

Higher Diploma in Computing, DT8265

Programming for Mobile and Smart Devices

Coursework 2 – Lab Test - FlagQuiz App

Add the software requirements listed below to the existing FlagQuiz app. Include a **//TODO x** (where x is the requirement number below) commenting with a description for all new code you add to java and xml resources.

1	add your class photo as the app icon.	2	/2
2	update the app name to FlagQuiz_firstname	2	/2
3	Currently, if a user deselects all the Regions in the Preferences Settings, North America is set by default. Change the default to Europe and update the appropriate string resources and Toast message.	6	/6
4	If a user should reset the value of FLAGS_IN_QUIZ (via a possible future Preference setting), ensure that the current code will not break and make updates as necessary.	4	/4
5	Keep track of the score as the user proceeds through the quiz. Add a TextView to the top right of the UI connected to a string resource. Position the TextView just below the <i>questionNumberTextView</i> with the text colored green, reusing the same theme as was used to output the correct flag answer. e.g. Score: 21 Give the user the most points for answering correctly on the first guess, fewer points for answering correctly on the next guess, etc. see Note1	6 10	/6 /10
6	Count the number of questions that were answered correctly on the FIRST attempt. Update the dialog message that is displayed after all the questions have been answered, to include how well the user performed on the first guesses.	5 5	/5 /5
7	Add the material design Help icon to the AppBar. Change its fillcolor to white. When clicked, a Dialog appears that is linked to a help Resource explaining the scoring system. Configure the menu so that the Help icon is visible in all orientations. The existing settings menu icon is only visible in Portrait orientations. The dialog can be dismissed by clicking on the background. see Note2	4 8 4	/4 /8 /4
8	If the user guesses the correct flag, include a “bonus question” asking the user to name the capital of the country using the same guess buttons. If the user answers correctly on the first guess for the capital, add 10 bonus points to the score; otherwise, simply display the correct answer in a Dialog, then allow the user to proceed to the next flag. see Note3	20 20	/20 /20
9	Investigate class SharedPreferences. When the user achieves a new high score, they are informed via a Dialog. The high score is stored in the Shared-Preferences file that also stores this app’s settings. see Note4	10	/10

Note1: There are several ways you could calculate the score. For example, if the user has the app configured to show only 4 guesses for each flag, you could give the user 4 points for a correct guess on the 1st try, 3 points for a correct guess on the 2nd, 1 on the 3rd attempt, and no points for a correct guess on the last attempt. If the user has the app configured to show 6 or 8 guesses for each flag, you could increase the points appropriately.

Note2: create the resource with an explanation of the scoring system and game functionality.

Note3: Delete all the flags in the Europe folder and replace them with the ones in the share folder containing the capital in the image name. The image files have been adapted for capitals for Europe only. Copy the Europe folder from the Share drive into your Project and remember to set the regions in the Preferences to Europe when you test the App. You include the capital in a similar manner to how we named the app’s images to include each flag’s region and country name. You’d then parse the file names appropriately to capture the capital information in addition to what we do to obtain the country names in the app.

Note4: This requires the same SharedPreferences techniques that the app currently uses via the PreferencesFragment to store the enabled regions and number of guesses to display with each flag.