University of Westminster
School of Computer Science & Engineering

5COSC011C Coursework 1 (Semester 2)

Module leader	Torin Wirasingha
Unit	Coursework 1
Weighting:	50%
Qualifying mark	30%
Description	
Learning Outcomes Covered in this Assignment:	LO1, LO4, LO5
Handed Out:	15/2/2021
Due Date	8/3/2021 13:00
Expected deliverables	Source code/XML files/Resources (images, etc)
Method of Submission:	Online via Blackboard
Type of Feedback and Due Date:	Individual feedback via Blackboard within 3 weeks of submission **All marks will remain provisional until formally agreed by an Assessment Board.**

Assessment regulations

Refer to section 4 of the "How you study" guide for undergraduate students for a clarification of how you are assessed, penalties and late submissions, what constitutes plagiarism etc.

Penalty for Late Submission

If you submit your coursework late but within 24 hours or one working day of the specified deadline, 10 marks will be deducted from the final mark, as a penalty for late submission, except for work which obtains a mark in the range 40 – 49%, in which case the mark will be capped at the pass mark (40%). If you submit your coursework more than 24 hours or more than one working day after the specified deadline you will be given a mark of zero for the work in question unless a claim of Mitigating Circumstances has been submitted and accepted as valid.

It is recognised that on occasion, illness or a personal crisis can mean that you fail to submit a piece of work on time. In such cases you must inform the Campus Office in writing on a mitigating circumstances form, giving the reason for your late or non-submission. You must provide relevant documentary evidence with the form. This information will be reported to the relevant Assessment Board that will decide whether the mark of zero shall stand. For more detailed information regarding University Assessment Regulations, please refer to the following website :http://www.westminster.ac.uk/study/current-students/resources/academic-regulations

It is recognised that on occasion, illness or a personal crisis can mean that you fail to submit a piece of work on time. In such cases you must inform the Campus Office in writing on a mitigating circumstances form, giving the reason for your late or non-submission. You must provide relevant documentary evidence with the form. This information will be reported to the relevant Assessment Board that will decide whether the mark of zero

shall stand. For more detailed information regarding University Assessment Regulations, please refer to the following website:**http://www.westminster.ac.uk/study/current-students/resources/academic-regulations**

It is recognised that on occasion, illness or a personal crisis can mean that you fail to submit a piece of work on time. In such cases you must inform the Campus Office in writing on a mitigating circumstances form, giving the reason for your late or non-submission. You must provide relevant documentary evidence with the form. This information will be reported to the relevant Assessment Board that will decide whether the mark of zero shall stand. For more detailed information regarding University Assessment Regulations, please refer to the following website:**http://www.westminster.ac.uk/study/current-students/resources/academic-regulations**

5COSC011C MOBILE APPLICATION DEVELOPMENT - Assignment 1
Deadline 8/3/2021, 13:00

Torin Wirasingha
Email: torin.w@iit.ac.lk

Description

You are required to implement an Android application described by the specifications below.

You are not allowed to use third-party libraries. The only libraries that you can use are the standard Android API libraries found in the following URL:

  `https://developer.android.com/reference/`

It is important to follow exactly the specifications and your implementation <u>must</u> conform to these:

The application developed will be displaying cars and you can use any car images found in the internet (make sure there are no copyright issues):

1. When the application starts, it presents the user with 4 buttons labelled *Identify the Car Make*, *Hints*, *Identify the Car Image* and *Advanced Level*. *(4 marks)*

2. (a) Clicking on the *Identify the Car Make* button, it should display to the user one random car image picked randomly (and NOT in the same random sequence every time the application restarts) from the set of all the car images that your application contains (the minimum number of images in your application should be 30).

   The screen should also display a dropdown list with ALL the names of the car makes that your application contains (e.g. BMW, Ford, etc.) and a button labelled **Identify**. The user should use the list to select the car make that the car image corresponds to and clicking the "Identify" button he/she submits the guess to the app. (10 marks)

   (b) As soon as the user submits the answer, the message *CORRECT!* (in green colour) or the message *WRONG!* (in red colour) appears on the screen with the name of the correct car make in Yellow colour, depending on whether the answer given is correct or incorrect respectively. *(5 marks)*

   (c) Following this, the label of the "Identify" button should change to display "Next". The user should click the "Next" button to advance to another screen presenting him with a new random car image and giving him the chance to play again. Every time that this option is chosen a different image should be displayed.
   (4 marks)

3. (a) Clicking on the *Hints* button, it should display to the user one random car image picked from all the car images that the application contains.

The screen should also display as many dashes as the name of the car make that the image corresponds to, a textbox and a **Submit** button. The user will be using the textbox to type a single character which is part of the name of the car make in an attempt to guess the make character by character and submit his guesses using the "Submit" button. Any correct guesses of characters will be replacing the corresponding dashes with the actual character (**ALL** instances of the dashes corresponding to that character will be substituted).

For example, if the image displayed belongs to a Mercedes car the following dashes appear initially:

```
--------
```

If the user types the 'C' character in the textbox (either in uppercase or lowercase, i.e. case insensitive insertion) and clicks on the "Submit" button the area displaying the dashes will look as:

```
---C----
```

Following the above, if the user types the 'E' character and presses submit the dashes area will look as:

```
-E-CE-E-
```

If the user types a character not existing in the name of the car make, the dashes area remains the same and it does not change its contents.
(11 marks)

(b) The user is only allowed up to 3 incorrect character guesses, after which the message *WRONG!* (in red colour) appears on the screen with the name of the correct car make in Yellow colour (when all 3 attempts are used). In case that the user has guessed all the characters of the car make's name the message *CORRECT!* should appear in green colour.

Following either of the cases (i.e. correct or incorrect), the label of the "Submit" button should change to display "Next". The user should click the "Next" button to advance to another screen presenting him with a new car image and giving him the chance to play again. Every time that this option is chosen a different car image should be displayed.
(3 marks)

4. Clicking on the *Identify the Car Image* button, it should display to the user 3 different unique (in terms of make, i.e. no two images should correspond to the same car manufacturer) random car images picked from the images that the application contain. The images should be clickable.

The screen should also display the correct name of a car manufacturer corresponding to one of the displayed images and a button labelled **Next**.

The user's aim is to click on the image corresponding to the displayed name, after which (a single attempt is only allowed) the message *CORRECT!* (in green colour) or the message *WRONG!* (in red colour), depending on whether the answer given is correct or incorrect respectively.

Following this, the user should click the "Next" button to advance to another screen presenting him with 3 new random images and giving him the chance to play again. Every time that this option is chosen different images should be displayed.
(11marks)

5. (a) Clicking on the *Advanced Level* button, it should display to the user 3 different <u>unique</u> random car images picked from all the car images in the application together with 3 textboxes and a **Submit** button.

   The aim of the user in this level is to guess the car manufacturer names of all the 3 displayed cars using the corresponding 3 textboxes. The name of the car manufacturer corresponding to the first image should be typed in the first textbox, the name of the manufacturer corresponding to the second image should be typed in the second textbox and the name of the manufacturer corresponding to the third image should be typed in the third textbox.

   The user will be using the "Submit" button to submit his answers. Once he clicks on that button the answers are checked. If any of the 3 textboxes contains the correct car make name then that textbox becomes uneditable (i.e. the user cannot change any characters in it) and in <u>*GREEN colour*</u>. Any textboxes containing an incorrect guess remain editable but with a <u>*RED colour*</u> (either the textbox background becomes red or the characters face becomes red) .

   If the user gets all the 3 manufacturers correct, then the message *CORRECT!* (in green colour) appears, the label of the "Submit" button becomes "Next" and clicking on the button leads the users to a new game with 3 new images being displayed.
   (11 marks)

   (b) The user is allowed only 3 incorrect attempts. A single attempt is considered complete every time that the "Submit" button is clicked.

   After 3 incorrect attempts (i.e. the user has clicked the "Submit" button 3 times and at least 1 of the 3 car make guesses is incorrect) the message *WRONG!* (in red colour) appears on the screen with the names of the correct manufacturers in Yellow colour displayed just below the incorrect answer(s). (If 1 or 2 guesses are correct the *"Yellow"* name does not appear below the correct answers.) After the 3 incorrect attempts the label of the "Submit" button becomes "Next" and clicking on the button leads the users to a new game with 3 new images being displayed.
   (3 marks))

   (c) Implement a score for the user which is displayed on the top right of the screen. The user gets 1 point for every image he guesses correctly, i.e. if in the first screen he guesses all 3 car makes correctly he gets 3 points. If after 3 attempts in a screen, he guesses correctly only 2 manufacturers out of the 3, he gets 2 points and so on.
   (3 marks)

6. Extend the application by providing a "Switch" button in the app home screen (first screen with the 4 buttons game levels) which can be switched on or off. Every time it is switched on, all of the game levels (all 4 screens) will include a countdown timer, counting from 20 down to 1, every tick occurring after 1 second exactly. The countdown timer is displayed in the main screen of each game level.

As soon as the counter reaches the value of 0, the current game screen stops, i.e. the message *CORRECT!* or *WRONG!* is displayed to the user and the "Submit" button changes status and becomes the "Next" button as described before. Effectively, every time the

counter reaches zero is equivalent to the user pressing once the "Submit" button for that game level. Levels allowing more than one attempts to the user should start a new timer for that screen until all attempts are exhausted. Thus, a value of the timer reaching 0 is functionally EXACTLY like pressing the "Submit" button once.

(10 marks)

Marking Scheme: The marks achieved for each part of the program are indicated in the description of the task above. In addition to these the following will be taken into account:

- *Code readability* (structure, comments, variable naming, etc,): 5%

- *Implementation* (e.g. quality, efficiency, look and feel of the application, based on fonts, colours, etc.): 20%

The maximum for work which does not compile (or XML files with syntax errors causing the Java code not to compile) is 30%.

Submission of assignments using a different method other than Blackboard will not be accepted and zero (0) marks will be awarded in such cases.

Deadline: Monday 8th of March 2021, 13:00.

Submission Instructions

Files to submit: All of the files of the Android Studio project of your application in a zip file. *Referencing code:* Any code taken from other resources (i.e. a textbook or internet) should be referenced in comments within your code (full textbook details or full web URL), identifying the exact code that you used it as part of your application and the exact portions of the original source code that you reused.

You should submit via BlackBoard's Assignment functionality (do NOT use email, as email submissions will be ignored.), all the files described above. A single zip file with the name `wNNNNNNNN` (where `wNNNNNNN` is your university ID login name) containing all the above files could be submitted alternatively. You can create such a file by using the main menu in Android Studio and choose `File->Manage IDE Settings->Export to Zip File...`

Note that Blackboard will allow to make a submission multiple times. Make sure before submitting (i.e. before pressing the `Submit` button), that all the files you want to submit are contained there (or in the zip file you submit).

In the case of more than one submissions, only your last submission before the deadline given to you will be marked, so make sure that all the files are included in the last submission attempt and the last attempt is before the coursework deadline.

Request to mark submissions which are earlier than the last submission before the given deadline will be ignored as it is your responsibility to make sure everything is included in your last submission.

The following describes how to submit your work via BlackBoard:

1. Access `https://learning.westminster.ac.uk` and login using your username and password (if either of those is not known to you, contact the Service Desk, tel: +44 (0) 207 915 5488 or log a call via `https://servicedesk.westminster.ac.uk`.).

2. Click on the module's name, MODULE: 5COSC011C.2020 MOBILE APPLICATION DEVELOPMENT found under `My Modules & Courses`.

3. Click on the `Assessment->Submit Coursework->Coursework`.

4. Click on `View Assignment`.

5. Attach your zip file containing all the Java source code files, by using the `Browse` button.

6. Create a Word or PDF file with the following information:

   - *Comments*: Type your full name and your registration number, followed by:

     "I confirm that I understand what plagiarism is and have read and understood the section on Assessment Offences in the Essential Information for Students. The work that I have submitted is entirely my own. Any work from other authors is duly referenced and acknowledged."

7. Attach the file with the statement above.

8. Check that you have attached both the zip and the statement file.

9. Click the `Submit` button.

If Blackboard is unavailable before the deadline you must email the Registry at `com_submission@iit.ac.lk` with cc: `to myself and your personal tutor` before the deadline with a copy of the assignment, following the naming, title and comments conventions as given above and stating the time that you tried to access Blackboard. You are still expected to submit your assignment via Blackboard. Please keep checking Blackboard's availability at regular intervals up to and after the deadline for submission. You must submit your coursework through Blackboard as soon as you can after Blackboard becomes available again even if you have also emailed the coursework to the above recipients.