Angular
Lab 1: Data Binding
Exercises

1 Lab setup

2 Generating a new Angular project

3 General structure of an Angular app

4 Template expressions and interpolations

Q1: Add in a property to the root component called `myAge` and give it any random value

Q2: Display the value of this property with a random string (for e.g. `I am 60 years old`) in the template HTML

Q3: Add in a method called `getAgeMessage` which returns a string depending on the value of `myAge`. The content of the string returned is:

`myAge` less than 30: I am sooo young
`myAge` between 31 and 50 : I am middle aged
`myAge` more than 51: I am soooo old.

Q4: Using this method, display a message similar to the following in the template:
`I am 40 years old, and I am middle aged`
Change the value of `myAge` to check that the correct messages are displayed for the correct age.

Q5: Add in a property called `socialMediaSite` and set it to `https://www.tiktok.com/`

Q6: Add in another method called `getOpenStyle` which returns one of the possible vales for the target attribute to open a linked document:

This method should accept a single number and return a string depending on the value of the number

If number is 0, return `_self`
If number is not 0, return `_blank`

Q7: Using this method and property, create an <a> element in the root template which will either open the linked site in the same window/tab or in a different window or tab

5 Angular app scripts and the DOM

6 Property binding

Q1: Add in two properties `socialMediaSite` (which holds any random valid URL) and `myOpenStyle` (which holds the value of either `_self` or `_blank` (see Q6 from Topic 4) to the root component

Q2: Add in an <a> element into the template which uses these two properties in the syntax of property binding. Check that it works properly.

7 Class binding

First Part: (after `app.component-v3.ts, app.component-v3.html`)

Q1: Create a element and nest a <p> with some text within this . The CSS Box model provides a set of properties to delineate an element and its child elements. You can use a variety of properties to control the border look for the . Set a suitable `padding` for the .

Q2: Add 2 different classes in the template CSS which sets different values for the `border-style` property, another 2 different classes which sets different values for the `border-color` property, and another 2 different classes which sets different values for the `border-width` property. Give these classes any suitable names.

Q3: Add any one of these 6 classes to the , and then use single binding to add another different class to the , depending on a Boolean property in the component.

Q4: Add any 2 of these 6 classes to the , and then use single binding to remove one of these classes, depending on a Boolean property in the component.

Second Part:

Reuse the same and classes from First Part: Q1 and Q2

Q5: Add any one of the 6 classes to the , and use a string to add another 2 more of these classes to the

Q6: Add any one of the 6 classes to the , and use an object to add another 2 more of these classes to the while removing the existing class.

Q7. Add any one of the 6 classes to the , and use an array of strings to add another 2 more of these classes to the

8 Style binding

Q1: Create a <p> element and use single style binding to set its `font-family` property.

Q2: Create a element and nest a <p> with some text within this . Use a string containing the relevant properties to style the

Q3: Create another element and nest a <p> with some text within this . Use an object containing the relevant properties to style this

9 Event binding

Q1: Add in 3 radio buttons, each with different values.

Q2. Provide event binding for the click event for all these 3 radio buttons to the same component method

Q3. Implement this component method so that it logs the value of the particular radio button clicked.

9.1 Template reference variables for event binding

Repeat Q1 - Q3 from the previous section, but implement the event binding using template reference variables instead.

9.2 Combining interpolation, property, class and event binding

Using any 2 or more HTML elements of your choice (text field, normal button, radio button, check box, paragraph, etc) create a scenario of your design that combines interpolation, property, class and event binding.

10 Two way binding with ngModel

Create a radio button. Use banana-in-the-box syntax to perform 2 way binding between the button value and a component property.