

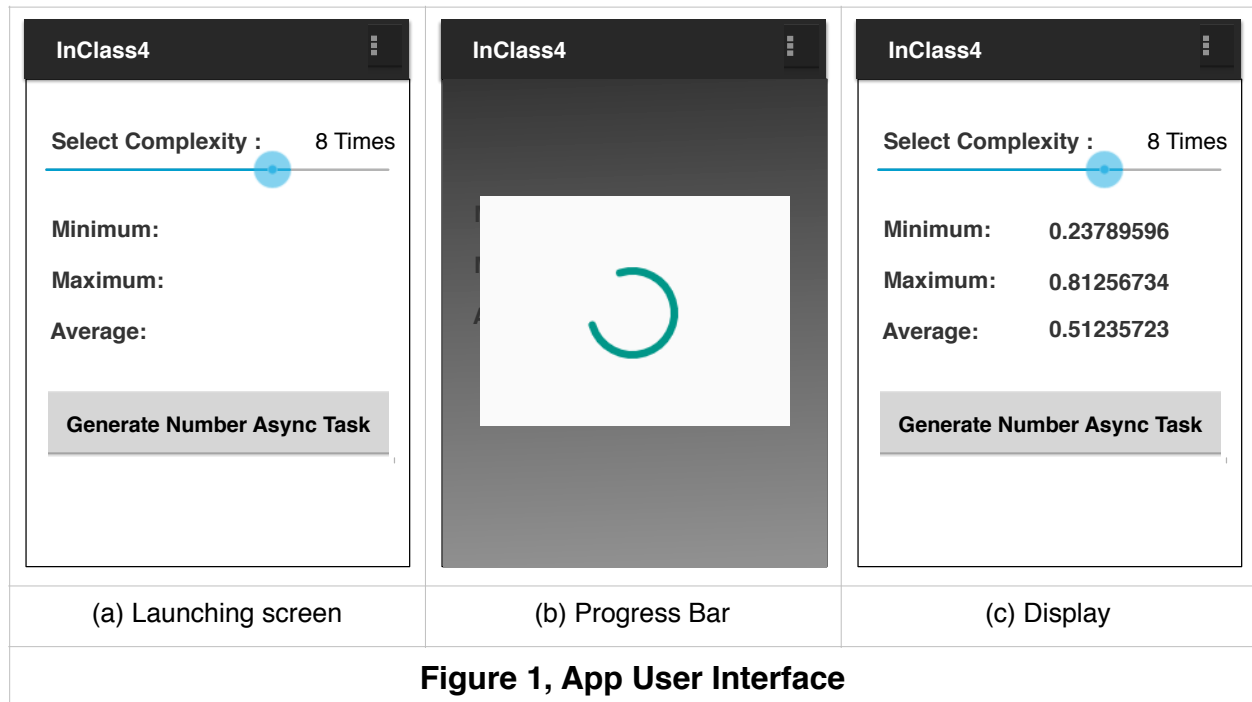
ITIS/ITCS 5180 Mobile Application Development
In Class Assignment 04

Basic Instructions:

1. Each team is required to submit the assignment on Canvas.
2. Please download the support files provided with this assignment and use them when implementing your project.
3. Submission details:
 - a. Compress the contents of your project folder. The file name is very important and should follow the following format: **InClass04.zip**
 - b. Only one group member is required to submit on behalf of the whole group.**
 - c. You should submit the assignment through Canvas: Submit the zip file.
4. The required Android Virtual Device (AVD) should have **minimum SDK version set to 25 and target SDK at 29.**
- 5. Failure to follow the above instructions will result in point deductions.**

In Class Assignment 4 (100 Points)

In this assignment you will get familiar with Android concurrency models. This



application is composed of a single activity: Main Activity.

The interface should be created to match the user interface (UI) presented in Figure 1. You will be using layout files, and strings.xml to create the user interface. Perform the following tasks:

1. Create a new android project called “In Class 4”.
2. You are provided with a HeavyWork class that contains a static method `getArrayNumbers(int n)`. This method takes a long time to execute. Import the provided Java file by simply dragging the file into the src folder under your project package.
3. Your task is to use an AsyncTask to execute this method in a background thread. Do not use the main thread to generate the numbers. The UI should be manipulated by the only main thread.
4. Use a SeekBar to set the complexity of the heavy work. The SeekBar maximum should be set to 10. Also note, the TextView showing the selected complexity number which is displayed to the right of the “Select Complexity” label, this number should be updated whenever the user moves the SeekBar. The selected **number** defines the number **n** of `getArrayNumbers(int n)` method in the background AsyncTask.
5. Tapping on the “Generate Number AsyncTask” button should start the execution of a background AsyncTask and compute the minimum, maximum, and average of all the numbers returned as an ArrayList by the `getArrayNumbers()` method. For example, if the complexity was set to 5, the `getArrayNumbers()` method will return an ArrayList of 5 numbers. The minimum, maximum, and average of these 5 numbers should be

computed and displayed in the TextViews. While these numbers are being generated, display a ProgressBar indicating the progress, see Figure1(b). The ProgressBar should be dismissed automatically after the computation is completed and the numbers generated and displayed in the TextViews, see Figure 1(c).

Homework part: Using Threads and Handlers

This part is similar to in class 04, but you should use threads and handlers to implement the same functionality. Perform the following tasks:

1. You should create a thread pool of size 2. Use the thread pool to execute the created thread.
2. Tapping on the “Generate Number Thread” (in stead of “Generate Number AsyncTask”) button should start the execution of a background thread and compute the ArrayList (based on the selected complexity) returned by the `getArrayNumbers(int n)` method, as done in the in-class assignment.
3. To be able to exchange messages between the child thread and the main thread use the Handler class. Either use messaging or setup a runnable message.