**COMP 3123 – Full Stack Development – Lab 10**

* Angular Data Binding & Directives
* Building Custom Pipes & Directives

**Developer Note:**When working on your exercises, please use the same Angular project from previous labs. This way you way you won’t have to re-generate a new Angular application every lab.

Only submit the files components, typescript files that you have created and not the entire application.

**Exercise #1 – Directives and Data Binding**

\* If you have not previously completed part one of Angular Tutorial, complete it first.  
 <https://angular.io/tutorial/toh-pt1>

1. Complete Part Two of Angular Tour of Hero’s Tutorial on Structural Directives and Data Binding.

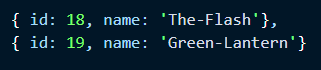
<https://angular.io/tutorial/toh-pt2>

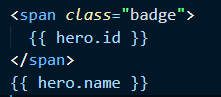
**Exercise #2 – Creating a Custom Pipe with Angular CLI**

1. Use the following ***Angular CLI*** command to generate a ***Custom Pipe*** named ***remove-spaces***



1. Review the generated boiler plate code and navigate to the ***app.module.ts*** file and review the updated ***import*** statements and **@NgModule declarations** for the generated ***remove-spaces Pipe***.
2. Update the ***mock.heros.ts*** file from Exercise #1. Add the following list items to the array of Heros



1. Write the code in the Custom Pipe to replace any occurrence of a dash ‘-‘ with an empty spaces.
2. Once completed apply it the hero name in the file ***heros.component.html*** to filter out the dashes in names  
   

**Exercise #3 – Creating a Custom Directive with Angular CLI**

1. Use the following ***Angular CLI*** command to generate a ***Custom Directive*** named ***inputFormat***
2. Review the generated boiler plate code and navigate to the ***app.module.ts*** file and review the updated ***import*** statements and **@NgModule declarations** for the generated ***input-format Directive***.
3. Create a ***textbox input*** element in the HTML markup that has the ***input-format*** Directive as an attribute.  
   
4. In the generated ***input-format.directive.ts*** file implement the HostListener interface and ElementRef class  
     
   
5. Implement the ***HostListener*** interface and add logic such that when the User clicks away from the textbox (OnBlur event) the textbox text will be converted to UpperCase text.  
     
     
   

**Homework: Nested Components**

* Complete Part Three of Angular Tour of Hero’s Tutorial on Nested Child Components.

<https://angular.io/tutorial/toh-pt3>