Homework 05 – Listener Madness

Due: Day 21 – March 21st, 2022 (beginning of class)

Objective:

The goal of this assignment is for you to demonstrate your understanding of how to setup Listeners for different Views and learn about a new Android class: the CountDownTimer class

Instructions:

Since it is March Madness, it only seems appropriate that we create a basketball-themed, score-tracking app. Assume that you have been hired by the NCAA to create a new single-activity app, **BucketTracker**, to be used by NCAA officials for tracking points scored and time remaining on the game clock. On Moodle, I have provided a starting Android project that I want you to use for implementing the required features described below. **Important**: The Views in the layout file have already been given an appropriate ID name for this homework to model the types of id names that I want you to create in future assessments/assignments. Therefore, do not add/modify/remove any of the Views in the layout file that I provided you. You will need to setup a View Binding in your Java controller file.

Your app needs to work as follows (i.e., the features/behavior listed below will be explicitly graded for this assignment). Be sure to start on this assignment early – the top portion will be easier to implement, however, the bottom portion will take more time – as a result, the bottom portion is worth a larger percentage of this assignment's total points:

- **Top Portion (Scoretracker)**
  - The official can select any combination of the number of point options presented as checkboxes.
  - The official can select which team has possession of the ball by pressing the toggle button. Whichever team is currently selected on the toggle button should have their name (i.e., Home vs Guest) and current score color in red, otherwise their name and score color should be black.
  - When the official long presses the Add to Score button, the combination of points that they have selected should be added to the selected team's current score.
    - When you retrieve data from a View in Android, it will most often be as a String object. A useful static method called the **valueOf**( ) method can be found in the Boolean, Byte, Short, Integer, Long, Float, and Double classes. This method takes a String as an input and returns its corresponding value as the class (e.g., **int temp = Integer.valueOf("428");** will convert the String "428" to the integer 428)
    - Similarly, to display data in a View, it will most often need to be converted to a String object. A useful static method called the **valueOf**( ) method can be found in the String class. This method can take a boolean, byte, short, int, long, float, or double value as an input and converts it into a String (i.e., text). For example, **String temp = String.valueOf(428)** will convert the int 428 to the String "428"
  - After the points have been added, all of the point options should be de-selected and the opposite team should be selected/colored as having possession of the ball.
- **Bottom Portion (Game Clock)**
  - The game clock will begin at 20:00 (i.e., 20 minutes) which is the length of each half of a NCAA basketball game
  - The game clock will not begin counting down until the official uses the switch to turn the countdown on
  - Once activated, the game clock will count down in 1-second increments (i.e., 20:00 … 19:59 … 19:58 … etc)
  - If the official turns the switch to the off position, the game clock will pause at the current time remaining

- If the official turns the switch to the on position, the game clock will resume its countdown from the current time remaining
- If the official needs to reset the game clock to a different remaining time, they will enter the number of minutes and seconds, that they wish to appear on the game clock using the EditTexts. When the official presses the Set Time button, the game clock's time remaining should be set to the number of minutes and seconds the official entered. Also, the game clock should be paused and will require the official to re-activate it using the switch
  - Note: Be sure to perform error checking to ensure that the official entered valid values for the number of minutes and seconds, however, you are not permitted to use try-catch statements to catch exceptions, this will be worth 0% credit as these are costly for a computer to execute.
- For this portion of the assignment, I want you to use Android's CountDownTimer class (https://developer.android.com/reference/android/os/CountDownTimer)
- As you were taught in CSC232, having "magic numbers" appear in your code is not acceptable in professional development. Therefore, you should define and use the following static constant variables in your Activity's class:
  - **DEFAULT_NUM_MINS** - stores the number of minutes for a NCAA basketball half
  - **MILLIS_PER_MIN** - stores the number of milliseconds in 1 minute
  - **MILLIS_PER_SEC** - stores the number of milliseconds in 1 second
  - **SECS_PER_MIN** - stores the number of seconds in 1 minute
- You must create a method in your Activity's class named **getNewTimer** that takes two parameters: (1) the timer's total length in milliseconds represented as a **long** and (2) the timer's tick length in milliseconds also represented as a **long**. This method's **return type** should be a **CountDownTimer** object. You should call/use this method any time that you need to construct a new CountDownTimer object in your code.
  - Tip #1: You will need to create a private member variable (i.e., field) in your activity's class to store a reference to a CountDownTimer object
  - Tip #2: As you will notice, there is a method to **start( )** and **cancel( )** a CountDownTimer once it has been created but there is no pause/stop method. Therefore, when the game clock is paused, you will need to record the amount of time remaining … when the game clock is restarted, you will need to (a) **cancel** the previous CountDownTimer object and (b) construct/start a new CountDownTimer object.

Submission:

When you are finished, you must **zip** your Android Studio project. On Moodle, you should upload your zip file to the Homework 05 assignment box