IMD 4008: Mobile User Interfaces – Design & Development – Fall 2019
Assignment 1: Android Calculator

Due Date: Friday, October 4, 11:59pm (midnight).
Late Submissions: Accepted up to 4 days late at a 25% per day penalty.
Grading: This assignment is worth 6% of your final grade.
Group Work: This assignment can be completed alone or in pairs.

Introduction

For this assignment, you will build a simple Android calculator that supports basic mathematical operations. An example is seen to the right, but your app does not need to look like this (hopefully it will be more visually appealing!).

Your calculator should run on API 26. This is what the TAs will test on when grading, so deviation from this may result in your calculator not working properly when they get it.

Main Features [15 marks total]

Your calculator should support at minimum the following capabilities:

- Press a number key to append a digit to the current number shown in the number field **[*2 marks*]**

- Press an operator to start an operation (how this functions exactly is up to you – carefully consider how you design your calculator's behavior/interaction) **[*3 marks*]**
  - It should support addition (+), subtraction (-), multiplication (*) and division (/)
  - Upon pressing an operator key, it should then await another number
  - Pressing the equals key will carry out the operation

- A "clear" key and a "back" key (you can call these what you want) **[*2 marks*]**
  - Clear will wipe out the entire number field and possibly the last operand entered, at your discretion (again, think carefully about how you design the behavior of your calculator)
  - The back key will delete the most recently entered digit or operand

- A decimal key (.) **[*2 marks*]**
  - Pressing this will add a decimal to the current number, and subsequent digits entered will be to the right of the decimal place (i.e., the calculator should support floating point math)

- The positive/negative key (+/-) **[*1 mark*]**
  - Pressing this will reverse the sign of the current number

- A history **[*2 marks*]**
  - Recent completed operations are recorded and presented on the screen (in my example, these are seen below the number field). It's up to you how this is laid out exactly (e.g., most recent first, most recent last, where it's positioned, the maximum number of entries it displays, and so on) but again, think about your app design carefully here. You should support at least the last five calculations (the upper limit is up to you, but obviously constrained by your layout).

- Rotation **[*1 mark*]**
    - Rotating the app should not lose the values entered (either in the number field or the history). Note that it may not be nicely laid out in landscape mode (but see Additional Features below)

- Nice Layout, Visual Design **[*2 marks*]**
    - Make the calculator look nice. Look into improving the visual style of Buttons (perhaps ImageButtons) and other Views you use.

Additional Features [5 marks total]

For full credit, implement one the following features.

- *Scrollable history*
    - The limit on the history above is mostly due to the constrained space of the calculator. Make it scrollable instead, so you can have a limitless number of stored previous calculations
- *Memory*
    - Add a new button to remember the current number, and a button to recall it later on.
- *Proper landscape view support*
    - Rotating the display can switch to a second landscape layout that properly presents a nice visual layout for the calculator
- *Parentheses* ( )
    - Add buttons and support for parentheses. Recall BEDMAS – the parentheses will influence how the calculations are carried out.
- *Formula Entry*
    - Rather than only displaying a single operand at a time, allow the user to enter an entire expression, displaying it as it's entered. Upon pressing the "=" key, the entire equation is calculated according to the order of operations (BEDMAS). Also add a radio button (or similar widget) to allow switching between "basic" mode (standard functionality) and formula mode.

Note that some of the above are easier than others, but all are equally weighted.

Other Notes:

- Calculator displays should not be directly editable (i.e., no use of the system keyboard for entry please!)
- Consider using a GridLayout (although this is not mandatory, it may make your life easier)
- Use good coding style, variable names, method decomposition, commenting, etc. Part of your grade **[*2 marks*]** will depend on this.

Submission Notes

Find your project folder on your hard drive. Zip the entire project folder and submit to the CULearn link by the date and time specified above. Include a comment with your submission (e.g., in the CUlearn submission form) that indicates which of the Additional Features you implemented. If you work with a partner, ensure the partner's name also appears in the submission field (only one partner need submit).

A grading rubric will be made available shortly on CULearn.