

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:	15/2/2022
Due Date	15/3/2022 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>

5COSC023W MOBILE APPLICATION DEVELOPMENT - Assignment 1

Deadline 15/3/2022, 13:00

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

Description

You are required to implement an Android application using Kotlin 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 must conform to these:

The application developed will be a game which players will be using to practice their arithmetic calculative skills:

1. When the application starts, it presents the user with 2 buttons labelled *New Game*, and *About*. (2 marks)
2. Clicking on the *About* button, it should present the user with a popup window which describes the author (student id and name) and the message:

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.

(2 marks)

3. Clicking on the *New Game* button, the user will be presented with the game screen which they interact with.

The screen displays 2 random arithmetic expressions one on the left and one on the right and 3 buttons below them (Figure 1).

Each of the 2 arithmetic expressions involves 1, 2, 3 or 4 terms and it should be random (i.e. not hardcoded, or fixed, neither for the terms nor for the operations). Every time the

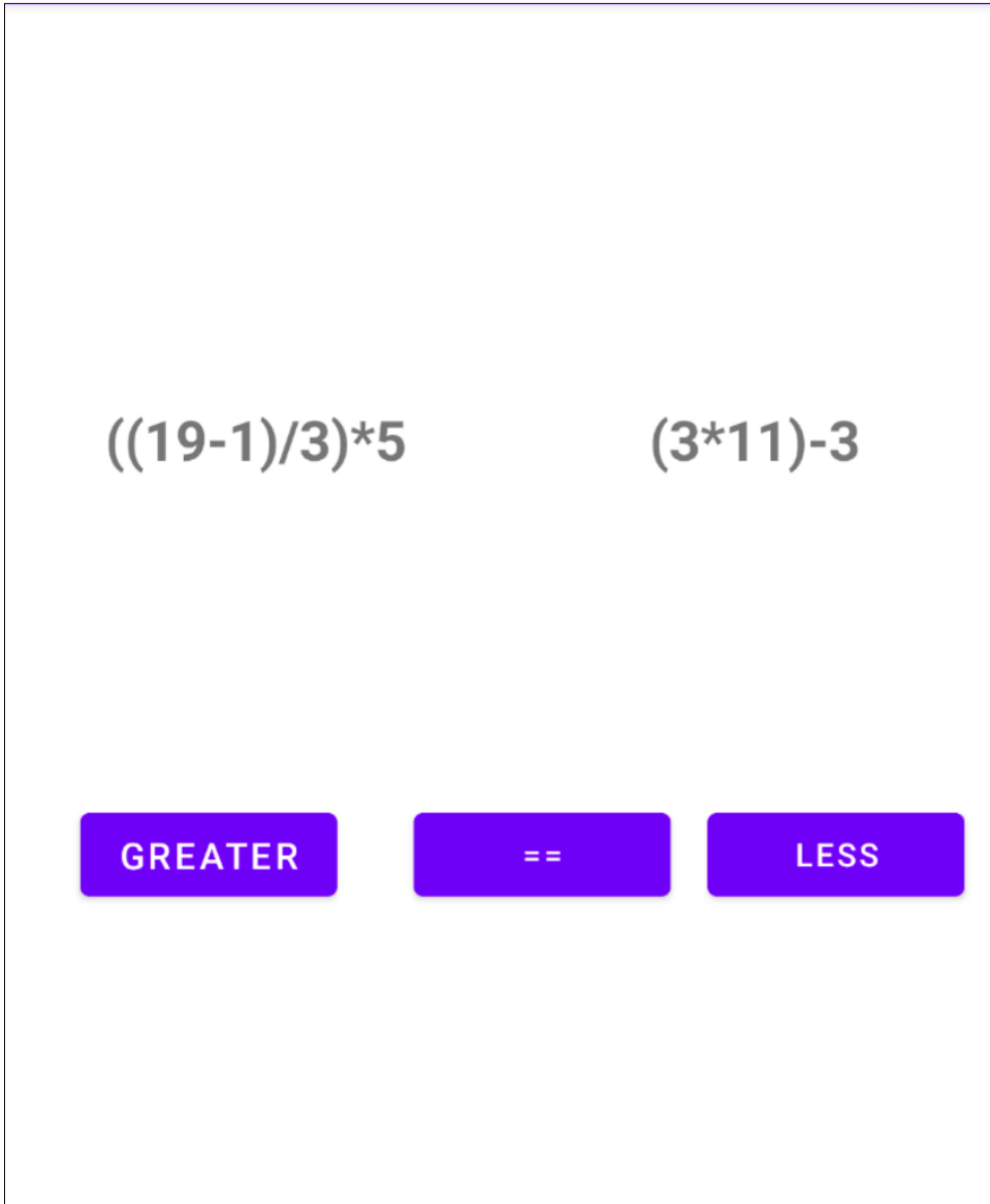


Figure 1: How the Game Screen looks like.

game screen reappears, the user is displayed with 2 new different arithmetic expressions. The 2 arithmetic expressions could be of different lengths, e.g. the left could have 2 terms and the right one 3 terms. The next time that a new game screen appears to the user the left expression might involve 4 terms and the right expression 2 terms.

The operations allowed between terms are addition (+), subtraction (-), multiplication (*) and division (/).

The range of the random numbers (the random numbers are generated dynamically based on a random generator) should be in the between 1 and 20 inclusive.

Both the terms and the operations must be random and different every time that the user is presented with a new game screen.

(15 marks)

4. The user needs to indicate whether the evaluation of the expression on the left is greater than, equal or less than the evaluation of the right expression, by pressing the buttons **Greater**, **==** or **Less** respectively.

The order of evaluation is always left to right and this is indicated by “compulsory” parentheses if the arithmetic expression involves more than 2 terms (i.e. 3 or 4 terms). An expression with 1 or 2 terms should NOT have any parentheses.

Examples of valid arithmetic expressions are:

2*3
(6+4)/2
((10-2)*3)/4

Examples of invalid arithmetic expressions are:

(2*3) no parentheses for 2 terms, i.e. should be 2*3
6+(4/2) parentheses should be placed from left to right, i.e. should be (6+4)/2
(10-2)*3/4 parentheses are missing, i.e. should be ((10-2)*3)/4

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.

(15 marks)

5. Once the user submits an answer, the application displays 2 different random arithmetic expressions and the user can interact with the game again.

(8 marks)

6. Modify the application so that the displayed arithmetic expressions involve sub-expressions which evaluate to whole numbers (i.e. integers). This means that evaluating any part of the expression from left to right always results in an integer.

For example, (19-1)/3 is valid because (19-1) equals to 18 which is divisible by 3 (i.e. 3 is a factor of 18). In other words if we divide 18 by 3 the remainder is 0.

(19-1)/4 is would not be valid because 18 is not divisible by 4 (i.e. 4 is not a factor of 18). In other words, if we divide 18 by 4 the remainder is not 0.

(14 marks)

7. Modify the application so that the displayed arithmetic expressions involve sub-expressions which do not evaluate above 100. This means that evaluating any part of the expression from left to right always results in a number less or equal than 100.

(6 marks)

8. Extend the application so that it includes a count down timer counting from 50 seconds down to 0, every tick occurring after 1 second exactly.

As soon as the counter reaches the value of 0, the game stops and the application displays how many correct and incorrect answers the user has given. At this point the game is not playable any more. To play again the user must press the **Back** mobile device button to return to the main screen (displaying the **New Game** and the **About** buttons).

(12 marks)

9. Extend the application (assuming that you implement the countdown timer question) so that every time that the user gets 5 answers correct, 10 extra seconds are added to the timer. For example, if the countdown timer displays 20 seconds as remaining and at that point the user completes 5 correct answers the counter goes up to 30 seconds. Then again, while the timer shows 9 seconds as remaining, the user completes another 5 correct answers (they do not have to be 5 consecutive correct answers) the timer goes up to 19 seconds.

(8 marks)

10. 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 the user was seeing before the rotation (e.g. the same arithmetic expression should be displayed, the timer should continue its countdown from the same second that it was before the rotation, etc.)

(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 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: Tuesday 15th of March 2022, 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->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.2021 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 fitzregistry@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.