**block 4**

**TOTAL POINTS 9**

1.Question 1

What audio or video sources can you play in your application?

**1 / 1 point**



data streamed from a network connection

**Correct**



multimedia files stored as resources of the application

**Correct**



multimedia files stored on the file system of the device

**Correct**

2.Question 2

In which folder must you put the multimedia files which are part of your application?

**1 / 1 point**

raw

**Correct**

3.Question 3

Which of the following propositions are valid names for MP3 files placed in the res/raw folder?

**1 / 1 point**



my-music



007\_mymusic



mymusic

**Correct**



007



\_mymusic



mymusic2

**Correct**



myMusic



my\_music

**Correct**

4.Question 4

Which class can you use to play a music file?

**1 / 1 point**

MediaPlayer

**Correct**

5.Question 5

It's possible to add music to the output audio flow during a phone call.

**1 / 1 point**



true



false

**Correct**

For now, music can only be sent to the standard output, i.e. the phone speaker or bluetooth speaker.

6.Question 6

The music files in the application resources folder must be in the raw ("Real Audio Wrapper") audio file format, i.e. uncompressed and without metadata.

**1 / 1 point**



true



false

**Correct**

The folder name is "raw" but the file can be encoded in one of the supported formats such as MP3 or MIDI for instance.

7.Question 7

Which method of the [MediaPlayer](http://developer.android.com/reference/android/media/MediaPlayer.html" \t "_blank) class allows you to read a music file from a given point in time (rather than from the beginning of the song)?

**0 / 1 point**

setLooping

**Incorrect**

[Use the API documentation to find out the right answer.](http://developer.android.com/reference/android/media/MediaPlayer.html)

8.Question 8

Once your application no longer plays music, it's a good practice to

**1 / 1 point**



call the reset method of the MediaPlayer object



not destroy the MediaPlayer object in case it might be useful elsewhere in the application



free the associated resources by using the release method from the MediaPlayer class



terminate the current activity (the one which was playing the music)

**Correct**

[Read more about this in the Android documentation.](http://developer.android.com/reference/android/media/MediaPlayer.html#StateDiagram)

9.Question 9

If you call the stop method of the MediaPlayer object to turn the music off, calling next the start method will resume the music playing from the beginning of the file.

**1 / 1 point**



true



false

**Correct**

Calling the start method is not enough. Indeed calling the stop method passes the MediaPlayer into a state which does not allow the start method to be called. You need to call the create method again (for instance, but there are other ways to reach a state in which a call to start is legit).

[Read more about this in the Android documentation.](http://developer.android.com/reference/android/media/MediaPlayer.html)

Instead of stop, you can call the pause method ; you will be allowed to call start next. But the music will resume from where it has been paused. If you want to start the music over from the beginning, add a call to seekTo to position the playback at 0.