

# Coding Bootcamp for Deaf Learners

Code College is delivering a Coding Bootcamp for 24 Deaf learners via the Deaf Empowerment Firm and the National Council for Persons with Disabilities.



---

## Code College now an Oracle Workforce Development Partner

Code College has joined the Oracle Workforce Development Program. This will give us access to the latest learning products and support in our [Java Bootcamp](#). It also adds exposure to our students who pass the Oracle Certification Exams.

---

# 2 -6 Oct Java Beginner

## Prerequisites / Further Training

You should not be a complete beginner for this course. If you cannot [pass this test](#), you must do [Intro To Programming](#) first.

Recommended sequence	Summary
<i>Beginner Java</i>	<i>Java Language and OO Fundamentals</i>
<a href="#">Advanced Java (Full-Stack)</a>	Building Apps with Java
<a href="#">Java Servlets Beginner</a>	Java Web development

Also have a look at our [Java Bootcamp](#)

## Alignment

# OCA: Oracle Certified Associate Java SE 8 Programmer Exam 1Z0-808

## Intended Audience

- Intended for people who has some knowledge of programming and wanting to learn Java and OO
- **NB:** This is NOT an [Introduction to Programming](#) course.

## After this course you should be able to

- Have a good understanding of programming and the building blocks of an OO programming language, with an emphasis on JAVA.
- Prepare for Oracle OCA Exam 1Z0-803
- Proceed to the [Advanced Java](#) Training Course

## Course Material

Course Material Provided

## **Course Contents**

### **Day 1**

#### **Introducing Java Technology**

- Breaking the Surface
- The way Java works
- Code Structure in Java
- Anatomy of a class
- The main() method
- Netbeans IDE and Debugging

#### **Loop and decision constructs**

- Looping
- Conditional branching
- A trip to Objectville
- Inheritance,Overriding
- Class variables and methods
- Making your first object,Using main

### **Day 2**

#### **Primitives**

- Know your Variables
- Declaring a variable
- Primitive types
- Java keywords

#### **Objects**

- Reference variables
- Object declaration and assignment
- Objects on the garbage collectible heap
- Arrays
- How Objects Behave
- Methods use object state
- Method arguments and return types
- Pass-by-value

## Encapsulation

- Getters and Setters
- Encapsulation
- Using references in an array

## Day 3

### Arrays and ArrayLists

- Extra Strength Methods
- Building a one-dim ArrayList game
- Preparing to code
- Coding
- Random numbers<
- Using user-input
- For loops
- Casting primitives
- String conversion
- Using the Java Library
- Two Dimensional ArrayList Structures
- Enhancing the game
- Coding the game
- Boolean expressions
- Using the Java library (API)
- Using packages
- Using the HTML API docs and

## Day 4

### Polymorphism. Method Overloading

- Better Living in Objectville
- Understanding inheritance
- Designing an inheritance tree
- Avoiding duplicate code
- Overriding methods
- IS-A and HAS-A · What do you inherit from your superclass?

- What does inheritance really buy you?
- Polymorphism
- Rules for overriding
- Method overloading

## **Advanced OO Concepts**

- Serious Polymorphism
- Some classes should not be instantiated
- Abstract classes
- Abstract methods
- Polymorphism in action
- Class Object
- Taking objects out of an arraylist
- Compiler checks the reference type
- Get in touch with your inner object
- Polymorphic references
- Casting an object reference (moving lower in the inheritance tree)
- Deadly Diamond of Death
- Using interfaces (the best solution

## **Day 5**

### **Garbage Collection**

- Life and Death of an Object
- The stack and the heap
- Methods on the stack
- Where local variables live
- Where instance variables live
- The miracle of object creation

### **Constructors**

- Constructors, Initializing state of a new Object
- Overloaded constructors
- Superclass constructors
- Invoking overloaded constructors using this()

- Life of an object, Garbage collection

## Handling Errors

- Handling Errors
- Handling Exceptions

## Duration and pricing

In [Price Group B](#)

## Certificate

1. Upon completion of this course we will issue you with attendance certificate to certify your attendance and / or completion of the prescribed minimum examples.
2. You may sit for our competency assessment test and on passing you will obtain our competency certificate.
3. Our competency assessment can be booked and taken by someone who has not attended the course at a cost of R950.

## Bookings

You can download the course registration form on our home page or by clicking [here](#)

## Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

## Questions

Please [email us](#)

## Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page.

---

# 18 – 22 Sept : MEAN Stack (Mongo, Express, Angular, Node.JS)

Our MEAN Stack (Mongo, Express, Angular, Node) training course teaches you how to develop web applications using the MEAN stack. A powerful platform that uses only one language, top to bottom: JavaScript.

Prerequisite

- [Angular](#)
- [Node.js](#)

## DAY 1 – INTRODUCTION

- Introducing full-stack development
- Designing a MEAN stack architecture

## DAY 2 – BUILDING A NODE WEB APPLICATION

- Creating and setting up a MEAN project
- Building a static site with Node and Express
- Building a data model with MongoDB and Mongoose
- Writing a REST API: Exposing the MongoDB database to the application
- Consuming a REST API: Using an API from inside Express

## DAY 3 – ADDING A DYNAMIC FRONT END WITH ANGULAR

- Adding Angular components to an Express application
- Building a single-page application with Angular:
  - Foundations
  - The next level

## DAY 4 – MANAGING AUTHENTICATION AND USER SESSIONS

- Authenticating users
- Managing sessions
- Securing APIs

## DAY 5 – YOUR OWN PROJECT

### **Duration and pricing**

- *Full-time* over 5 days (R9995)
- *Part-time* over 4 weeks (2 nights per week, 3 hour sessions) (R11995)
- *Part-time* over 8 Saturdays, 3 hour sessions (R11995)
- Please note : For *part-time* courses we do not have a fixed schedule and you will be placed on a waiting list until we get a group of 4+ together. Please book with no dates on the bookings form. This will automatically put you on the waiting list. We will confirm with you as soon as we have a part-time group together.
- [Distance-learning](#) over up to 3 months (R9995)
- International exams are not included in the course price.
- Prices exclude Vat for Vat-registered companies

### **Certificate**

- Upon completion of this course we will issue you with attendance certificate to certify your attendance and / or completion of the prescribed minimum examples.
- You may sit for our competency assessment test and on passing you will obtain our competency certificate.
- Our competency assessment can be booked and taken by someone who has not attended the course at a cost of R2950.

### **Bookings**

You can download the course registration form on our home page or by clicking [here](#)



## Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

## Questions

Please [email us](#)

## Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page.

---

# PHP 7 vs. Node.JS ?

Node.js is a tool for back-ends creation. As well as PHP. But questions like “Which is better for back-end – PHP or Node js?” or even “Is Node js killing PHP?” or “Is NodeJS eating PHP Market?” are asked frequently. Let's be honest: Node.js couldn't kill PHP. Why?

<https://belitsoft.com/php-development-services/php7-vs-nodejs#.Wal2k77LKNM.linkedin>

---

# Angular 11 – 15 Sept

Our **Angular Training Course** is intended for front-end developers who are familiar with Javascript and want to master

Angular to move a gear up. Angular is the leading framework for building dynamic JavaScript applications that take advantage of the capabilities of modern browsers and devices.

## **Prerequisites**

[JavaScript](#)

## **Further training**

[Node.JS](#)

[MEAN Stack](#)

[Ionic](#)

## **Intended Audience**

Front end developers who found that JavaScript is inadequate to develop a rich, fully-fledged front-end. Our Angular Training Course starts with the MVC pattern and the many benefits that can be gained from separating your logic and presentation code. We then start from the nuts-and-bolts and building up to the most advanced and sophisticated features Angular, going in-depth to

## **Course material:**

# **DAY 1**

## **Getting Started**

- Plunker
- Intro to TypeScript .
- Writing our first app.
- String Interpolation
- Looping
- Property & Event Binding
- Domain Model
- Nesting Components & Inputs .

- User Interaction & Outputs

## **ES6 JavaScript & TypeScript Overview**

# **DAY 2**

## **Angular CLI**

### **Components**

- Architecting with Components
- Templates, Styles & View Encapsulation
- Content Projection
- Lifecycle Hooks
- ViewChildren & ContentChildren

### **Built-in Directives**

- NgIf & NgSwitch.
- NgStyle & NgClass .
- NgNonBindable
- Structural Directives.

# **DAY 3**

## **Custom Directives**

- HostListener & HostBinding
- Inputs & Configuration

## **Reactive Programming with RxJS**

- Streams & Reactive Programming
- Observables & RxJS .
- RxJS & Angular

## **Pipes**

- Built-in Pipes
- Async Pipe .

- Custom Pipes

## DAY 4

### Forms

- Model Driven Forms
- Model Driven Form Validation
- Submitting & Resetting
- Reactive Model Form
- Template Driven Forms

### Dependency Injection & Providers.

- Injectors
- Provider
- Tokens
- Configuring Dependency Injection in Angular
- `NgModule.providers` vs `Component.providers` vs `Component.viewProviders`

## DAY 5

### HTTP

- Core HTTP API
- HTTP Example with Promises .
- HTTP Example with Observables . . . .
- JSONP Example with Observables.

### Routing

- Route Configuration . .
- Navigation.
- Parameterised Routes
- Nested Routes .
- Router Guards
- Routing Strategies

## **Unit Testing Overview**

### **Duration and pricing**

In [Pricing Group A](#)

### **Certificate**

- Upon completion of this course we will issue you with attendance certificate to certify your attendance and / or completion of the prescribed minimum examples.
- You may sit for our competency assessment test and on passing you will obtain our competency certificate.
- Our competency assessment can be booked and taken by someone who has not attended the course at a cost of R2950.

### **Bookings**

You can download the course registration form on our home page or by clicking [here](#)

### **Brochure**

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

### **Questions**

Please [email us](#)

### **Schedule**

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page.

---

# Intro to Programming 28 Aug – 2 Sept

This Intro to Programming Course will teach you the fundamentals of coding. These topics that are common to all coding languages today: Variables, program flow, data structures, modularisation, functions and how to build basic solutions using these fundamentals. Although we are using JavaScript in this course, the emphasis is not on language, rather on generic coding fundamentals applicable to all coding languages today. Using JavaScript is a bonus because it is the most widely used coding language in the world and used by all other coding environments.

## Prerequisites

None. No Matric required, neither any previous training in computer programming for our Intro to Programming course. If you can pass this [Entry Test](#), you may skip this course before doing any Beginner course like [Beginner Java](#), [Beginner C#](#), [Beginner PHP](#) , [Beginner Python](#) [Beginner C++](#) and more or any of the [Bootcamps](#).

## Intended Audience:

Anybody who has not programmed before and wants to get into programming the right way

## Further Training

All [Coding Bootcamps](#) and Beginner Courses

## Course Material

Provided

## Course Info

## **Fundamentals**

- The way JavaScript works
- How you're going to write JavaScript
- How to get JavaScript into your page
- Statements
- Variables and values
- Loops
- Conditionals
- When you need to make LOTS of decisions
- Communication with your user
- A closer look at console.log

## **Design, QA**

- Coding a Serious JavaScript Application
- Let's build a game
- First, a high-level design
- Working through the Pseudocode
- Setting up the loop, getting some input
- Doing a little Quality Assurance
- Generating a random number
- Getting functional

## **Functions**

- JavaScript is pass-by-value.
- Weird Functions
- Functions can return things too
- Tracing through a function with a return statement
- Global and local variables
- Scope of local and global variables
- Don't forget to declare your locals!

## **Data Structures**

- Arrays
- How to represent multiple values in JavaScript
- How arrays work

- How to access an array item
- Updating a value in the array.
- How to iterate over an array
- A better way to iterate over an array
- Loop with the post-increment operator
- Creating an array from scratch (and adding to it)

## Objects

- Object Orientation
- How properties work
- How does a variable hold an object?
- Comparing primitives and objects
- Pre-qualification
- Passing objects to functions
- Add behavior to your objects
- When method does not know about a property
- How does behavior alter state

## DOM

- Getting to know the DOM
- How JavaScript really interacts with your HTML page
- Getting an element with getElementById
- What, exactly are we getting from the DOM?
- Finding the inner HTML
- What happens when you change the DOM
- Don't even think about running the code until the page is fully loaded!
- Callbacks
- Setting an attribute with setAttribute (you can GET attributes too)
- So what else is a DOM good for?

## Types

- Serious types
- How to use null
- Dealing with NaN



- Understanding the equality operator (otherwise known as ==)
- How equality converts its operands
- Type conversions
- How to determine if two objects are equal
- Truthy and Falsey

## **Strings**

- How a string can look like a primitive and an object
- String methods (and properties)
- Building an example app from scratch
- How to design the game
- The View
- The Model
- Implementing the model object
- Setting up methods
- The Controller
- Passing the input to the controller

## **Events**

- Handling events
- What are events?
- What's an event handler?
- How to create your first event handler
- Getting your head around events... by creating a game
- Implementing the game
- Assign an event handler to an image's onclick property
- How to reuse the same handler
- How the event object work
- The event object and target
- Events and queues
- How setTimeout works

## **Projects, Exercises, Quizzes and Assessments**

## **Duration and pricing**

## [Pricing Group C](#)

### **Certificate**

Read about our [certificates](#)

### **Bookings**

You can download the course registration form on our home page or by clicking [here](#)

### **Brochure**

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

### **Questions**

Please [email us](#)

### **Schedule**

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page.

---

# **21-25 Aug Adv Java**

## **Prerequisites / Further Training**

- [Java Beginner](#)
- [Beginning SQL](#)

Also have a look at our [Java Bootcamp](#)

## **Alignment**

Oracle OCP Certification aligned to Oracle OCP Java Exam

## **After this course you should be able to**

- Have a good understanding of programming and the building blocks of an OO programming language, with an emphasis on JAVA.
- Build small apps in Java, making use of I/O, Networking, GUI
- Prepare for Oracle OCA and OCP exams

## **Course Material**

Course Material Provided

## **Course Contents**

### **Day 1:**

#### **Static Classes**

- Static Methods,
- Static Variables
- Math Class
- Math Methods
- Static Imports

#### **Wrapper classes**

- Autoboxing
- Number formatting

### **Day 2**

#### **Exception Handling**

- What if you need to call risky code?
- Checked Exceptions
- Flow control in try/catch blocks
- Exceptions are polymorphic
- When you don't want to handle an exception...

- Ducking (by declaring) only delays the inevitable

## **Nested Classes**

- Getting GUI: A Very Graphic Story
- Getting a user event
- Listeners, Sources, and Events
- An inner class instance must be tied to an outer class instance
- How to make an instance of an inner class

## **Day 3:**

### **I/O Fundamentals**

- Saving Object state
- Implement Serializable
- Transient variables

### **Highlights of the Calendar API**

- Getting an object that extends Calendar
- Working with Calendar objects

### **Date / Time API**

- Moving backward and forward in time

## **Day 4:**

### **Networking**

- Connecting, Sending, and Receiving
- Network Sockets
- Writing a simple server
- Writing a Chat Client

### **Threads & Concurrency**

- Thread class
- Runnable interface
- The Thread Scheduler

- Putting a thread to sleep
- Atomic methods
- Using an object's lock
- Synchronization

## Day 5:

### Collections and Generics

- Collections
- Sorting an ArrayList with Collections.sort()
- Generics and type-safety
- Comparable interface
- A Custom Comparator
- Collection API : lists, sets, maps
- HashSet
- HashMap
- Polymorphism wildcards

### Packaging

- Command-line Java
- Classpath
- Jars

### Duration and pricing

- *Full-time* over 5 days (R9995)
- *Part-time* over 4 weeks (2 nights per week, 3 hour sessions) (R11995)
- *Part-time* over 8 Saturdays, 3 hour sessions (R11995)
- Please note : For *part-time* courses we do not have a fixed schedule and you will be placed on a waiting list until we get a group of 4+ together. Please book with no dates on the bookings form. This will automatically put you on the waiting list. We will confirm with you as soon as we have a part-time group together.
- [Distance-learning](#) over up to 3 months (R9995)
- International exams are not included in the course

price.

- Prices exclude Vat for Vat-registered companies

## **Certificate**

1. Upon completion of this course we will issue you with attendance certificate to certify your attendance and / or completion of the prescribed minimum examples.
2. You may sit for our competency assessment test and on passing you will obtain our competency certificate.
3. Our competency assessment can be booked and taken by someone who has not attended the course at a cost of R950.

## **Bookings**

You can download the course registration form on our home page or by clicking [here](#)

## **Brochure**

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

## **Questions**

Please [email us](#)

## **Schedule**

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page.

---

# 7 Aug Mobile Web Development

Professional Mobile Web Development using mainly HTML5, CSS3 and JavaScript. Learn how to build lean and fast performing web mobile apps like native apps

## Prerequisites

Before attempting Mobile Web Development Programming on its own, you should have previous web design and development experience and knowledge of HTML, CSS and basic Javascript.

## Contents:

### Day 1:

#### Responsive Web Design

- Get on the mobile bandwagon
- The recipe for Responsive Web Design
- An example of a responsively designed site
- CSS media queries
- Analyze the current CSS
- Steps to creating the mobile-specific CS

#### Responsible Responsiveness

- Mobile-first Responsive Web Design
- Can I get a proxy to set up my proxy?
- What to do when things aren't blazing fast
- Find the drags on page speed
- It looks mobile friendly, but it isn't
- What is progressive enhancement?
- Mobile-first media queries
- Add the map back using JavaScript
- Move iframe attributes to CSS equivalents
- Breakpoints to the rescue

### Day 2:

## **A Separate Mobile Website**

- Creature Comforts has agents in the field
- Getting to know user agents
- Straight talk: Most major sites have a separate mobile website
- When what you really want to do is (re-)direct
- Make a mobile mockup
- Not all phones are smartphones...not by a sight
- Let's keep it basic: Meet XHTML-MP
- Access keys in action
- Mobile-savvy CSS

## **What devices should we support?**

- How do you know where to draw the line?
- Step away from the keyboard for a second
- Things you don't support vs. those you can't support
- Ask questions about your project
- Ingredients for your magic mobile potion
- Draw from your cupboard of tools and data

## **Day 3:**

### **Device Databases and Classes**

- A panic button for freaked-out students
- The button is for mobile phones only
- WURFL and its capabilities
- WURFL: Clever API code
- Steps for building our explore page
- Use WURFL to help differentiate content
- Make the page a bit smarter with WURFL
- The panic button: For phones only
- Expanding a lucrative part of AcedIt!'s business
- Get acquainted with the matching function
- Make something actually happen with device classes
- We need a bigger safety net



## **Build a Mobile Web App Using a Framework**

- HTML5 is a specific thing...
- How “traditional” websites typically behave
- A Games Unlimited mobile HTML5 web app
- The master plan for phase 1 of the Game
- Why use mobile web app frameworks?
- Our choice for the Game: jQuery Mobile
- Build a basic page with jQuery Mobile
- Link to multiple pages with jQuery Mobile
- Make the Game feel more applike: to-dos
- Time to make that tartan-building form
- Build an HTML5 form
- Give jQuery Mobile hints about the fields

### **Day 4:**

#### **Mobile Web Apps in the Real World**

- Mobile apps in the real world
- Make a better form
- A widget to manage the list of colors and sizes
- The two sides of generate.php
- Offline is important
- A basic recipe to create a cache manifest
- Dev tools to the rescue
- How to ask W3C-compliant browsers where they are
- Let’s integrate geolocation

#### **Build Hybrid Mobile Apps**

- How do hybrid apps work?
- Bridge the web-native gap with PhoneGap
- Get acquainted with PhoneGap Build
- Keep track of discovered tartans
- Anatomy of the Tartan Hunt project
- What makes localStorage so special?
- Use a function to show which tartans are found
- Rope in PhoneGap to take pictures

- Now we're ready for the mediaCapture API

## **Day 5:**

### **How to Be Future Friendly**

- Time to dispel our collective illusions of control
- A future-friendly manifesto
- There are no silver bullets
- App today, web page tomorrow
- Remove PhoneGap references

### **Project**

- Develop a mobile app from start to finish

### **Duration and pricing**

Pricing [Group A](#)

### **Certificate**

1. Upon completion of this course we will issue you with attendance certificate to certify your attendance.
2. You may sit for our competency assessment test and on passing you will obtain our competency certificate.
3. Our competency assessment can be booked and taken by someone who has not attended the course at a cost of R2950.

### **Bookings**

You can download the course registration form on our home page or by clicking [here](#)

### **Brochure**

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

### **Questions**

Please [email us](#)

## Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page

---

# Angular 24 – 28 July

Our **Angular Training Course** is intended for front-end developers who are familiar with Javascript and want to master Angular to move a gear up. Angular is the leading framework for building dynamic JavaScript applications that take advantage of the capabilities of modern browsers and devices.

## Prerequisites

[JavaScript](#)

## Further training

[Node.JS](#)

[MEAN Stack](#)

[Ionic](#)

## Intended Audience

Front end developers who found that JavaScript is inadequate to develop a rich, fully-fledged front-end. Our Angular Training Course starts with the MVC pattern and the many benefits that can be gained from separating your logic and presentation code. We then start from the nuts-and-bolts and building up to the most advanced and sophisticated features Angular, going in-depth to give you the knowledge you need.

## **Course Material**

Provided in electronic format – we do have an extension monitor for you in our training room.

## **Course Info**

# **DAY 1**

## **HTML, JavaScript Review**

- HTML Elements
- HTML Attributes
- Attributes without values
- Attributes with Literal values
- HTML Element Content
- Document Structure
- Bootstrap Classes (Contextual, Margin, Padding, Element Sizes)
- Styling Bootstrap Tables
- Bootstrap Forms
- Bootstrap Grids (Responsive, Simplified)
- Example Project Preparation
- Script Element
- Functions
- Variables and Types
- JavaScript Operators
- Arrays
- Objects (Literals, Methods)
- Defining Classes
- Getters and Setters
- Class Inheritance
- Modules
- Importing Specific Types
- Renaming Imports
- Importing All of the Types in a Module
- Type Annotations
- Properties and Variables

- Specifying Multiple Types or any Type
- Using Tuples
- Indexable Tuples
- Access Modifiers

## **Building A Real App**

- Folder Structure
- NPM Packages
- Restful Web Service
- Index.html
- Root Component
- Root Module
- Bootstrap File
- Data Model
- Starting The Store
- Adding Store Features

## **Orders and Checkout**

- Creating The Shopping Cart
- Adding URL Routing
- Processing Orders
- Using RestFul Web Service

## **Admin**

- Module
- Routing System
- Navigating Admin URL
- Implementing Authentication
- Structuring the Admin Feature
- Implementing the Product Editor

# **DAY 2**

## **Data Bindings**

- Data Bindings (One-way, Property, Attribute, Classes,

Styles)

- Built-in Directives (ngIf, ngSwitch, ngFor, ngTemplateOutlet)
- One-way Data Binding Restrictions

## **Events and Forms**

- Event Binding
- Two-way Data Bindings (ngModel Directive)
- Form Data Validation
- Entire Form Validation
- Summary Validation Messages
- Disabling the Submit Button
- Model-Based Forms
- Form Model Classes
- Using the Model for Validation
- Generating the Elements from the Model
- Custom Form Validators

## **Attribute Directives**

- A Simple Attribute Directive
- Applying a Custom Directive
- Accessing Application Data in a Directive
- Reading Host Element Attributes
- Using a Single Host Element Attribute
- Creating Data-Bound Input Properties
- Responding To Input Property Changes
- Binding To a Custom Event
- Creating Host Element Bindings
- Creating a Two-Way Binding on the Host Element
- Exporting a Directive for Use in a Template Variable

## **Structural Directives**

- A Simple Attribute Directive
- Applying a Custom Directive
- Accessing Application Data in a Directive
- Reading Host Element Attributes

- Using a Single Host Element Attribute
- Creating Data-Bound Input Properties
- Responding To Input Property Changes
- Binding To a Custom Event
- Creating Host Element Bindings
- Creating a Two-Way Binding on the Host Element
- Exporting a Directive for Use in a Template Variable

## DAY 3

### Components

- Structuring an Application With Components
- Creating New Components
- Defining External Templates
- Using Data Bindings in Component Templates
- Using Input Properties to Coordinate Between Components
- Using Directives in a Child Component Template
- Using Output Properties To Coordinate Between Components
- Projecting Host Element Content
- Completing the Component Restructure

### Pipes

- Installing The Internationalization PolyFill
- Registering a Custom Pipe
- Applying A Custom Pipe
- Combining Pipes
- Creating Impure Pipes
- Formatting Numbers
- Formatting Currency Values
- Formatting Percentages
- Formatting Dates
- Changing String Case
- Serializing Data as JSON
- Slicing Data Arrays

### Services

- The Object Distribution Problem
- Distributing Objects as Services Using Dependency Injection
- Preparing The Service
- Preparing Dependent Components
- Registering the Service
- Reviewing the Dependency Injection Changes
- Declaring a Dependency in a Pipe and in Directives
- Isolating Components Using Services and Dependency Injection
- Preparing and Registering The Service
- Preparing The Dependent Component
- Updating the Root Component
- Updating The Child Components

## **Service Providers**

- Using The Class Provider
- Using Opaque Tokens
- Understanding the useClass property
- Resolving a Dependency with Multiple Objects
- Using the Value Provider
- Using the Factory Provider
- Using The Existing Service Provider
- The Limitations of Single Service Objects
- Creating Local Providers In a Directive
- Creating a Local Provider for All Children of an Object
- Creating a Local Provider for View Children
- Restricting the Provider Search
- Skipping Self-Defined Providers

## **Modules**

- The Imports Property of the Root Module
- The Declarations Property of the Root Module
- The Providers Property of the Root Module
- The Bootstrap Property of the Root Module
- The Model Module Definition



- Other Classes In The Application
- Updating The Root Module
- Creating a Utility Feature Module
- Creating a Feature Module With Components

## DAY 4

### Project

- Configuring Packages
- Model Module
- Core Module
- Messages Module
- The Angular Bootstrap
- The Reactive Extensions Module
- The HTML Document

### Reactive Extensions

- Understanding Observables
- Understanding Observers
- Understanding Subjects
- Using The Async Pipe
- Scaling Up Application Feature Modules
- Filtering Events
- Transforming Events
- Using Different Event Objects
- Receiving Only Distinct Events
- Using a Custom Equality Checker
- Taking and Skipping Events

### Making Asynchronous HTTP Requests

- Understanding RESTful Web Services
- Replacing the Static Data Source Service
- Setting Up the HTTP Request
- Processing the Response
- Configuring the Data Source

- Using The Rest Data Source
- Saving and Deleting Data
- Consolidating HTTP Requests
- Making Cross-Origin Requests
- Using JSONP Requests
- Configuring Request Headers
- Handling Errors
- Generating User-Ready Messages
- Handling the Errors

## **Routing and Navigation – Starting**

- Disabling The State Change Event Display
- Creating A Routing Configuration
- Creating the Routing Component
- Updating The Route Module
- Adding Navigation Links
- Understanding The Effect Of Routing
- Handling Route Changes in Components
- Using Multiple Route Parameters
- Using Optional Route Parameters
- Navigating in Code
- Receiving Navigation Events
- Removing The Event Bindings and Supporting Code

# **DAY 5**

## **Routing and Navigation – Expanding**

- Adding Components to the Project
- Using Wildcards in Routes
- Using Redirections in Routes
- Responding to Ongoing Routing Changes
- Styling Links for Active Routes
- Creating the Child Route Outlet
- Accessing Parameters from Child Routes

## **Routing and Navigation – Guarding**

- Guarding Routes
- Delaying a Resolver Service
- Creating, Registering and Applying a Resolver Service
- Displaying Placeholder Content
- Using a Resolver to Prevent URL Entry Problems
- Preventing Navigation With Guards
- Preventing Route Activation
- Consolidating Child Route Guards
- Preventing Route Deactivation
- Loading Feature Modules Dynamically
- Creating a Route to Dynamically Load a Module
- Using a Dynamically Loaded Module
- Guarding Dynamic Modules with a Loading Guard
- Targeting Named Outlets
- Creating Additional Outlet Elements
- Navigating when Using Multiple Outlets

## **Animation**

- Adding the Animations Polyfill
- Disabling The HTTP Delay
- Simplifying the Table and Routing Configuration
- Defining Style Groups
- Defining Element States
- Defining State Transitions
- Defining The Trigger
- Applying The Animation
- Testing The Animation Effect
- Understanding The Built-in Animation States
- Understanding Element Transitions
- Creating Transitions for Built-in States
- Animating Element Addition and Removal
- Controlling Transition Animations (Timing Function, Initial Delay, Additional Styles, Parallel Animations)
- Animation Style Groups
- Defining Common Styles
- Using Element transformations, CSS Framework Styles,

## Animation Trigger Events

### Unit Testing

- Configuring Karma, TypeScript
- Creating a Simple Unit Test
- Starting the Tools
- Working with Jasmine
- Testing an Angular Component
- Configuring the TestBed for Dependencies
- Testing Data Bindings
- Testing a Component with an External Template
- Testing Component Events
- Testing Output Properties
- Testing Input Properties
- Testing with Asynchronous Operations
- Testing an Angular Directive

---

### Duration and pricing

In [Pricing Group A](#)

### Certificate

- Upon completion of this course we will issue you with attendance certificate to certify your attendance and / or completion of the prescribed minimum examples.
- You may sit for our competency assessment test and on passing you will obtain our competency certificate.
- Our competency assessment can be booked and taken by someone who has not attended the course at a cost of R2950.

### Bookings

You can download the course registration form on our home page or by clicking [here](#)

## **Brochure**

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

## **Questions**

Please [email us](#)

## **Schedule**

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page.