

Task A10.1P: not breaking the rules

Thinking about concurrent tasks

Due: by Monday in week 12, for review ahead of week 12's lab

Aim

The aim of this task is for you to consider how Android handles threads and background processing.

Tasks

1. Some rules are not to be broken

You have been given some code to write a basic countdown timer that updates a TextView with the current value every second:

```
try {
    for (int i = 3; i >= 0; i--) {
        Thread.sleep(1000);
        status.setText(Integer.toString(i));
    }
} catch (InterruptedException ie) {
    ie.printStackTrace();
}
```

- Put this code into an onClick() method (as well as create a basic layout with a button and a TextView) and run the app. What happens? Why is this approach not advised?
- Refactor the code using AsyncTask.

Checklist

- ☐ Code snippet of original code is included.
- ☐ Description of what happens on run is included, with reference to good practice using threads on Android.
- ☐ Code snippet of AsyncTask refactoring is included with correct UI updating.

Tasks for COS80019 students only

The following tasks are core tasks that only Masters students should attempt. These tasks are optional for undergraduate students. The general concepts related to these questions are covered in the lectures, but Masters students are also expected to read, briefly research a bit more broadly to answer these questions.

2. Processes and threads

Android treats each application as a separate process and all components within that (including Activities) run on the main thread. The architecture also has concepts for Foreground, Visible, Service, Empty, and Background processes. What are the different types of high-level components within an application, and what do the concepts Foreground, Visible, Empty, and Background mean?

Reference: <https://developer.android.com/guide/components/processes-and-threads>,
<https://developer.android.com/guide/components/activities/process-lifecycle>

Checklist

- ☐ What are the different types of high-level components within an application (e.g. Activity, Service, Receiver, Provider are explained – can be as a table).
- ☐ What do the concepts Foreground, Visible, Empty, and Background mean? -- elaborated as a simple table or in another way.

Core/Extension Tasks

All tasks in this assignment are “core”. You must complete all core tasks, submit for feedback, and achieve a pass for all tasks in order to be eligible for a pass grade in this unit.

Submission

You are required to submit a PDF report using Doubtfire:

- login to Doubtfire at <http://doubtfire.ict.swin.edu.au>
- The header (or) footer of the document must contain your name, student id, and unit code.
- The document must have a title (e.g. Submission for Assignment <number>)
- Evidence that shows you completed each task must be presented in a separate section.



COS30017/COS80019 Semester 2 2018

- The document does NOT need a table of contents nor a cover page.

The reports are assessed and feedback given via Doubtfire and, if required, in your lab. You are expected to incorporate the feedback (esp. if changes are required) and submit the changed reports as part of the final portfolio.

Note: This is a formative assignment. That is, an assignment designed to provide feedback. If you fail this assignment, you have one week to make corrections and resubmit to pass.

Demonstration

You may be asked to demonstrate your assignment in the lab. You should be able to do this and explain your code when asked in the lab session.

