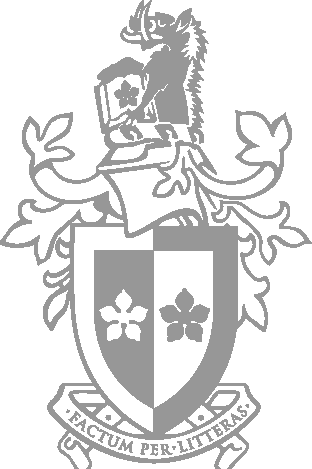
Faculty of Science, Engineering and Technology





**Interface Design and Development**

Credit Task 2: Creating Custom Filter & Directive

**Overview**

Module enables you to create custom filter and custom directives for your web application. In this task you will create a custom filter that will format a number to Roman numerals. On top of that, you will need to create a custom directive that will generate product information/card based on information assigned to its attribute.

**Purpose:** Learn how to use module, create a custom filter and custom directive.

**Task:** Create a web app that allows users to enter a number and display it as a Roman numeral. Uses a custom directive to initialise a model with product information and create a view that display the product information.

**Time:** This task should be completed before the start of week 7, but may be completed later (see following note).

**Resources:**

* + - Lecture notes #3, #4 and #5

**Note:** It If you are not currently up to date you should skip this task and return to it once you are up to date with the Pass Tasks. Do not allow Credit Tasks to delay you in keeping up with the unit's Pass Tasks.

***Submission Details***

You must submit the following files to Blackboard:

* HTML source code (romannumber.html & appproduct.html).
* Custom filter and custom directive source code (appromannumber.js & appproduct.js).
* Screenshot of the web app.

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

* Both web applications is HTML5 compliant.
* Demonstrates understanding in using the AngularJS framework.
* Demonstrates use of AngularJS module, custom filter and custom directive.



Component 1: Creating Custom Filter

**Instructions**

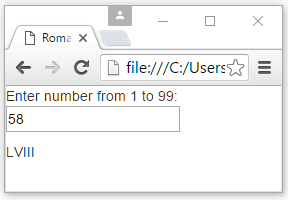
Implement the Roman numeral conversion web app. In this web application, you will need to implement a filter to return the Roman numeral equivalent on integer from 1 to 99.

1. Start by creating a new HTML file in Brackets.
2. Implement the basic outline of an AngularJS web app with the appropriate scripts.

**Note:** It is a good practice to write the controller in a separate file.

<script src="js/appromannumber.js"></script>

1. The HTML file will have
   * User input for the user to enter an integer from 1 to 99
   * View that shows the Roman numeral equivalent of the number



1. Create the module file appromannumber.js with the filter template.

Web App: **appromannumber.js**

————————————————

var app = angular.module("myApp", []);

app.filter ("<your filter name>", function(){

return function(<your parameter>) {

var <return variable> = <initial value>;

:

<your code here>

return <return variable>;

}

;// semi colon for return statement

});

1. Your web app should now be complete. Make sure that you have added comments in your code and a proper bootstrap styling then test it on the browser to make sure that it works as you expect.

Component 2: Creating Custom Directive

**Instructions**

In this web application, you will need to implement a custom directive (element) that takes two attributes value and use it to generate a product card.

1. Start by creating a new HTML file in Brackets.
2. Implement the basic outline of an AngularJS web app with the appropriate scripts.

**Note:** It is a good practice to write the controller in a separate file.

<script src="js/appproduct.js"></script>

1. The HTML file will use the directive as an element named **product** with two attributes, **model** and **price** as shown below.

<**product** **model**="Apple IPhone XS" **price**="4999"></**product**>

1. Create the module file appproduct.js with the directive template.

Web App: **appproduct.js**

————————————————

app.directive('**product**', function($parse) {

<your code, you may either write the object with the return statement, or create an object and return it.>

var <obj> = {};

var <function object> = function(scope,element,attributes){

<initialises the model>

};

<obj>.restrict = '<your restrict value>';

<obj>.link = <function object>;

<obj>.template = '<your template>';

<obj>.scope = {} //for isolated scope

return <obj>;

});

**Note:** Use restrict 'E'. The directive will produce and render phone information in a bootstraps column with a header and paragraph.

<div class='col-md-3'>  
 <h4>{{model}}</h4>  
 <p><b>Price:</b> RM {{price}}</p>  
</div>

1. Your web app should now be complete. Make sure you test it on the browser to make sure that it works as you expect. Feel free to add the phone images. Also please add

