

## **5COSC023W Coursework 1 (Semester 2)**

Module leader	Dr D. Dracopoulos
Unit	Coursework 1
Weighting:	50%
Qualifying mark	30%
Description	
Learning Outcomes Covered in this Assignment:	LO1, LO4, LO5
Handed Out:	February 2024
Due Date	25/3/2024 13:00
Expected deliverables	Source code/XML files/Resources (images, etc)/Video
Method of Submission:	Online via Blackboard
Type of Feedback and Due Date:	Individual feedback via Blackboard within 3 weeks of submission <b>All marks will remain provisional until formally agreed by an Assessment Board.</b>

### **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>

# 5COSC023W MOBILE APPLICATION DEVELOPMENT - Assignment 1

*Deadline 25/3/2024, 13:00*

Dr Dimitris C. Dracopoulos  
*Email: d.dracopoulos@westminster.ac.uk*

## Description

You are required to implement an Android application using Kotlin and Jetpack Compose 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 (with the exception of Views that you should NOT use):**

<https://developer.android.com/reference/>

**You should NOT disable Activity recreation (e.g. when screen orientation changes) for any parts of your work**

The application developed will be displaying country flags and you should use the images found in the following website:

<https://github.com/hjnilsson/country-flags>

The website provides 256 countries flags images in different resolutions and also a file (`countries.json`) which maps the 2-letter ISO-3166 country code (used in the images filenames) to country names.

1. When the application starts, it presents the user with 4 buttons labelled *Guess the Country*, *Guess-Hints*, *Guess the Flag* and *Advanced Level*. (4 marks)
2. (a) Clicking on the *Guess the Country* button, it should display to the user one random flag image picked randomly (and NOT in the same random sequence every time the application restarts) from the 256 countries.  
The screen should also display a list with ALL the names of the countries and a button labelled **Submit**. The user should use the list to select the country that the flag corresponds to and clicking the “Submit” button he/she submits the guess to the app. (11 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 country in Blue colour, depending on whether the answer given is correct or incorrect respectively. (4 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 flag and giving him the chance to play again. Every time that this option is chosen a different flag should be displayed.

(4 marks)

3. (a) Clicking on the *Guess-Hints* button, it should display to the user one random flag image picked from the 256 countries.

The screen should also display as many dashes as the name of the country that the flag 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 country’s name in an attempt to guess the flag’s country 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 flag displayed belongs to France 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 ‘R’ character and presses submit the dashes area will look as:

-R--C-

If the user types a character not existing in the country’s name, the dashes area remains the same and it does not change its contents.

(14 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 country in Blue colour (when all 3 attempts are used). In case that the user has guessed all the characters of the country’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 random flag and giving him the chance to play again. Every time that this option is chosen a different flag should be displayed.

(2 marks)

4. Clicking on the *Guess the Flag* button, it should display to the user 3 different unique random flags images picked from the 256 countries. The images should be clickable.

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

The user’s aim is to click on the flag 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 flags and giving him the chance to play again. Every time that this option is chosen different flags should be displayed.

(12 marks)

5. (a) Clicking on the *Advanced Level* button, it should display to the user 3 different unique random flags images picked from the 256 countries together with 3 textboxes and a **Submit** button.

The aim of the user in this level is to guess the country names of all the 3 displayed flags using the corresponding 3 textboxes. The name of the country corresponding to the first flag should be typed in the first textbox, the name of the country corresponding to the second flag should be typed in the second textbox and the name of the country corresponding to the third flag 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 country name then that textbox becomes grayed-out (i.e. the user cannot change any characters in it) and in *GREEN* colour. Any textboxes containing an incorrect guess remain editable but with a *RED* colour (either the textbox background becomes red or the characters face becomes red) .

If the user gets all the 3 countries 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 flags being displayed.

(15 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 flags guesses is incorrect) the message *WRONG!* (in red colour) appears on the screen with the names of the correct countries in Blue colour displayed just below the incorrect answer(s). (If 1 or 2 guesses are correct the “Blue” 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 flags being displayed.

(4 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 country he guesses correctly, i.e. if in the first screen he guesses all 3 flags correctly he gets 3 points. If after 3 attempts in a screen, he guesses correctly only 2 flags out of the 3, he gets 2 points and so on.

(3 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. The rotation of the device should not change what was the user was seeing before the rotation.

**You should NOT disable Activity recreation (e.g. when screen orientation changes) for this or any parts of your work**

(9 marks)

7. 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 10 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.): 4%
- *Implementation* (e.g. quality, efficiency, look and feel of the application, based on fonts, colours, etc.): 4%

**In addition to the above marks indicated in each of the sub-questions in the specification, additional marks will be deducted if an application behaves in an unexpected way. For example, an application should not crash, the application should work properly even if a user enters invalid data or rotate the device.**

**The maximum for work which does not compile (or XML or other resources files with syntax errors causes the project not to build) is 30%.**

**Failure to submit a video or a working link to a video will result to a capping of your mark to 30% (i.e. if your submitted work is given initially a mark above 30% it will be capped to 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. The exception is a working link to a video outside Blackboard.**

**Deadline:** Monday 25th of March 2024, 13:00.

## Submission Instructions

*Files to submit:*

1. All of the files of the Android Studio project of your application in a zip file.

2. A video demonstrating all the functionality that you implemented for your application. If the video size exceeds the size allowed by Blackboard you can submit a link to a video to another site (e.g. Google Drive, Youtube, etc.) where you have uploaded the video. Make sure that if your video is located to another web site and not Blackboard that you have given appropriate permissions to external people to access your video.

*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->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: 5COSC023W.2023 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 Kotlin source code files and resources of the Android Studio Project, by using the `Browse` button.
6. Attach your video or include a link of your video as the first comment line) in the `Main-Activity` of your code.
7. 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."

8. Attach the file with the statement above.
9. Check that you have attached both the zip, the video (or the video link) and the statement file.
10. Click the **Submit** button.

If Blackboard is unavailable before the deadline you must email the Registry at [studentcentre@westminster.ac.uk](mailto:studentcentre@westminster.ac.uk) 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.