MONASH University

FIT2081 Mobile application development - S1 2023 MUM

Dashboard / My units / FIT2081 - S1 2023 MUM / Assessments / Week 3: Workshop Quiz

Started on	Tuesday, 14 March 2023, 9:12 PM
State	Finished
Completed on	Tuesday, 14 March 2023, 9:21 PM
Time taken	8 mins 38 secs
Grade	Not yet graded

Print friendly format

Question **1**

Complete

Marked out of 10.00

Question 1

Line 2, private mode is 0 so instead of MODE_PRIVATE, it should be 0.

Line 2, since getPreference is used, a file name is not required, just the mode so "week3.quiz" is not required unless their intention is to specifically save in another specific file, they should use getSharedPreferences instead so the line of code will look like: SharedPreferences sharedPreferences = getSharedPreferences ("week3.quiz", 0);

Line 3 should be SharedPreferences.Editor editor = sharedPreferences.edit();

Line 4, 15 is of int datatype, not string.

Line 5, "true" is of String datatype, not boolean.

Line 6, should be sharedPreferences.commit(); to save the data but sharedPreferences.apply() should work the same as learnt in forum.

Question **2**

Complete

Not graded

Question 2

(a) If onSaveInstanceState() is not called, it means that the user is done with the activity. Since onSaveInstanceState() callback is mainly called when there is any configuration changing within the device like when the device screen is rotated so the activity going to the background will be destroyed and recreated on another different orientation for example, when a phone is rotated from landscape mode to portrait mode. So if onSaveInstanceState() callback is not implemented, the view state of an activity will not be saved.

(b) Calling super of any method is to perform a bunch of standard processing as per requirement of any activity as there are customised processing specifically to each particular activity. If the onSaveInstanceState() callback were to be implemented but not calling the super.onSaveInstanceState() method will result in no overriding done so none of the state of the view is saved so if the application is placed in the background, android will auto kill the application and once onSaveInstanceState() is used, none of the old process will be reused and so android will just open a new process for the activity and hence relaunch a new activity that is not the old activity that was once placed in the background.

(c) When onSaveInstanceState() is implemented with super.onSaveInstanceState() method

(c) When onSaveInstanceState() is implemented with super.onSaveInstanceState() method, user can retrieve and reuse their application's old process when the application was placed in the background as with the super.onSaveInstanceState() method called, Android will know that the view state will be saved and stored for future reuse purposes and when the user returns to the application to continue on their unfinished progress, android will use super.onCreate() to restore the view state saved when super.onSaveInstanceState() method is called.

Question **3**

Not answered

Not graded

Question 3

Question **4**

Not answered

Not graded

Question 4

Question 5

Not answered

Not graded

Question 5

◄ Week 2: Workshop Quiz

Jump to... ⬍

Week 2: LAB ►