

# Project 1C - MultiQuiz Code

## 30 Possible Points

2/15/2022

Attempt 1

**IN PROGRESS**

Next Up: Submit Assignment



Add Comment

0

**Unlimited Attempts Allowed**

### Details

**Due:** Tuesday, February 15**Points:** 30**Resources:**

- **Install these:**

- **build.gradle** ([https://drive.google.com/file/d/1BohleQ7XEQywcc\\_O\\_ffMhr9zjQIW5q1/view?usp=sharing](https://drive.google.com/file/d/1BohleQ7XEQywcc_O_ffMhr9zjQIW5q1/view?usp=sharing)) (Module: MultiQuiz.app)
- **BaseQuizActivityTest** ([https://drive.google.com/file/d/19eibOpzVhwbXBh6FpR5DLTlzeWGe\\_5Kg/view?usp=sharing](https://drive.google.com/file/d/19eibOpzVhwbXBh6FpR5DLTlzeWGe_5Kg/view?usp=sharing))
- **OrientationChangeAction** ([https://drive.google.com/file/d/1jDnALONtxunTYH89bV\\_GkxJsCt9aVqd4/view?usp=sharing](https://drive.google.com/file/d/1jDnALONtxunTYH89bV_GkxJsCt9aVqd4/view?usp=sharing))

- **Check these:**

- **AndroidManifest.xml** ([https://drive.google.com/file/d/1PpFenBJ2ufdxVhIRxIPT3Y3E8SWh\\_HCY/view?usp=sharing](https://drive.google.com/file/d/1PpFenBJ2ufdxVhIRxIPT3Y3E8SWh_HCY/view?usp=sharing))
- **build.gradle** (<https://drive.google.com/file/d/1WeyyZFZcb-aY9RaTGX3dhmfj1qCVydMt/view?usp=sharing>) (Project: MultiQuiz)
- **gradle-wrapper.properties** (<https://drive.google.com/file/d/123oMFa3JZG8LNy1WmyDGan36NcIMD5CS/view?usp=sharing>) (Gradle version should be 7.2)

**Deliverables:**

- ZIP file containing
  - Entire project

## Overview

Modify the MultiQuiz that you created in **Project 1B**<https://canvas.vt.edu/courses/145416/assignments/1362967>). For P1B, you should have completed[Submit Assignment](#)

Step 4 involves creating the view-model for the main activity so that you can rotate the screen, and step 5 involves creating a second activity that shows the results of taking the quiz.

For step 4, focus on section "[Adding a ViewModel](https://learning.oreilly.com/library/view/android-programming-the/9780135257555/ch04s02.html) (<https://learning.oreilly.com/library/view/android-programming-the/9780135257555/ch04s02.html>)" in chapter 4 of BNR. To access the view-model from the activity, use the following code:

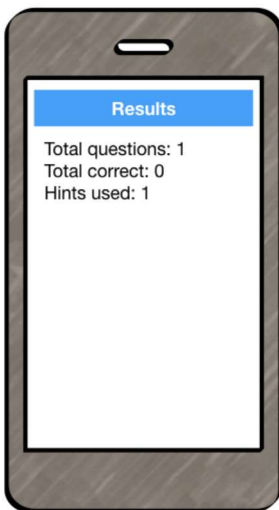
```
private val vm: QuizViewModel by lazy {  
    ViewModelProvider(this).get(QuizViewModel::class.java)  
}
```

The code that BNR gives for this is deprecated.

For step 5, create a constraint layout with **7** text views:

- A heading: Result
- Labels total questions, correct answers, and hints used
- Values of total questions, correct answers, and hints used. The values **must** have the following ids:
  - total\_questions\_value
  - total\_answers\_correct\_value
  - total\_hints\_used\_value

The text views should be arranged similar to the image below (you do **not** need a blue background for the heading).



As an example, here is text-view code for the total questions label and the total questions value:

```
<TextView  
    android:id="@+id/total_questions_label"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="8dp"
```

Submit Assignment

```
<TextView
    android:id="@+id/total_questions_value"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:text="@string/total_questions_value"
    android:textSize="18sp"
    app:layout_constraintBottom_toBottomOf="@+id/total_questions_label"
    app:layout_constraintStart_toEndOf="@+id/total_questions_label"/>
```

The results activity will be reached by clicking the "submit" button from the quiz activity after all four quiz questions have been answered. To pass data from the quiz activity to the results activity, see the section "[Passing Data Between Activities](https://learning.oreilly.com/library/view/android-programming-the/9780135257555/ch06s03.html) (<https://learning.oreilly.com/library/view/android-programming-the/9780135257555/ch06s03.html>)" from chapter 6 of BNR. While you should understand the concepts in the subsection "Getting a Result Back from a Child Activity", you will not need to pass any results back in this project.

## Testing MultiQuiz

Implement MultiQuiz by modifying the code from the AnswerButton app. As you evolve AnswerButton into MultiQuiz, make sure you can meet the following

## Naming and Other Project Requirements

In order for our tests to work correctly, you **must** meet the following naming requirements.

- Your application package name must be "edu.vt.cs.cs5254.multiquiz"
- Your application name must be "MultiQuiz"
- Your main activity must be named "QuizActivity" (do not use MainActivity)
- Your second activity must be named "ResultsActivity" (note that "Results" is plural)
- In activity\_quiz.xml, the IDs for your buttons must be:
  - *answer\_0\_button*
  - *answer\_1\_button*
  - *answer\_2\_button*
  - *answer\_3\_button*
  - *hint\_button*
  - *submit\_button*
- In activity\_results.xml, the IDs for the text-views that hold your values must be:
  - *total\_questions\_value*
  - *total\_answers\_correct\_value*
  - *total\_hints\_used\_value*

You must use the following files in your project. You may not alter these files in any way. If something

Submit Assignment

- **build.gradle** ([https://drive.google.com/file/d/1BohleQ7XEQywcc\\_O\\_ffMhr9zjQIW05q1/view?usp=sharing](https://drive.google.com/file/d/1BohleQ7XEQywcc_O_ffMhr9zjQIW05q1/view?usp=sharing)) (Module: MultiQuiz.app)
- **BaseQuizActivityTest** ([https://drive.google.com/file/d/19eibOpzVhwbXBh6FpR5DLTlzeWGe\\_5Kg/view?usp=sharing](https://drive.google.com/file/d/19eibOpzVhwbXBh6FpR5DLTlzeWGe_5Kg/view?usp=sharing))
- **OrientationChangeAction** ([https://drive.google.com/file/d/1jDnALONtxunTYH89bV\\_GkxJsCt9aVqd4/view?usp=sharing](https://drive.google.com/file/d/1jDnALONtxunTYH89bV_GkxJsCt9aVqd4/view?usp=sharing))

In addition, check the code in your project against the following files to ensure that it is the same. When in doubt, use the code here.

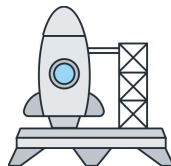
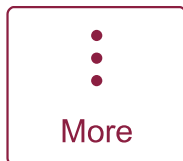
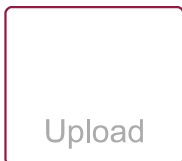
- AndroidManifest.xml
- build.gradle (Project: MultiQuiz)
- gradle-wrapper.properties (Gradle Version)

Finally, your application should run on the following emulator:

- Pixel 2 API 30

When using the base test cases to test your code, you must use a Pixel 2 emulator running API 30. This is what we will use for the final test suite.

### Choose a submission type



Choose a file to upload  
File permitted: ZIP

Submit Assignment