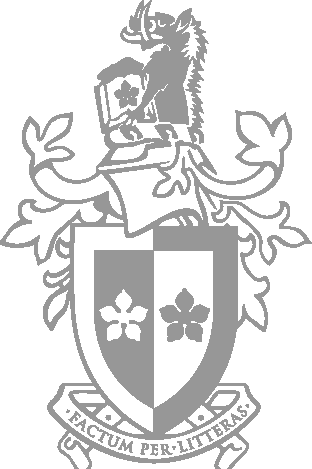
Faculty of Science, Engineering and Technology





**Interface Design and Development**

Distinction Task 1: Calculator Web App

**Overview**

Conditional and loop directive in AngularJS makes it possible to create interactive user forms. However, these are design to interactively update the DOM and not behave as a program control flow.

**Purpose:** Learn the use of conditional and loop directives.

**Task:** Add key-to-display functionality to the calculator web app created in Pass Task 2 (Component 2).

**Time:** This task should be completed before the start of week 6.

**Resources:**

* + - Lecture notes #3 and #4
    - AngularJS <https://angularjs.org/>

***Submission Details***

You must submit the following files to Blackboard:

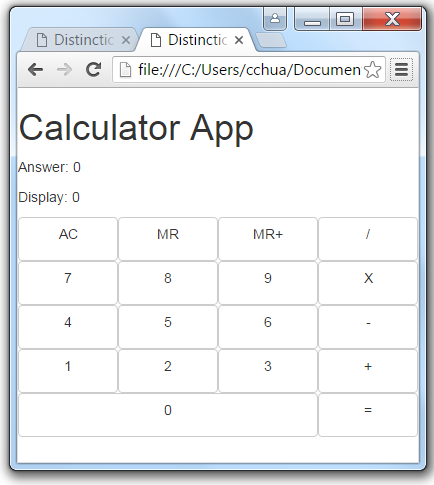
* **Calculator source code (calculator2.html).**
* **Calculator js file (calculator-app.js).**

Make sure that your task has the following in your submission:

* The calculator web application is HTML5 compliant.
* Demonstrates understanding of using AngularJS expression and conditional and loop directives.

**Instructions**

Without AngularJS, writing a web application with user interactive can be quite tedious. Using JavaScript will require a number of listeners to be hooked up to the input form elements in order to interactively update the view with the updated computed values. With AngularJS, the task can be achieved using model and expressions linked to the ng-click directive.

1. Copy the calculator.html from lab 2 as calculator2.html in lab04, or create a new calculator app. Only integers and basic arithmetic functions are support in this calculator app.
2. Open calculator2.html using Brackets, and add the Bootstrap and AngularJS scripts.
3. Modify the code to support key to display interactivity.
   * Add another HTML element to show the answer. The existing display will represent the number currently being entered.
   * Initialise a display, answer, memory and temporary variables that you need.
   * Replace the constant in your display with the display variable
   * Add ng-click to each key and necessary expressions to update the display accordingly
4. Implement the calculator-app.js file with AngularJS Module and Controller.

Web App: calculator2.html

————————————————

Uses: Bootstrap and AngularJS

---------------------------------

Model:- program variables

---------------------------------

Steps:

1: initialise all necessary variables

2: set up the calculator

3: link all keys to an ng-click

**Tip:** The answer and display variable are of a numeric data type, and each ng-click may contain several expressions that updates the variables.

You can only save the current operator function as a numeric value and not as an operator.

**Note:** This task is aimed at improving the understanding of how the view-model works. Using conditional and loop directives is a view functionality and not a controller functionality.

**This web application is entirely based on model and expressions. No controller is used as such it will have limited functionality. No conditional or loop directives are also used in this task.**

This task requires formulation of appropriate expressions to create the functionality of the calculator. The expressions will be very similar for all keys (both numeric and operator), however they will have to be hard coded as no controllers are used.

Once the task is complete you can submit it for assessment, which will help prepare it for your portfolio.

**Please add comments to your codes and apply your own additional bootstrap styling in the HTML file.**