().toString(),
                        courseTracksEdt.getText().toString(), courseDurationEdt.getText().toString());

                Toast.makeText( context: UpdateCourseActivity.this,  text: "Course Updated..", Toast.LENGTH_SHORT).show();

                // launching our main activity.
                Intent i = new Intent( packageContext: UpdateCourseActivity.this, MainActivity.class);
                startActivity(i);
            }//onClick
        });//setOnClickListener
```



```java
        // adding on click listener for delete button to delete our course.
        deleteCourseBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // calling a method to delete the course.
                dbHandler.deleteCourse(courseName);
                Toast.makeText( context: UpdateCourseActivity.this,  text: "Deleted the course", Toast.LENGTH_SHORT).show();
                Intent i = new Intent( packageContext: UpdateCourseActivity.this, MainActivity.class);
                startActivity(i);
            }//onClick
        });//setOnClickListener
    }//onCreate
}//UpdateCourseActivity
```