**Classes that you need to copy as they are (without worrying about the implementation) :**

1. AsyncTask
2. Continents
3. Neighbours
4. QuizResults
5. DBHelper
6. DataManager

**That just leaves :**

1. Main Activity

**Some Important points:**

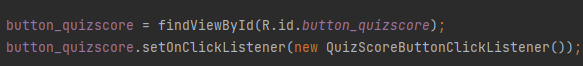
**MainActivity :**

* Everything is happening inside onCreate(). Outside it there are just function and nested class definitions.
* **Imp :** I haven’t implemented onPause() and onResume(), since they weren’t required for the DB part. But professor has asked to implement them such that, if a user stops the app in between a quiz, you need to store everything (the quiz questions he gave, how many he answered, his/her score so far. I have added it in the ‘Handling interrupts’ part of the Excel sheet).
* **Classes** **ContinentsTableWriter and NeighboursTableWriter :**
  + So here ContinentsTableWriter and NeighboursTableWriter are classes, extending the Async task. We’re creating their objects and calling their execute().
  + The execute methods inside them are creating the ‘Continents’ and ‘Neighbours’ tables, using the two csv files :

**A black background with white text

Description automatically generated**

* + **Ignore :** How the classes and execute work.
  + **Use :** Both above execute calls at the launch of your app, as they are going to create the Continents and Neighbours tables and populate them.
* **Classes QuizResultsTableWriter :**
  + Similar to above, this one also extends Async task. It’s execute method is designed to create the QuizResults table.
  + However, the execute() of this one is not called just in onCreate(). It’s tied to the ‘Save quiz score’ button.
  + In our final app, it won’t be tied to any button. Once the user answers all 6 que of quiz, you should call this method.
  + In my current app, this is how this method is getting called :

****

**A computer screen shot of a program code

Description automatically generated**

* + Just like previous case, **Ignore :** How the QuizResultsTableWriter class and it’s execute() works.
  + **Use :** The execute method at an appropriate place. Note that I’m passing just the quiz score. The function has been designed to auto calculate QuizID and the QuizDate and along with the Quiz score that you pass, add everything into table ‘QuizResults’.
* **Class ContinentsTableReader :**
  + As you might’ve expected, this one also has an execute method, that fetches data from Continents Table.
  + I’ve tied it’s execute method to the ‘Retrieve Continents Table Data’ button.
  + Here’s the flow to execute() :

**A screen shot of a computer

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

* + **Imp :** Here, you’ll need to modify the execute method a bit. Because my execute() method, fetched the continents table data and displays it on the screen as well. We don’t want that. We just need the array of ‘Continents’ objects. Each object = One row of the Continents table. This is what you’ll need to do :
    - Whenever execute() is called, it leads to execution of doInBackground() and onPostExecute(). You won’t find execute() method’s definition explicitly in the ContinentsTableReader class (An inheritor of the Async class). That’s how ASYNC works.
    - **doInBackground()** is good, no changes needed. The ‘arr\_Continents’ is passed to the next method onPostExecute() by the system (Again that’s how Async works).A screen shot of a computer program

      Description automatically generated
    - **onPostExecute()** Here, the for loop is being used for displaying all the contents of the Continents table onto the screen. Just delete the entire for loop.

**A screen shot of a computer

Description automatically generated**

* + - The array I’ve circled in green, is the one which will contain the entire data of the Continents table. It’ll be an array of ‘Continents’ class objects. Each object is going to store one row of the Continents table.
    - **Imp :** At the end of the for loop, there’s this statement which is going to close the database. DO NOT forget to add it in your code.

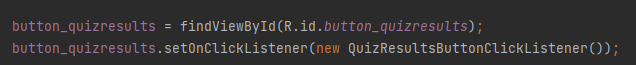
****

* + - So to summarize in the **ContinentsTableReader class** :
      * **Ignore :** How the QuizResultsTableWriter class and it’s execute() works.
      * **Modify :** It’s onPostExecute() method.
      * **Use :** The arr\_Continents array that you get at the end of onPostExecute(). That’s what contains the complete data of the Continents table.
* **Class NeighboursTableReader :**
  + It has been structured in the exact same way for ContinentsTableReader.
  + So do the same steps as mentioned above for Class ContinentsTableReader.
  + You’’ have to modify it’s onPostExecute() method as well.
  + **Use :** The arr\_Neighbours array that you get at the end of onPostExecute(). That’s what contains the complete data of the Neighbours table.

**A screen shot of a computer code

Description automatically generated**

* **Class QuizResultsTableReader :**
  + Just like Continents and Neighbours, this one also has an execute method, which fetches data from the QuizResults Table.
  + I have tied it to the ‘Retrieve Quiz Results’ button.
  + In our final app, you’ll also have to tie it to a button for fetching quiz results. So you can basically use my entire code for this one as it is.
  + Here’s the flow :



A screen shot of a computer program

Description automatically generated

* + Just like earlier, the execute method calls, doInBackground and onPostExecute.
  + It also has the ‘for’ loop that you had to erase in the previous cases, but in this case you can keep it, as we need to display the quiz results.

A screenshot of a computer program

Description automatically generated

* + You’ll only need to modify the for loop a bit, as in the QuizResults table that it displays, the last two columns (Date and Score) are too close to each other.

A screenshot of a quiz

Description automatically generated

* Take care of the below calls, they actually open and close the database.





* You’ll also have to create the assets folder manually and ass the csv files in it :

A screenshot of a computer

Description automatically generated

* **Imp :** Also, we have to manually add this dependency for being able to read csv files :

A screen shot of a computer

Description automatically generated

* You can view contents of databases created by your app in your app inspection window shown below. But you’ll need to comment out the below line in the last function that’s going to use your DB. If DB is closed, App Inspection won’t display the contents of our app’s DB.



**A screenshot of a computer

Description automatically generated**

* Hope this all helps! :)