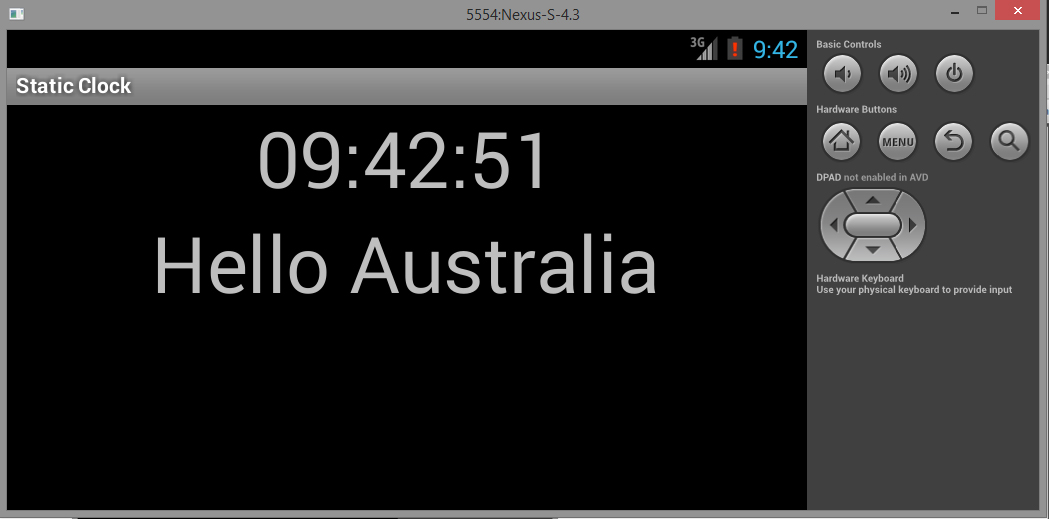
**Submission for Assignment 02**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Task 1**

a) The information updates during the orientation because the application is not constantly being refreshed. The time is only refreshed when the application is restarted (onCreate) as that is when the script is called.



b) **Pause** temporarily pauses the program, this usually happens when the user minimizes the application but it is still running /is visible in the background. No data is lost.

**Resume** resumes the program from the pause, so when the user returns to the application it will show the same data as before it was minimized.

**Stop** is when the application is no longer visible at all, to access the application again, you must Restart it, or Destroy it.

**Task 2**

In order to make the application restart, you need to minimize (but not close!) the application. To do this, you need to press the home button while in the app. This will stop and pause the application until the application is opened again.



**Task 3** (Same screenshot as Task 2)

The only state that is not shown in the screenshot is the onDestroy state. When closing the app (using the return button), it resets the current log of events because the program has been shut down and can no longer record the data. Therefore we must refer to the emulator's event log as opposed to the actual application's log.

**Possible events:**

onCreate, onStart, onResume, onPause, onStop, onRestart, onDestroy

**Logcat evidence:**

Click App Icon:

08-19 10:02:28.522 943-943/swin.examples I/LIFE-CYCLE-ACTIVITY-CURRENT-STATE﹕ onCreate

08-19 10:02:28.522 943-943/swin.examples I/LIFE-CYCLE-ACTIVITY-CURRENT-STATE﹕ onStart

08-19 10:02:28.522 943-943/swin.examples I/LIFE-CYCLE-ACTIVITY-CURRENT-STATE﹕ onResume

Click Home Icon:

08-19 10:03:30.911 943-943/swin.examples I/LIFE-CYCLE-ACTIVITY-CURRENT-STATE﹕ onPause

08-19 10:03:32.361 943-943/swin.examples I/LIFE-CYCLE-ACTIVITY-CURRENT-STATE﹕ onStop

Click App Icon: (again)

08-19 10:03:56.011 943-943/swin.examples I/LIFE-CYCLE-ACTIVITY-CURRENT-STATE﹕ onRestart

08-19 10:03:56.021 943-943/swin.examples I/LIFE-CYCLE-ACTIVITY-CURRENT-STATE﹕ onStart

08-19 10:03:56.031 943-943/swin.examples I/LIFE-CYCLE-ACTIVITY-CURRENT-STATE﹕ onResume

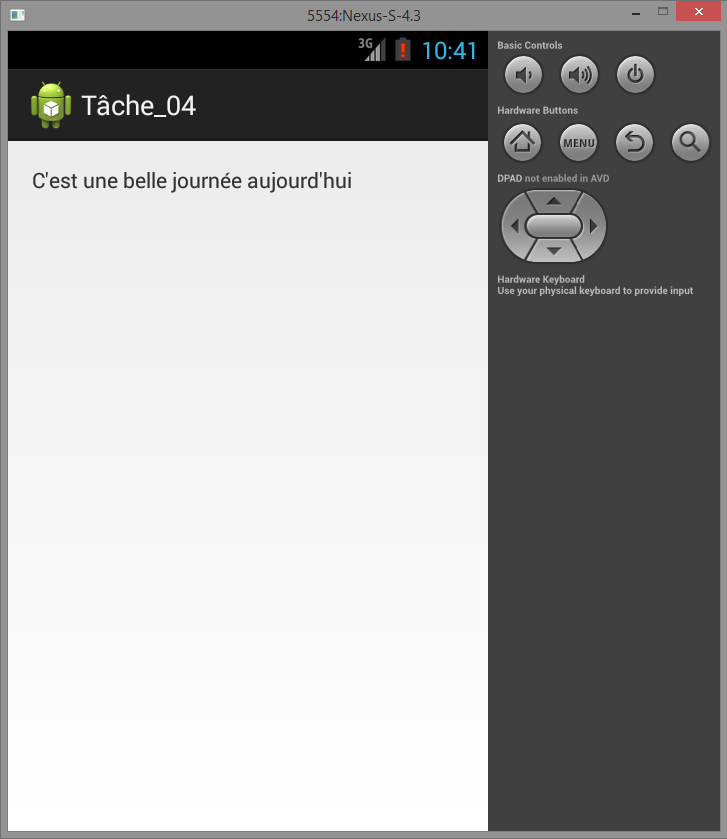
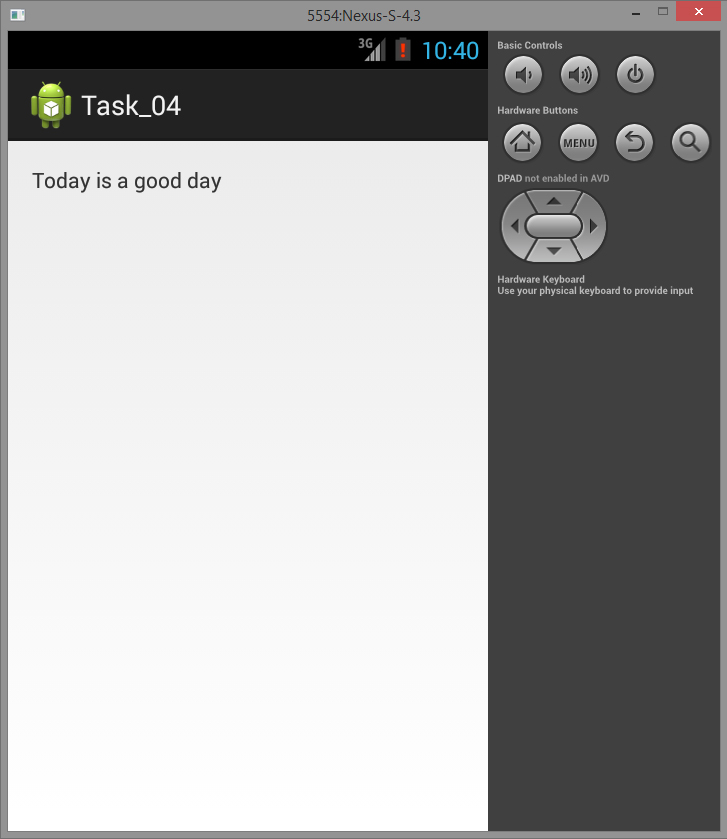
Click Return:

08-19 10:06:09.601 943-943/swin.examples I/LIFE-CYCLE-ACTIVITY-CURRENT-STATE﹕ onPause

08-19 10:06:10.862 943-943/swin.examples I/LIFE-CYCLE-ACTIVITY-CURRENT-STATE﹕ onStop

08-19 10:06:10.924 943-943/swin.examples I/LIFE-CYCLE-ACTIVITY-CURRENT-STATE﹕ onDestroy

**Task 4**



String Externalization allows for text to be translated to different languages without hard coding it into the java program (or requiring being recompiled). By doing this, you save space on the hardware and time of the programmer.

<TextView  
 android:text="@string/good\_day"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content" />  
  
**values.xml -**<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <string name="app\_name">Task\_04</string>  
 <string name="good\_day">Today is a good day</string>  
 <string name="action\_settings">Settings</string>  
</resources>

**fr\strings.xml -**<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <string name="app\_name">Tâche\_04</string>  
 <string name="good\_day">C\'est une belle journée aujourd\'hui</string>  
 <string name="action\_settings">Paramètres</string>  
</resources>