5COSC023W - Tutorial 10 Exercises

As part of the tutorial for this week, you should complete **ALL** the tasks described in the following specifications: (**make sure that you ask questions to your tutor for anything that you do not understand or if you are stuck at any point**).

> Tutorial sessions are practical sessions that you need to work towards the exercises set. They will give you the chance to practice the material learned in the lectures and learn new things as well.
>
> **You should not use these sessions to work towards the assessed assignments!** If you decide to work towards your assessed work instead, then you are not considered as part of the tutorial session. You will not get any help on the code of the assessed work by your tutor but you can ask your tutor ONLY about any clarifications you might need regarding the specification of the coursework.

Like all other modules, you are expected to study towards you module outside the lecture and dedicated tutorial slots for a number of hours. If you do not finish all of the exercises in the tutorial session, make sure that you finish them on your own time and by the end of the week. This is a normal process and part of your university learning.

For all the tasks you should use Jetpack Compose and NOT Views!

1 Shopping Database Example

Implement the database example (keeping track of products for supermarket shopping) that we have seen in the lecture using the Room library.

Implement the simple version using the `GUI` composable having 2 button for saving a product and retrieving all products from the database.

2 Extending the Database Example

Implement the extension version of the program having checkboxes to select the products the user would like to purchase and email to themselves. Implement the `GUIExtendedFunctionality()` function and call it inside the main activity instead of calling the `GUI`.

Try to understand every single line of code and ask your tutor if you do not understand something or you would like to go over the entire example explaining the code to you.

3 A Quiz App

Implement an Android application which asks the user a question and whether this question evaluates to **True** or **False**.

All the questions should come from an SQLite database using the Room library. You might would like to add additional textboxes and buttons to insert new questions and the correct answers to the database. These could be in the same or a separate activity.

A random question is chosen from the database and presented to the user.

Typical questions could be:

1. Chelsea won the Premier League title in 2023–2024? (false)
2. Buenos Aires is the capital of Argentina? (true)
3. Newcastle has won the Premier League in the past? (true)
4. Zurich is the capital of Switzerland? (false)

The main screen (activity) of the application should look like Figure 1.

Every time the user presses the "True" or "False" buttons a `Toast` screen should appear with the text "Correct!" or "Incorrect!" respectively. The user moves to the next question by pressing the `Next` button and as soon as all the questions are exhausted the user is presented with the first question again.

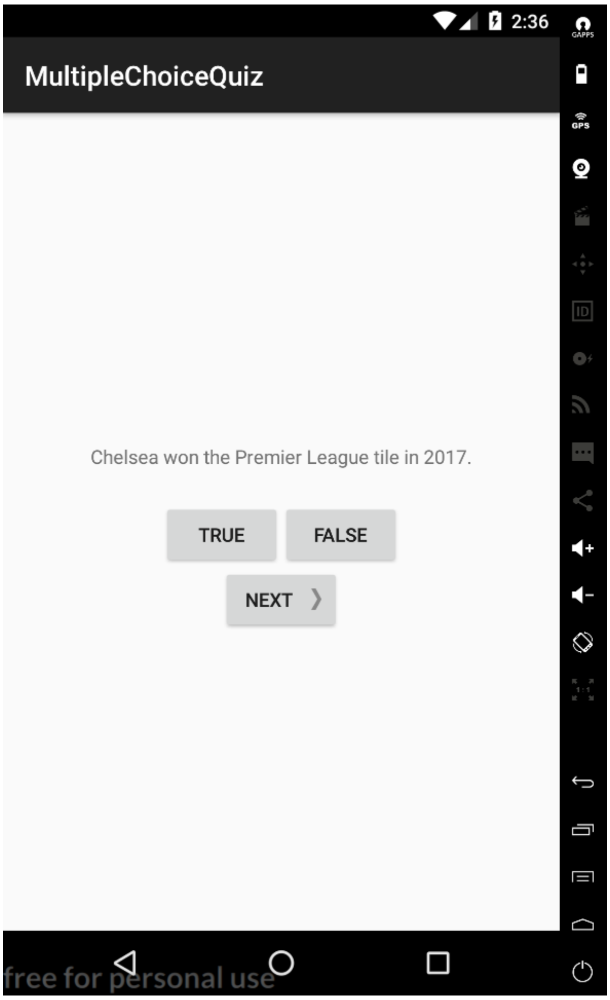
4 Extending the Quiz app with a second activity

Extend the **Quiz** app by adding another button to the main activity called the `Cheat` button as in Figure 2.

When the button is pressed a new activity `CheatActivity` will start and its layout will be as shown in Figure 3. Pressing the "Show Answer" button will display the correct answer for the corresponding question and then the cheater user can press the back button to go to the main screen and answer the question.

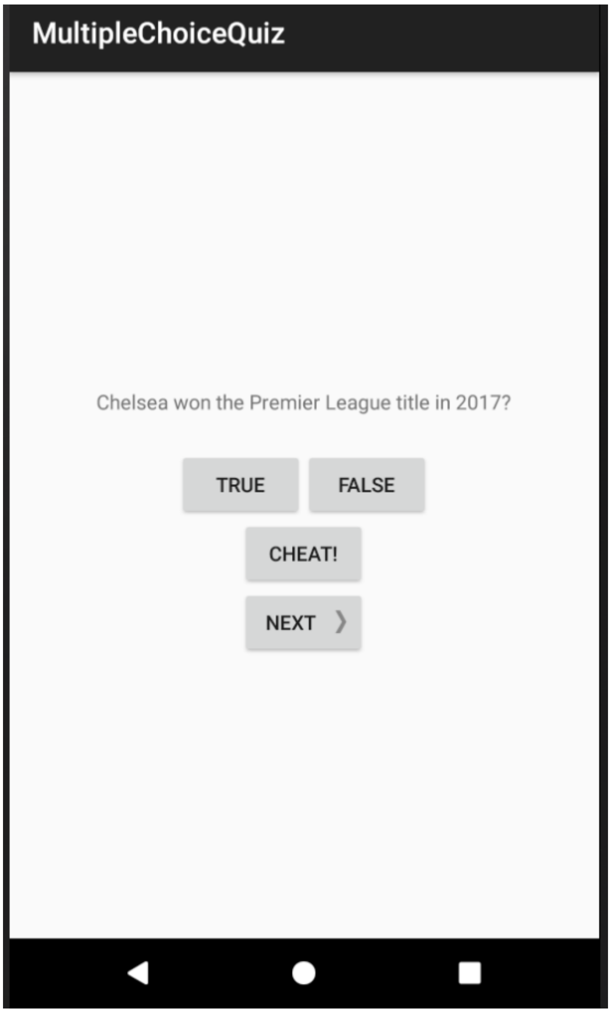
- Using bidirectional communication between 2 activities a new activity can return back a result to the activity that created the second activity (see a previous tutorial exercise on how to do this).

  Using the approach described above, extend the **Quiz** application so that if the user presses the `ShowAnswer` button in the `Cheat` activity and then returns to the main `Quiz` activity and answers that same question (that the answer was revealed in the cheat activity) the toast pop-up message will display "You cheated! Shame on you!" instead of a "Correct" or "Incorrect" message.



Chelsea won the Premier League tile in 2017.

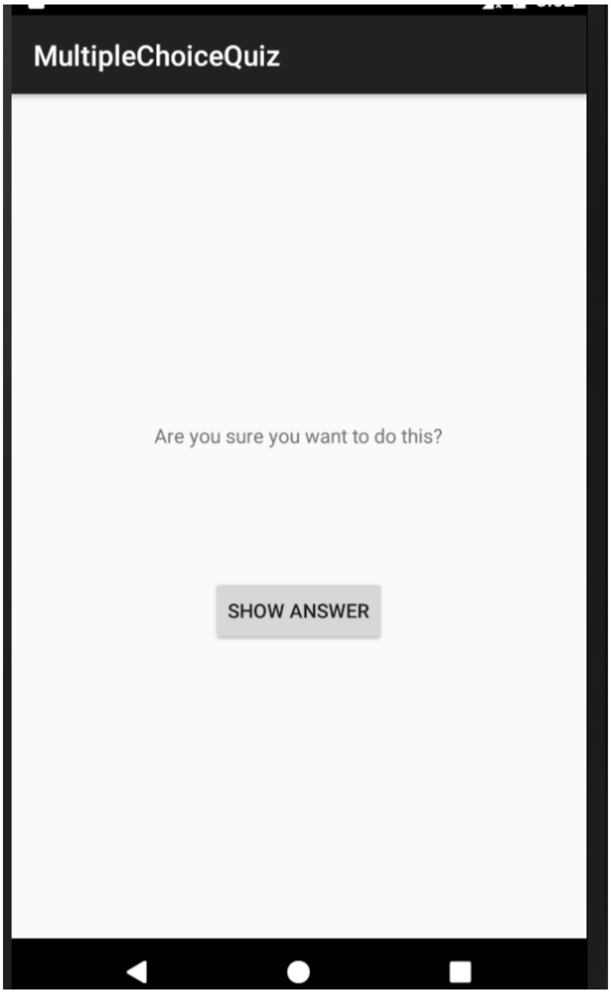
Figure 1: The main activity screen of the Quiz app.



MultipleChoiceQuiz

Chelsea won the Premier League title in 2017?

Figure 2: Adding a cheat button in the Quiz app.



MultipleChoiceQuiz

Are you sure you want to do this?

Figure 3: The *Cheat* activity screen of the Quiz app.

Extending the Quiz app with Radio buttons

Modify the quiz app so that instead of 2 buttons “True” and “False” it uses 2 radio buttons with the labels “True” and “False”. Every time the user makes a selection on a radio button, the “Correct” or “Incorect” message appears.

Hint: An example of how to use radio buttons can be found in:

https://developer.android.com/develop/ui/compose/components/radio-button