Court Counter App

Project Overview:

This project is a chance for you to combine and practice everything you learned in this section of the Diploma program. You will be making an app that allows a user to keep track of points within a game.

The goal is to create a Scorekeeper app which gives a user the ability to keep track of the score of two different teams playing a game of your choice. To build this project, you can follow along with the practice set and customize the Court Counter app to track scores from a different sport.

Why this project?

In the most recent portion of the Diploma program, you learned how to make use of Java in your Android apps to make them interactive. With Score Keeper, you will be practicing this skill set, with particular focus on button code, variable scope, and updating views.

What will I learn?

This project is about combining various ideas and skills we've been practicing throughout the course. They include:

- Adding button code to your app
- Updating views
- Properly scoping variables
- Finding views by their ID

Build Your Project:

To complete this project, you will need to build a Scorekeeper app. You'll need to select a sport with either multiple ways to score points or multiple metrics to track. If you have trouble choosing, here are some suggestions:

- **Baseball Scores App:** Track both runs and outs
- **American Football Scores App:** Add a button for each of the 4 amounts of points a team can score in american football.
- **Football Scores app:** Track both points and fouls

Your project will be evaluated using the Scorekeeper project rubric shown below.

Project Rubric:

Content Review

CRITERIA	MEETS SPECIFICATIONS
Game Chosen	The chosen game has either multiple amounts of points that can be scored, as in american football, or multiple important metrics to track, such as fouls, outs, and innings in baseball.
Overall layout	App is divided into two columns, one for each team.
Column contents	Each column contains a large TextView to keep track of the current score for that team. Optionally, a second TextView to track another important metric such as fouls can be added.
Score buttons	Each column contains multiple buttons. The buttons must track either: Each track a different kind of scoring

	Or
	Each track a different metric (one score, the other fouls, for instance).
Reset button	The layout contains a 'reset' button.
Best practices	"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."

Functionality

CRITERIA	MEETS SPECIFICATIONS
Errors	The code runs without errors.
Score Button Function	Each score button updates the score TextView in its column by adding the correct number of points.
Reset Button Function	The reset button resets the scores on both of the score TextViews.

Code Readability

CRITERIA	MEETS SPECIFICATIONS
Naming conventions	"Any classes are named after the object they represent. All variables are named by their intended contents. All methods are named by their intended effect or in the

	style required by a callback interface."
Code style	"There are no unnecessary blank lines. One variable is declared per declaration line. The code within a method is indented with respect to the method declaration line."

Prepare for Submission:

Clean Your Build

Before submitting, please follow the instructions for cleaning your project files. This removes some temporary files and greatly decreases the size of your project.

[Clean Your Project Files](#)

Final Submission Checklist

Before submitting your project for evaluation, we recommend that you check that each of the following is true:

1. Your app compiles and runs as expected.
2. You are proud of your app and its output.
3. You completed this project according to instructions.
4. You cleaned the project using the instructions above.
5. You checked your project against the rubric.