



## Lab 4 – Keeping the App Responsive

Topic: Threads and Async Tasks

Deadline and Submission: Check Sakai

**Submission Instruction:** (1) code – submit only one program that uses an Async Task, i.e., the version after step #6. Keep the Thread-based solution commented inside your program so that we can see. If the source code is too big, you should upload it to GitHub/etc and provide us the link using Sakai. Video/screenshot is not needed for this assignment. (2) Writing – You can write the answers to questions #5 and #7 inside the text box in Sakai, or upload a separate text/doc file in Sakai.

#	Task	Check
1	Create a function <code>void foo(int m)</code> that takes approximately “m” seconds to execute. It should print a line after every second, e.g., <code>Log.v("TAG", "xx seconds elapsed");</code>	
2	Create a button and the corresponding click event handler and call the <code>foo()</code> function with varying parameters 1, 2, 3, ... seconds, until the app is not responsive.	
3	Make the app responsive using a new <b>Thread</b> every time the button is clicked.	
4	Add a <code>textView</code> to your app and modify the <code>foo()</code> function to update its text every second, e.g., <code>textView.setText("mm seconds elapsed");</code>	
5	Does the Thread-based solution in #3 work anymore? Why or why not?	
6	Fix the problem by using an <b>Async Task</b> . (Hint: Implement the loop inside <code>doInBackground()</code> and use <code>onProgressUpdate()</code> to update the view)	
7	Is there any other alternative solutions that you can think of?	