CSC 214 ASSIGNMENT 09

Note: Cheating is still against the rules. Don't.

This is new, you should read it: The goal of this assignment is to implement a multithreaded application using both AsyncTask and HandlerThread.

Assignment Instructions

1. Each part of this assignment will be implemented using a separate activity. The Main Activity should include a button used to start each of the other activities.

2. Your second activity should feature 4 widgets:
    a. An `EditText` into which a user may type any valid long (64-bit integer) that is greater than or equal to 2. If the user types an invalid number, pop up a `Toast` correcting them.
    b. A button that, when pressed, begins a new `AsyncTask` that calculates the largest prime number that is less than or equal to the number that the user entered using a brute force algorithm. See this page for ideas, but do not attempt to use the Sieve of Eratosthenes; the goal here is for the algorithm to take some time to run: be purposely inefficient.
    c. A button that starts an `AsyncTask` to calculate the square root of the number that the user entered.
    d. A `TextView` that is used to display the result of the calculation performed by either button. Whenever the `TextView` is updated, a `Toast` Should also display indicating the kind of result (prime number or square root).

    Try calculating the largest prime for a very large number and then trying to calculate the square root at the same time. Explain what happens (and why) in your README.

3. Your third activity should duplicate the first, but should use two `HandlerThread`s rather than `AsyncTask`s. One `HandlerThread` should calculate prime numbers, and the other should calculate square roots. Try calculating the largest prime for a very large number and then trying to calculate the square root at the same time. Explain what happens (and why) in your README.

4. Your fourth activity should include 3 widgets:
    a. An `ImageView` that displays a default image of your choice.
    b. An `EditText` that allows the user to type in an image URL (HTTP). You may pre-populate the `EditText` for testing.

c. A button that, when pressed, will fetch the image at the specified URL and display it in the `ImageView`. The button should be disabled while the image is downloading to prevent spamming.

In your README explain what technique you chose to implement the solution and why. "Because it was easier" is not a valid reason.

HAND IN

Before handing in, create two additional files in your lab directory:

1. Create a README that contains the following:
    a. Your contact information, TA name, and assignment number.
    b. A brief (one paragraph at most) description of the assignment.
    c. The answers to the the questions in parts 2, 3, and 4.
2. Create a directory titled "SampleOutput." This directory should contain:
    a. A file called "logs.txt" that contains examples any relevant logs generated by your application. Remember to use LogCat filters to show *only* your log messages before copying them into the file.
    b. Screenshots (in PNG or JPG format) taken from an Android Virtual Device that show examples of your application's user interface.

Hand in by uploading the compressed (i.e. "zipped") folder containing your Android Studio project and the required additional files to Blackboard. Remember that **late submissions are not accepted**.