COMP7506 Smart phone apps development
Individual Assignment

Deadline: **on or before 5th Nov., 2017 (Sun) 11:55pm.**
Late penalty: Marks will be deducted by N*2%, where N is the number of days after submission deadline (minimum marks = 0).
Weighting: 20% (of the whole course)

Background:

You are asked to develop an Android application to practice students' ability in solving algebraic equations. The application should focus on linear and quadratic equations. Your application should be in the form of a quiz in which 10 random questions are generated in sequence. The first 5 questions are about linear equation while the next 5 questions are about quadratic equation. Details are as follows:

1) Linear equation: Given a linear equation Ax + B = 0 where A and B are two random integers in the range [–99, 99]. The student will be asked to provide a value of x. Two sample questions are as follows:
   Sample question 1: "35x + 70 = 0, what is x?" (correct answer: x = –2)
   Sample question 2: "–16x – 8 = 0, what is x?" (correct answer: x = –0.5)
   Note: If B is negative, the symbol before B should be adjusted accordingly. That is if A = 2 and B = –3, your linear equation should be displayed as 2x – 3 = 0 instead of 2x + –3 = 0.
2) Quadratic equation: Given a quadratic equation $Ax^2 + Bx + C = 0$ where A, B and C are three random integers in the range [–99, 99]. The student will be asked to provide value(s) of x. Two sample questions are as follows:
   Sample question 1: "2x^2 – x – 78 = 0, what is x?" (correct answer: x = 6.5 / –6)
   Sample question 2: "x^2 + 18x + 81 = 0, what is x?" (correct answer: x = –9)
   Note 1: If B or C is negative, the symbol before B or C should be adjusted accordingly. That is if A = 2, B = –3 and C = –4, your linear equation should be displayed as 2x – 3x – 4 = 0 instead of 2x + –3x + –4 = 0.
   Note 2: Please make sure that the generated quadratic equation has real roots (that is determinant $B^2 – 4AC >= 0$). You may need to repeat the generation process until the generated equation has real roots.
   Note 3: You may add the following to prompt the user to round their answers to 2 decimal places in case their answers are not integers: "If your answers are not integers, please round them to 2 decimal places."

Implementation Details:

You should develop your Android application by following the steps below:

1) The application should contain 2 pages. The first page should contain your name and university number for identification purpose. This page should also provide a button "Start Quiz" for starting the quiz. The second page is the quiz main frame which will be loaded when the "Start Quiz" button on the first page is pressed.
2) On the second page, you should design a layout which contains:
   - 1 TextView field for displaying the question
   - 2 EditText fields for inputting the answer (Note: If the question is about linear equation or if the question is about quadratic equation but there is only one root, one of the EditText fields should be disabled.)
   - 1 Button for submitting the answer
   - 1 TextView for displaying the correctness of the input answer and the correct answer
   - 1 Button for proceeding to the next question
3) Construct a random question generator. For linear equation, the generator should pick A and B at random from the range [–99, 99]. For quadratic equation, the generator should pick A, B and C at random from the range [–99, 99] but please make sure that the generated equation has real roots (that is determinant $B^2 - 4AC >= 0$). You may need to repeat the generation process until the generated equation has real roots.
4) Calculate the correct answer(s). For the linear equation $Ax + B = 0$, the answer can be calculated as –B / A. For the quadratic equation $Ax^2 + Bx + C = 0$, the answers can be calculated as $\frac{-B \pm \sqrt{B^2 - 4AC}}{2A}$.
5) Display the generated question in proper format. You can display "x^2" as "x^2" for simplicity.
6) Accept one or two numerical answers from the user. Proper validation (e.g. the answer must be a number and cannot be a character) should be carried out. Error messages should be displayed if the answer is not of proper format. However, the user is allowed to leave the answer blank (i.e. give up this question and proceed to the next). For quadratic equation question, the user's answer is considered to be incomplete if only one root is provided. Also if the answers are not integers, please ask the user to round the answers to 2 decimal places.

Hint: You may try to catch Java exceptions in order to test whether an input answer is of proper format or not. For example, the following try-catch block can test whether `String S` contains an integer.

```
int i;
try {
     i = Integer.parseInt(S);
} catch (NumberFormatException e) {
     // display warning message
}
```

7) Compare the input answer with the correct one. Inform the user about the correctness of his/her answer. If his/her answer is wrong, tell him/her the correct one.

   Note: For double values A and B, they should be treated as equal if their difference is small enough. In Java, you can have something like the following in your program. Here A and B are treated as equal is their difference is small than 10^{-6}.

```
boolean areEqual(Double A, Double B) {
     if (Math.abs(A - B) < 1e-6)
          return true;
     return false;
}
```

8) Let the user proceed to the next question and repeat step 3) to step 7) above.
9) After completing 10 questions, display a summary telling the user the number of questions that he/she has answered correctly, wrongly and given up. The summary should also inform the user the average time he/she spent on each linear equation question and that on each quadratic equation question. This summary can be in form of a new page or just a TextView on the quiz page.

   Hint: The following can return the duration (in milliseconds) between two points of time A and B:

```
Long timeA, timeB;
```

At time A, run `timeA = System.currentTimeMillis();`

At time B, run `timeB = System.currentTimeMillis();`

```
Long Duration = timeB - timeA;
```

Marking Scheme:

Marks will be given based on the above descriptions [2% for each step + 2% for readme file (to be described below) = 20%]. So you are advised to complete the Android application with the readme documentation by following the steps above.

Marks will be given to each of the above steps. You are advised to complete the Android application by following the steps above. Each step will be worth 2% of marks.

Bonus marks: At most 2% marks of this assignment will be given to

1. Nice user interface.
2. Fluent quiz flow.
3. Additional features (e.g. audio, etc.).

NOTE:

i) The bonus marks are given relatively by comparing the works of students in the class.

ii) The upper bound of the marks in this assignment is 20%. Suppose a student gets 19% marks plus 2% bonus marks, the student will only get 20% instead of 21% in the final marks.

Submission:

You should submit a zipped or 7zipped file, which contains the followings on or before the deadline using Moodle system. Furthermore, the file should be named as "<your uid>.zip or "<your uid>.7z":

1. A readme documentation (DOC or PDF format) which contains information like how to open your project (e.g. API Level, AVD setting), limitation, reference, etc.
2. Source project folder, which includes manifest file, layout file, images and Java source codes. Please make sure the submitted project can be compiled in the Android Studio 2.1.2 using API Level 24 with minimum API Level 21. Students may verify their submission using the Android Studio installed on PCs in HW312. Also make sure the generated APK file can be run properly in the AVD (Nexus S with resolution 800 x 480).