# Assignment: # 1 Mobile Math Calculator – Individual

## Overview:

This Mobile Math Calculator assignment is intended as an opportunity for students to demonstrate their knowledge of table layouts, regex validation, business classes and testing.

## Functionality:

Write a single screen Android application in portrait mode that allows a user to perform simple mathematical calculations in a **‘rolling’** fashion. Rolling with reference to a calculator’s functionality means that there is no order of operations. So, 2+2*3 becomes 4*3.

Each operation is executed, and an answer calculated by **either** a second operator being pressed, **or** the equals operator being pressed. Since only one operator is ever considered in the logic, this avoids needing to figure out order of operations. In addition, you do not need to provide ‘accumulation capability’ (i.e. when the “=” is pressed again without additional numbers being entered).

For full marks, the following functionality must be provided in either a switch statement or in separate functions within a business object. The Remove and Clear operations should stay in the UI logic):

| | | |
|---|---|---|
| Addition (+) | Division (/) | Decimal (.) |
| Subtraction (-) | Neg / Pos (+/-) | Remove Digit (←) |
| Multiplication (*) | Calculate (=) | Clear (C) |

In addition, to the application itself, you will need to create a testing document using the Testing Template provided. This will also help clarify the functional requirements.

## Layout Options:

Please see common layouts for calculators on-line and or that provided by your operating system in Applications. You must provide a keyboard (i.e. buttons in a grid) rather than one provided to applications by your device.

## Getting Started/Recommended Steps:

- Before you begin coding the assignment, create a testing document using the Testing Template provided. This will help clarify the functional and validation requirements of the application. Keep in mind that there are far more scenarios to be tested that you might initially imagine.
- Build out the UI and store any labels (including #’s and symbols) to your strings repository. Remember to lock the display field down so that nothing may be typed into it that does not come from your calculator’s limited keyboard.
- Create your business class and a method with a case statement to discover which operator and numbers are to be processed. For example, calc(leftNum,rightNum,operator).
- Use Regex to establish patterns that are valid for processing.
- Make the program recognize when division by zero is being attempted and replace the default message with NaN (Not A Number). Note to use NaN and not zero because zero is not the mathematically correct answer to division by zero.