

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:	10/2/2020
Due Date	9/3/2020 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 verbally straight after the viva and written individual feedback via Blackboard within 2 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>

5COSC005W MOBILE APPLICATION DEVELOPMENT - Assignment 1 *Deadline 9/3/2020, 13:00*

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

Description

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

You are not allowed to use third-party libraries. The only libraries 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 must conform to these:

The application developed will be displaying dog images of different breeds and you should use the images found in the following website:

<http://vision.stanford.edu/aditya86/ImageNetDogs/>

The website provides images for 120 dog breeds. Your application does not need to deal with all breeds and all images but it can use a subset of them as long as it gives a sufficient variety to the user. At a minimum, it should deal with images of 10 breeds but you might want to include more to keep the user more interested. For simplicity and for all the tasks below, you can hardcode in your program breed names and/or mappings of breed names to filenames.

You can download all images from the **Download** section of the above URL, from the link called **Images** (filename: *images.tar*).

1. When the application starts, it presents the user with 3 buttons labelled *Identify the Breed*, *Identify the Dog* and *Search Dog Breeds*. (4 marks)
2. (a) Clicking on the *Identify the Breed* button, it should display to the user one random dog image picked randomly (and NOT in the same random sequence every time the application restarts) from the dog images that your application contains.
The screen should also display a dropdown list with ALL the names of the breeds that you chose to include (see the comment above) and a button labelled **Submit**. The user should use the list to select the breed that the dog images corresponds to and clicking the “Submit” button he/she submits the guess to the app. (13 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 breed in Blue colour, depending on whether the answer given is correct or incorrect respectively. (11 marks)
- (c) Following this, 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 random dog image and giving him the chance to play again. Every time that this option is chosen a different dog image (possibly the same breed though) should be displayed.
(5 marks)
3. Clicking on the *Identify the Dog* button, it should display to the user 3 different unique random different breed dog images. The images should be clickable. It is not allowed to display the same image more than once, i.e. the 3 images should be unique, as it should also be the breed.
- The screen should also display the name of a breed corresponding to one of the displayed images and a button labelled **Next**.
- The user’s aim is to click on the dog image corresponding to the displayed breed name, after which (a single attempt is only allowed) the message *CORRECT!* (in green colour) or the message *WRONG!* (in red colour) appears, 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.
(18 marks)
4. Clicking on the *Search Dog Breeds* button, it should display to the user a textbox, a **Submit** button and a **Stop** button.
- The user will be using this part of the app to extend his/her knowledge about dog breeds. Typing the name of a breed and pressing the “Submit” button will start displaying to the user dog images belonging to the typed breed. A new image will automatically appear to the user every 5 seconds, like a slide show (without the need to press again the “Submit” button). The images will keep alternating in a completely random order for ever until the user presses the “Stop” button. Once the “Stop” button is pressed the image slide show stops and the user can type the name of a new breed and resubmit to start the display of a new sequence of images.
(17 marks)
5. Extend the application by providing a “Switch” button in the app home screen (first screen with the 3 buttons game levels) which can be switched on or off. Every time it is switched on, the first 2 game levels (“Identify the Breed” and “Identify the Dog”) will include a countdown timer, counting from 10 down to 1, every tick occurring after 1 second exactly. The countdown timer is displayed in the main screen of these 2 game levels.
- 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.

(15 marks)

6. For all the tasks the application should behave in a user friendly manner when the device is rotated from portrait to landscape and back to portrait mode. I.e. the application should resume from exactly the same point (same screen and data) when the orientation changes.

(7 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.): 5%

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

Based on the functionality implemented, the marks awarded will consist of 2 parts:

- 30% of the marks achieved will be awarded based on the submission.
- The remaining 70% of the marks for the implementation will be awarded after a compulsory viva, that will test the understanding of the code by the student. The student will be asked to demonstrate the application and will be asked questions about the code to demonstrate his/her understanding. No understanding of any parts of the code will result in no marks given for this component (i.e. a maximum of 30% can only be achieved if a student cannot explain his/her code and he is unable to point to any references within the code of where this code was found in a textbook or internet).

A compulsory viva for each student based on his/her submission will take place during the next tutorial session after the submission (all of them taking place in the weeks starting the 2nd of March and 9th of March). Each student will be notified a specific slot that he/she needs to attend. Failure to turn up in the viva slot designated (no changes will be allowed as this a normal examination) will result in awarding only 30% of the marks achieved for the submission (see marking scheme above)

It is the responsibility of each student to make sure that during the viva the code runs properly in the lab used during the viva, i.e. you should make sure in advance (allow enough time before the viva day) that everything is running properly in the machine you will be using. If you developed code at your home computer, it is your responsibility that you port it to the lab in advance, before the viva. Marks will be awarded based on the demo/viva and excuses of the type “it used to run - don’t know what happened since last time” will not be accepted or awarded with extra marks.

You are allowed to use your own laptop during the viva if you wish to.

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 9th of March 2020, 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 `wNNNNNNNN` 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->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, ask the HelpDesk at the Library.).
2. Click on the module's name, `MODULE: 5COSC011C.2019 MOBILE APPLICATION DEVELOPMENT` found under `My Modules & Courses`.
3. Click on the `Assignments` button found on the left hand side menu.
4. Click on `View/Complete Assignment`.
5. Attach your zip file containing all your files of your Android project, by using the `Browse` button.
6. Fill in the requested 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. Click the `Submit` button.

If Blackboard is unavailable before the deadline you must email 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.

Coursework Marking scheme

The Coursework will be marked based on the following marking criteria:

Criteria	Mark per component	Mark provided	Comments
Implementation	100		
Functionality	90		For a split of the marks see the subquestions description in the main description of the coursework
Code Readability	5		structure, comments, variable naming, etc,
Software Quality	5		Quality, efficiency, etc.
Total	100		