Week 5 Quiz

Help

The **due date** for this quiz is **Mon 3 Mar 2014 6:01 AM CET**.

- [x] **In accordance with the Coursera Honor Code, I (Luigi Fugaro) certify that the answers here are my own work.** **Thank you!**

Question 1

What types of User Notifications are provided by Android?

- [x] Notification Area Notifications.
- [ ] Fragments.
- [x] Toast Messages.
- [x] Dialogs.
- [ ] Tabs.

Question 2

(True or False) Toast messages are used to get information from the user?

- ( ) True.
- (•) False.

Question 3

Why do Notification Area Notifications use PendingIntents?

- ( ) The PendingIntent can add a listener to the underlying Intent
- ( ) The underlying Intent has a reference to the sending component which can lead to memory leaks.

- To allow Extra data to be provided to the Activity that will be started.
- The underlying Intent will be used by the system, rather than by the component that created it.

Question 4

Which of the following capture why it is preferable to notify the user with a Notification Area Notification, rather than with a Dialog, or vice versa.

- Use a Dialog (DialogFragment) when using a large screen device such as a tablet.
- Use a Notification Area Notification to prevent onPause() from being called.
- Use a Dialog when the application needs to get user feedback.
- Use a Notification Area Notification when the user should be notified outside of any currently running application.

Question 5

When should your application send broadcasts using the LocalBroadcastManager class, rather than by using the Context class or vice versa?

- Use the Context class when the broadcast must be sticky.
- Use the Context class to improve application reliability.
- Use the LocalBroadcastManager to broadcast Intents that will only be received within the same application the sends the broadcasts.
- Use the LocalBroadcastManager to register BroadcastReceivers that don't want to receive broadcasts from outside the application.

Question 6

If your application only wants to receive certain broadcasts while it is active and in the foreground, which of the following scenarios might it implement?

- Load the Intents through a menu or ActionBar action.
- Statically register its BroadcastReceivers with low priority.
- Dynamically register its BroadcastReceivers in onResume() and unregister them in onPause().
- Dynamically register its BroadcastReceivers with low priority. The use abortBroadcast() at runtime to prevent delivery.

Question 7

Which of the following methods is guaranteed to run on the application's UI Thread?

- [x] View.post().
- [ ] AsyncTask.doInBackground().
- [ ] Handler.sendMessage().
- [x] Activity.runOnUIThread().

Question 8

Which of the following statements correctly capture why an application that uses a Handler, might send Messages to the Handler, rather than post Runnables to it, or vice versa?

- [x] Messages are used when the Handler implements the Message response.
- [x] Runnables are used when the Sender implements the action to be taken.
- [ ] Messages can take parameters. Runnables can't.
- [ ] Runnables are less efficient than messages.

Question 9

Which of the following statements capture how Alarms are different from other Android capabilities?

- [ ] Handlers cannot be used to send Intents at a future point in time.
- [x] Alarms are fired at a particular time in the future. Regular Intent Broadcasts are handled at the time the Intent is broadcast.
- [x] Notification Area Notifications inform users about events without interrupting their work, while Alarms don't directly inform users.

Question 10

How does an application get access to the AlarmManager?

- (•) Use the Context.getSystemService() method to retrieve a reference to the AlarmManager service.
- ( ) Put a <manager> tag in the application's AndroidManifest.xml file.
- ( ) Use the AlarmManager() constructor to create an instance of the AlarmManager.
- ( ) Use the AlarmManager.newInstance() method to retrieve the singleton instance of the AlarmManager.

Question 11

When setting alarms, it's often better to use the ELAPSED_REALTIME or ELAPSED_REALTIME_WAKEUP alarm types, rather than RTC or RTC_WAKEUP alarm types. Which of the following statements explains why RTC and RTC_WAKEUP alarms might not be the best approach in some cases?

- [ ] ELAPSED_REALTIME Alarms can fire when the CPU is in sleep mode.
- [x] If the network resets the system clock, RTC Alarms may fire at unpredictable times.
- [ ] It doesn't really matter, because you can easily convert from one time interpretation to the other.

- [x] If the user manually changes the time zone or modifies the system clock, RTC Alarms may fire at unpredictable times.

Question 12

For API targets prior to 19: The setInexactRepeating() method is intended to give Android flexibility in the exact timing of alarms. Assuming that mAlarmManager is a valid reference to the AlarmManager and that pi is a valid reference to a PendingIntent, why doesn't the following code snippet (modified from the AlarmCreate application shown in this lesson) accomplish that purpose?

```
mAlarmManager.setInexactRepeating(AlarmManager.ELAPSED_REALTIME,
SystemClock.elapsedRealtime(),15000, pi);
```

- ( ) setInexactRepeating() requires a time interval of 60000 or greater.
- (•) setInexactRepeating() requires a specific interval constant, such as INTERVAL_FIFTEEN_MINUTES.
- ( ) setInexactRepeating() requires an alarm type of RTC or RTC_WAKEUP.
- ( ) setInexactRepeating() is a method of the Alarm class.

Question 13

Android supports several HTTP clients. Which one of the following HTTP clients will be Android's preferred HTTP client in the future? See http://android-developers.blogspot.com/2011/09/androids-http-clients.html for more information.

- ( ) AndroidHttpClient.
- (•) HttpURLConnection.
- ( ) DefaultHttpClient.

Question 14

Which of the following statements are generally true about DOM parsers?

- [ ] DOM parsers use a streaming model in which the parser calls back into the application when specific elements are parsed.
- [x] DOM parsers tend to use more memory than the other kinds of Parsers we discussed in this lesson.
- [x] DOM parsers convert an XML document into a tree structure, which can make it easier to do whole document analyses.
- [ ] DOM parsers provide iterators that pull XML content into an application on demand.

- [x] **In accordance with the Coursera Honor Code, I (Luigi Fugaro) certify that the answers here are my own work.** **Thank you!**

Submit Answers Save Answers