



[< Back to Android Basics Nanodegree by Google](#)

Quiz App

REVIEW

CODE REVIEW 10

HISTORY

Meets Specifications

Hi !

Awesome job here,Your app is more than wonderful. I am very happy to have reviewed your project. You should be proud of yourself, if I can rating you I will give you Definitely the highest rating. Keep it up and you will have a great place in the future.

If you have any feedback regarding the review, you can put it in the feedback section.

Recharge the batteries, and get ready for the next challenges!
Keep up the good work!

Layout

App contains 4 - 10 questions, including at least one check box, one radio button, and one text entry.

Questions are in a variety of formats such as free text response, checkboxes, and radio buttons.

Checkboxes are only used for questions with multiple right answers. Radio buttons are only used for questions with a single right answer.

App includes a button for the user to submit their answers and receive a score.

The code adheres to all of the following best practices:

- Text sizes are defined in sp
- Lengths are defined in dp
- Padding and margin is used appropriately, such that the views are not crammed up against each other.

The app includes at least four of the following Views: TextView, ImageView, Button, Checkbox, EditText, LinearLayout, RelativeLayout, ScrollView, RadioButton, RadioGroup.

If applicable, the app uses nested ViewGroups to reduce the complexity of the layout.

The app gracefully handles displaying all the content on screen when rotated. Either by updating the layout, adding a scrollable feature or some other mechanism that adheres to Android development guidelines.

Functionality

The code runs without errors.

Each question has a correct answer.

Any question which uses radio buttons allows only one to be checked at once.

The app contains at least one if/else statement

The grading button displays a toast which accurately displays the results of the quiz.

The grading logic checks each answer correctly. The app accurately calculates the number of correct answers and does not include incorrect answers in the count.

Note: When applicable, in the grading logic remember to check that the correct answers are checked AND the incorrect answers are not checked.

Code Readability

All variables, methods, and resource IDs are descriptively named such that another developer reading the code can easily understand their function.

The code is properly formatted i.e. there are no unnecessary blank lines; there are no unused variables or methods; there is no commented out code.

The code also has proper indentation when defining variables and methods.

 [DOWNLOAD PROJECT](#)

10

[CODE REVIEW COMMENTS](#)



[RETURN TO PATH](#)