# 实验2：Activity

## 一、实验目的

* 掌握基于Activity的Android APP的设计和编码方法，根据需求完成APP界面和Activity之间的数据传递功能。

## 二、实验要求

* 实现上述的基本功能要求。说明：题库内容可以写在代码中。在此基础上，可以进一步优化界面，也可以进一步添加其他功能。
* 实验报告提交内容：1）结合系统界面（截图）介绍APP功能；2）核心实现代码（Java或Kotlin代码）和布局文件等。

## 三、实验内容

step1：

Creating the new Project

1. Click on the File option at the topmost corner in the left.

2. Then click on new and open a new project and name the project.

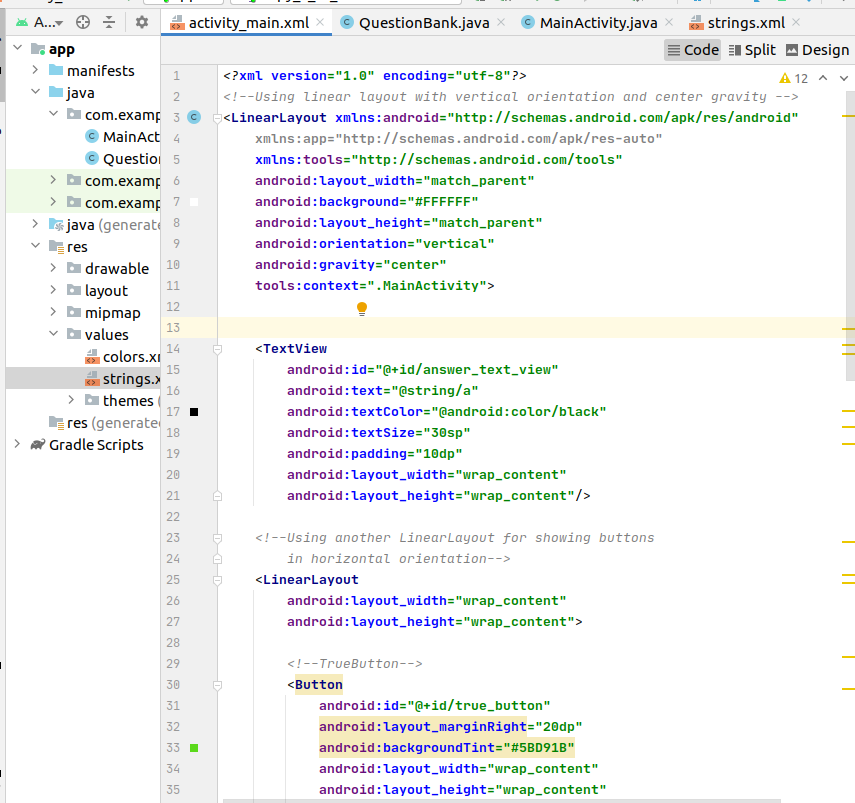
3. Now select the Empty Activity with language as Java.

4. Name it MyQuizApp.

Step2:

Designing the UI with Activity\_main.xml

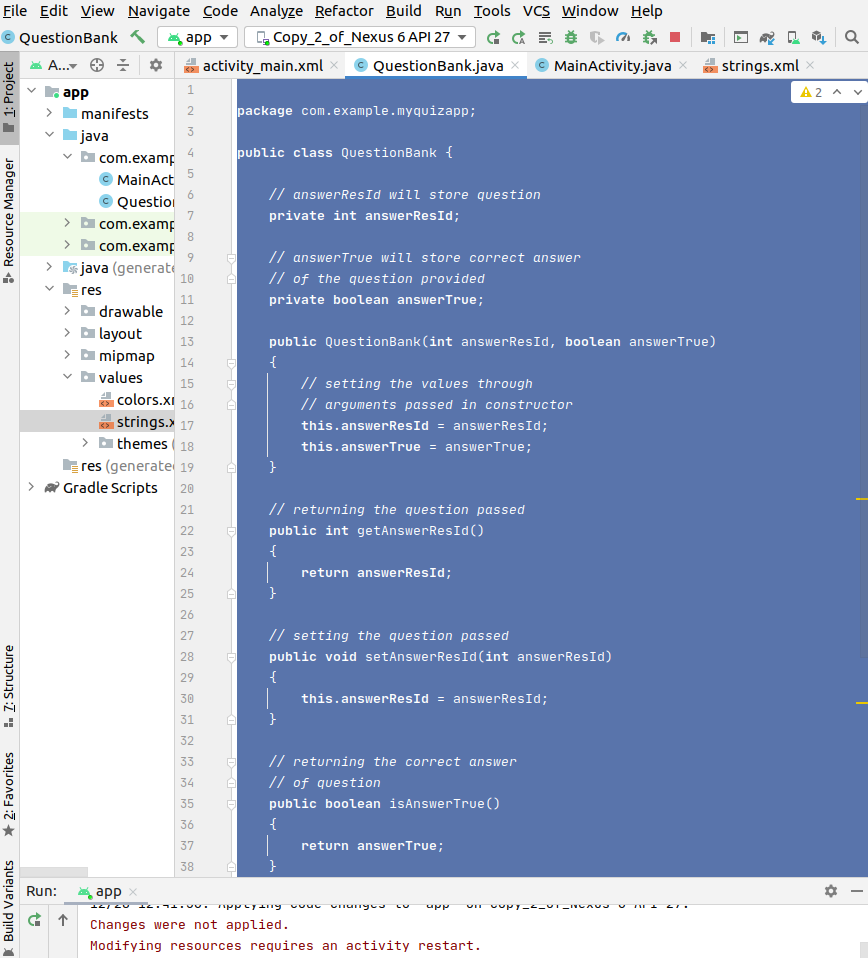
*<?***xml version="1.0" encoding="utf-8"***?>*  
*<!--Using linear layout with vertical orientation and center gravity -->*  
<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"**  
 **xmlns:app="http://schemas.android.com/apk/res-auto"**  
 **xmlns:tools="http://schemas.android.com/tools"**  
 **android:layout\_width="match\_parent"**  
 **android:background="#FFFFFF"**  
 **android:layout\_height="match\_parent"**  
 **android:orientation="vertical"**  
 **android:gravity="center"**  
 **tools:context=".MainActivity"**>  
  
  
 <**TextView**  
 **android:id="@+id/answer\_text\_view"**  
 **android:text="@string/a"**  
 **android:textColor="@android:color/black"**  
 **android:textSize="30sp"**  
 **android:padding="10dp"**  
 **android:layout\_width="wrap\_content"**  
 **android:layout\_height="wrap\_content"**/>  
  
 *<!--Using another LinearLayout for showing buttons*  
 *in horizontal orientation-->*  
<**LinearLayout**  
 **android:layout\_width="wrap\_content"**  
 **android:layout\_height="wrap\_content"**>  
  
 *<!--TrueButton-->*  
<**Button**  
 **android:id="@+id/true\_button"**  
 **android:layout\_marginRight="20dp"**  
 **android:backgroundTint="#5BD91B"**  
 **android:layout\_width="wrap\_content"**  
 **android:layout\_height="wrap\_content"**  
 **android:textSize="20sp"**  
 **android:text="@string/true\_text"** />  
  
 *<!--FalseButton-->*  
<**Button**  
 **android:id="@+id/false\_button"**  
 **android:layout\_marginLeft="20dp"**  
 **android:layout\_width="wrap\_content"**  
 **android:backgroundTint="#E33328"**  
 **android:layout\_height="wrap\_content"**  
 **android:textSize="20sp"**  
 **android:text="@string/false\_text"** />  
  
 </**LinearLayout**>  
  
 <**LinearLayout**  
 **android:layout\_width="wrap\_content"**  
 **android:layout\_height="wrap\_content"**>  
  
 *<!--PreviousButton-->*  
<**ImageButton**  
 **android:id="@+id/prev\_button"**  
 **android:layout\_width="wrap\_content"**  
 **android:layout\_height="wrap\_content"**  
 **android:backgroundTint="#DFD2D1"**  
 **android:text="prev\_text"**  
  
/>  
  
*<!-- android:src="@drawable/baseline\_keyboard\_arrow\_left\_black\_18dp"-->*  
  
  
  
 *<!--NextButton-->*  
<**ImageButton**  
 **android:id="@+id/next\_button"**  
 **android:layout\_width="wrap\_content"**  
 **android:layout\_height="wrap\_content"**  
 **android:backgroundTint="#DFD2D1"**  
/>  
*<!-- android:src="@drawable/baseline\_keyboard\_arrow\_right\_black\_18dp"*  
*android:text="@string/next\_text" -->*  
  
  
</**LinearLayout**>  
</**LinearLayout**>

.

step 3: Working with the Question

QuestionBank.java

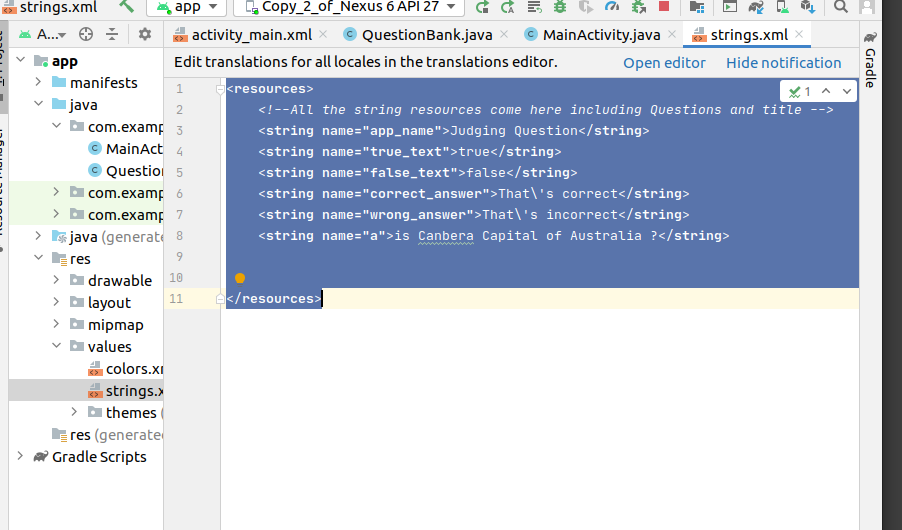
**package** com.example.myquizapp;  
  
**public class** QuestionBank {  
  
 *// answerResId will store question*  
**private int answerResId**;  
  
 *// answerTrue will store correct answer*  
 *// of the question provided*  
**private boolean answerTrue**;  
  
 **public** QuestionBank(**int** answerResId, **boolean** answerTrue)  
 {  
 *// setting the values through*  
 *// arguments passed in constructor*  
**this**.**answerResId** = answerResId;  
 **this**.**answerTrue** = answerTrue;  
 }  
  
 *// returning the question passed*  
**public int** getAnswerResId()  
 {  
 **return answerResId**;  
 }  
  
 *// setting the question passed*  
**public void** setAnswerResId(**int** answerResId)  
 {  
 **this**.**answerResId** = answerResId;  
 }  
  
 *// returning the correct answer*  
 *// of question*  
**public boolean** isAnswerTrue()  
 {  
 **return answerTrue**;  
 }  
  
 *// setting the correct*  
 *// ans of question*  
**public void** setAnswerTrue(**boolean** answerTrue)  
 {  
 **this**.**answerTrue** = answerTrue;  
 }  
  
}

.

Step4: Working with the String.xml file

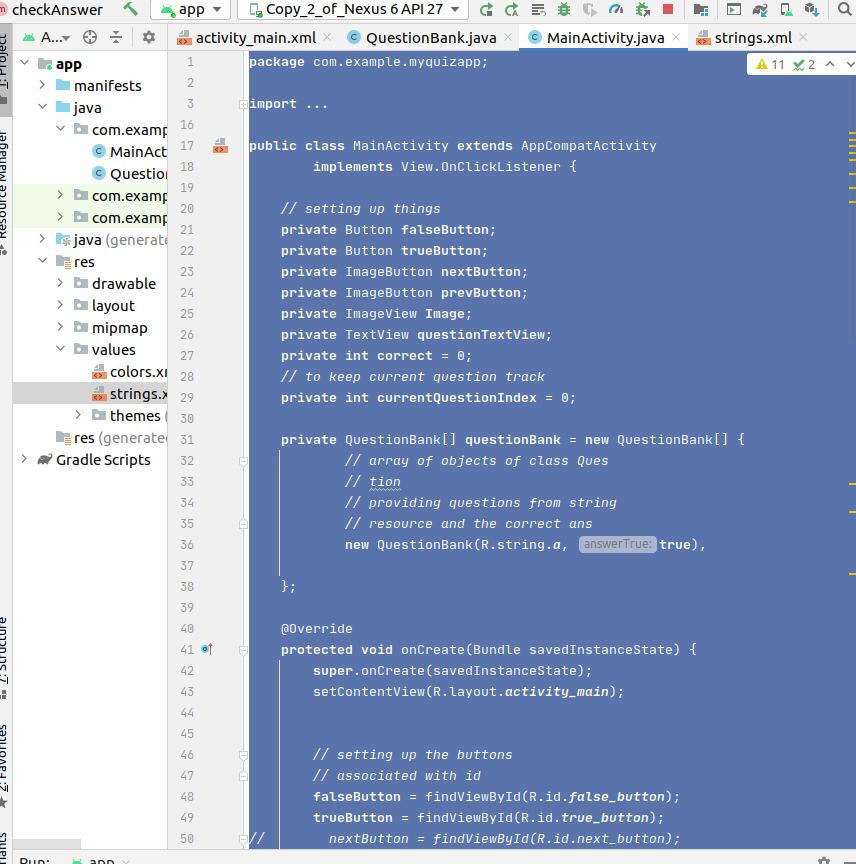
where we provide Questions for Judge

<**resources**>  
 *<!--All the string resources come here including Questions and title -->*  
<**string name="app\_name"**>Judging Question</**string**>  
 <**string name="true\_text"**>true</**string**>  
 <**string name="false\_text"**>false</**string**>  
 <**string name="correct\_answer"**>That\'s correct</**string**>  
 <**string name="wrong\_answer"**>That\'s incorrect</**string**>  
 <**string name="a"**>is Canbera Capital of Australia ?</**string**>  
  
  
</**resources**>

.

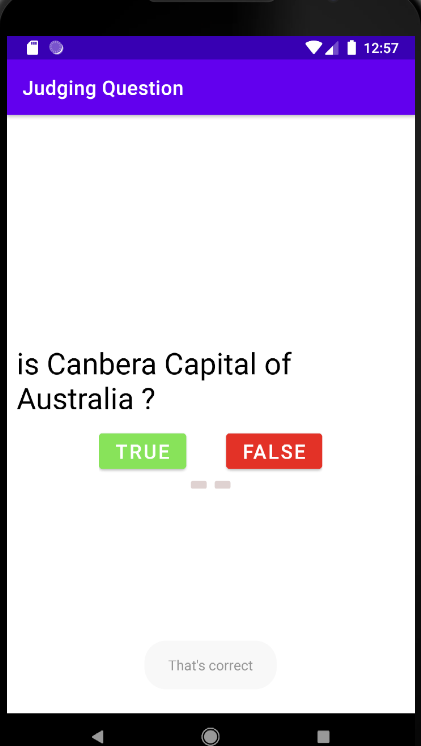
step5: Working with the MainActivity.java

**package** com.example.myquizapp;  
  
**import** androidx.annotation.RequiresApi;  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** android.annotation.SuppressLint;  
**import** android.os.Build;  
**import** android.os.Bundle;  
**import** android.util.Log;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.ImageButton;  
**import** android.widget.ImageView;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**public class** MainActivity **extends** AppCompatActivity  
 **implements** View.OnClickListener {  
  
 *// setting up things*  
**private** Button **falseButton**;  
 **private** Button **trueButton**;  
 **private** ImageButton **nextButton**;  
 **private** ImageButton **prevButton**;  
 **private** ImageView **Image**;  
 **private** TextView **questionTextView**;  
 **private int correct** = 0;  
 *// to keep current question track*  
**private int currentQuestionIndex** = 0;  
  
 **private** QuestionBank[] **questionBank** = **new** QuestionBank[] {  
 *// array of objects of class Ques*  
 *// tion*  
 *// providing questions from string*  
 *// resource and the correct ans*  
**new** QuestionBank(R.string.***a***, **true**),  
  
 };  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
  
 *// setting up the buttons*  
 *// associated with id*  
**falseButton** = findViewById(R.id.***false\_button***);  
 **trueButton** = findViewById(R.id.***true\_button***);  
*// nextButton = findViewById(R.id.next\_button);*  
*// prevButton = findViewById(R.id.prev\_button);*  
 *// register our buttons to listen to*  
 *// click events*  
**questionTextView**  
= findViewById(R.id.***answer\_text\_view***);  
 *//Image = findViewById(R.id.myimage);*  
**falseButton**.setOnClickListener(**this**);  
 **trueButton**.setOnClickListener(**this**);  
*// nextButton.setOnClickListener(this);*  
*// prevButton.setOnClickListener(this);*  
}  
  
 @SuppressLint(**"SetTextI18n"**)  
 @RequiresApi(api = Build.VERSION\_CODES.***LOLLIPOP***)  
 @Override  
 **public void** onClick(View v)  
 {  
 *// checking which button is*  
 *// clicked by user*  
 *// in this case user choose false*  
**switch** (v.getId()) {  
 **case** R.id.***false\_button***:  
 checkAnswer(**false**);  
 **break**;  
  
 **case** R.id.***true\_button***:  
 checkAnswer(**true**);  
 **break**;  
  
  
 }  
 }  
  
 @RequiresApi(api = Build.VERSION\_CODES.***LOLLIPOP***)  
 **private void** updateQuestion() {  
 Log.*d*(**"Current"**,  
 **"onClick: "** + **currentQuestionIndex**);  
  
 **questionTextView**.setText(  
 **questionBank**[**currentQuestionIndex**]  
 .getAnswerResId());  
 }  
  
 **private void** checkAnswer(**boolean** userChooseCorrect)  
 {  
 **boolean** answerIsTrue  
 = **questionBank**[**currentQuestionIndex**]  
 .isAnswerTrue();  
 *// getting correct ans of current question*  
**int** toastMessageId;  
 *// if ans matches with the*  
 *// button clicked*  
  
**if** (userChooseCorrect == answerIsTrue) {  
 toastMessageId = R.string.***correct\_answer***;  
 **correct**++;  
 }  
 **else** {  
 *// showing toast*  
 *// message correct*  
toastMessageId = R.string.***wrong\_answer***;  
 }  
  
 Toast  
 .*makeText*(MainActivity.**this**, toastMessageId,  
 Toast.***LENGTH\_SHORT***)  
 .show();  
 }  
}

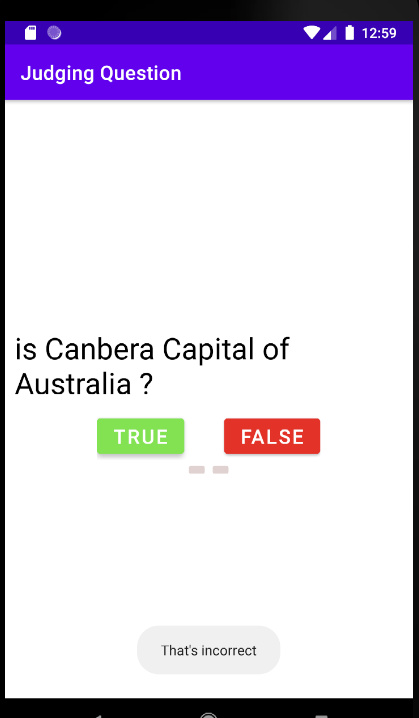
.

## 四，实验结果

If the answer is true:

.

if the answer is false:



.