Feedback — Week 2 Quiz

You submitted this quiz on **Thu 30 Jan 2014 3:07 AM PST**. You got a score of **10.00** out of **10.00**.

Question 1

Which two of the following statements capture the main purposes of the Service class?

Your Answer		Score	Explanation
☑ To support long-running, in the background operations.	✔	0.25	
☑ To enable remote (inter-process) operations.	✔	0.25	
☐ To provide an application's user interface.	✔	0.25	
☐ To manage concurrent access to shared databases.	✔	0.25	
Total		1.00 / 1.00	

Question 2

Ignoring the initial xml declaration, what is the outermost xml tag used in the AndroidManifest.xml file?

You entered:

<manifest>

Your Answer		Score	Explanation
<manifest>	✔	1.00	
Total		1.00 / 1.00	

Question 3

How would you specify a string called "exit_message" whose value is "Goodbye", within a strings.xml file?

Your Answer	Score	Explanation
⦿ <string name="exit_message">Goodbye</string>	✔ 1.00	
○ <value android:id="string_name">text_string</value>		
○ <string> "Goodbye" </string>		
○ <string name="string_name">text_string</string>		
Total	1.00 / 1.00	

Question 4

(True or False) One of the main jobs of the Activity class is to support concurrent access to shared, inter-application data.

Your Answer	Score	Explanation
○ True		
⦿ False	✔ 1.00	
Total	1.00 / 1.00	

Question 5

The example applications for this lecture contain Activities that call setContentView(), passing in a resource ID. Which one of the following statements correctly reflects what the

setContentView() method does?

Your Answer	Score	Explanation
○ It inserts a package in the PackageManger's database.		
○ It attaches button listeners to the code.		
● It processes the underlying resource file to create Java Objects corresponding to the elements specified in the resource file.	✔ 1.00	
○ It validates the data type of the resource ID.		
Total	1.00 / 1.00	

Question 6

Suppose that your application has brought ActivityA into the foreground. The user then presses a button shown by ActivityA, which causes ActivityB to be brought into the foreground, ready for user interaction. At this point, what was the last lifecycle method called on ActivityA and on ActivityB?

Your Answer	Score	Explanation
● ActivityA.onStop() and ActivityB.onResume().	✔ 1.00	
○ ActivityA.onResume() and ActivityB.onDestroy().		
○ ActivityA.onPause() and ActivityB.onRestart().		
○ ActivityA.onDestroy() and ActivityB.onCreate().		
Total	1.00 / 1.00	

Question 7

Suppose you write an Activity that edits some persistent user information such as the user's account name, but does not save it immediately. This important data must be saved before the application exits. In which lifecycle method should you normally save the data?

Your Answer	Score	Explanation
⚪ onDestroy().		
⦿ onPause().	✔ 1.00	
⚪ onStop().		
⚪ onResume().		
Total	1.00 / 1.00	

Question Explanation

Remember that onStop() and onDestroy() may not always be called.

Question 8

Suppose you attach an OnClickListener to a Button in your Activity. When will this Object's onClick() method be called?

Your Answer	Score	Explanation
⚪ When the user hits the Back Key.		
⦿ When the user presses and releases the Button in quick succession .	✔ 1.00	
⚪ When the user hits the Home Key.		
⚪ When the user presses and holds the Button.		
Total	1.00 / 1.00	

Question 9

One example of a configuration change, is when the user changes the device's global font size. If you want to handle this configuration change manually, what value would you add to a android:configChanges attribute in the application's AndroidManifest.xml file? See: http://developer.android.com/guide/topics/manifest/activity-element.html for more information.

You entered:

fontScale

Your Answer		Score	Explanation
fontScale	✔	1.00	
Total		1.00 / 1.00	

Question 10

(True or False) Unless you have a strong reason for doing so, you should generally avoid handling configuration changes manually.

Your Answer		Score	Explanation
◯ False			
⦿ True	✔	1.00	
Total		1.00 / 1.00	

Question Explanation

Sometimes you may decide to handle configuration changes manually. For example, when your application doesn't need to update resources during a particular configuration change and when your application's performance would suffer from an activity restart.