*"Charting the Course ...*

*... to Your Success!"*

# Comprehensive Angular 5

## Course Summary

**Description**

Comprehensive Angular teaches students the skills and best practices they need to design, build, test, and deploy applications that provide rich end-user experiences similar to a desktop application while still offering the ease of deployment of a web application.

**Objectives**

After taking this course, students will be able to:
- Understand how single-page web application architectures (including Angular) are different than traditional web development architectures
- Use new JavaScript (ES6) language features including Classes, Modules, and Arrow Functions
- Use new TypeScript language features including Static Types, Interfaces, and Generics
- Build an application from scratch using Angular 5
- Understand Angular coding and architecture best practices
- Understand and use Angular Model-driven Forms, Observables, Dependency Injection, and Routing
- Communicate with a backend server using Angular's HttpClient service to load and save data.
- Unit test all the parts of an Angular application including Modules, Components, Services, and Pipes
- Upgrade an existing application from AngularJS to Angular 5 over time by running both frameworks in the same project
- Start a new Angular project and scaffold modules, components, services, models, routes, and unit tests following best practices using the Angular CLI
- Build and deploy an Angular application including combining and minifying JavaScript and HTML files, Tree-shaking unused code, and doing Ahead-of-Time compilation to reduce the size of the Angular framework.
- Write End-to-End Tests if your application with Protractor which uses Selenium Web Driver
- Using Redux and NgRx to maintain the state in your application

**Topics**

- Introduction
- Why Angular?
- Understanding Angular Versions
- Upgrading to Angular 5 from Angular 2 or Angular 4
- Angular 5 Features
- TypeScript and ECMAScript 6 (ES6) Fundamentals
- Angular 5 Basics
- Template Syntax
- Components
- Services & Dependency Injection
- Dependency Injection
- Model-driven Forms (Reactive Forms)
- RxJS and Observables
- Communicating with the Server using the HttpClient Service
- Router
- Unit Testing
- Security
- Advanced Components
- Advanced Routing
- Advanced Dependency Injection
- Attribute Directives
- Pipes
- Creating, Building, and Deploying an Angular Application
- Template-driven Forms
- Upgrade Strategies from AngularJS
- Redux
- End-to-End Testing
- npm QuickStart
- Webpack Guide
- Conclusion

# Comprehensive Angular 5

## **Course Summary** (cont'd)

**Audience**

This course is designed for those wanting to learn how to build an application from scratch using Angular 5.

**Prerequisites**

Before taking this course, attendees must have substantial prior experience developing with JavaScript.

**Duration**

Five days

# Comprehensive Angular 5

## Course Outline

I. **Introduction**

II. **Why Angular?**
- A. User Experience similar to a Desktop Application
- B. Productivity and Tooling
- C. Performance
- D. Community
- E. Full-featured Framework
- F. Platform for Targeting Native Mobile not just Web Browsers

III. **Understanding Angular Versions**
- A. AngularJS (Angular 1.x)
- B. Angular
  - 1. Angular 2
  - 2. Angular 5

IV. **Upgrading to Angular 5 from Angular 2 or Angular 4**
- A. Angular Update Guide

V. **Angular 5 Features**
- A. Build Optimizer
- B. Angular Universal State Transfer API and DOM Support
- C. Compiler Improvements
- D. Internationalized Number, Date, and Currency Pipes
- E. Replace the ReflectiveInjector with StaticInjector
- F. Zone Speed Improvements
- G. ExportAs Multiple Names
- H. HttpClient
- I. Angular CLI v1.5
- J. Angular Forms adds updateOn Blur / Submit
- K. RxJS 5.5
- L. New Router Lifecycle Events

VI. **TypeScript and ECMAScript 6 (ES6) Fundamentals**
- A. Classes
- B. ES Modules
- C. Arrow Functions
- D. Template Literals
- E. Scoping using let and const Keywords
- F. Spread Syntax and Rest Parameters
- G. Destructuring
- H. Decorators (JavaScript Aspect-Oriented Programming)

VII. **Angular 5 Basics**
- A. Components
- B. Templates
  - 1. Inline Templates
  - 2. Multi-line Templates using ES6 Template Literals
  - 3. External with Component-relative Paths
- C. Modules
  - 1. Angular Modules vs. ES Modules
- D. Models

VIII. **Template Syntax**
- A. HTML in templates
- B. Interpolation
- C. Binding syntax
- D. Property binding
- E. Event binding
- F. Two-way data binding
- G. Attribute, class, and style bindings
- H. Built-in Directives
  - 1. Built-in attribute directives: NgClass, NgStyle, NgModel
  - 2. Built-in structural directives: NgIf (includes enhanced *ngIf syntax), NgFor
- I. Template Input Variables
- J. The NgSwitch Directives
- K. Template Reference Variables
- L. Input and output properties
- M. Template Expression Operators
- N. Pipe ( | )
- O. Safe Navigation Operator ( ?. )

IX. **Components**
- A. Component Lifecycle Hooks
  - 1. Implementing the OnInit Lifecycle Hook
- B. Component Communication
  - 1. Input properties
  - 2. Output properties: Custom Events using EventEmitters

X. **Services & Dependency Injection**
- A. Using a services to access data
- B. Using a service to encapsulate business logic
- C. Understanding the scope of services

XI. **Dependency Injection**
- A. Understanding Dependency Injection
- B. Angular's Dependency Injection System

# Comprehensive Angular 5

## Course Outline (cont'd)

C. Registering
D. Injecting
E. Hierarchical Injection

**XII. Model-driven Forms (Reactive Forms)**
A. Importing the ReactiveFormsModule
B. FormControl, FormGroup, and AbstractControl
C. Binding DOM Elements to FormGroups and FormControls
D. Validation Rules, Messages, and Styles
E. Refactoring ReactiveForms for Reuse
F. Custom Validators

**XIII. RxJS and Observables**
A. What is an Observable?
B. Observable Operators
C. Creating Observables Using Static Operators
D. What is an Observer?
E. Observer Example
F. Subject
G. Subject Example
H. EventEmitter or Observable

**XIV. Communicating with the Server using the HttpClient Service**
A. Deciding between Promises or Observables (RxJS)
B. Making a HTTP GET Request
C. Sending data to the server using Http POST and PUT Requests
D. Issuing a Http DELETE Request
E. Intercepting Requests and Responses
F. WebSockets

**XV. Router**
A. Importing the RouterModule and Routes
B. Configuring Routes
C. Displaying Components using a RouterOutlet
D. Navigating declaratively with RouterLink
E. Navigating with code using the Router
F. Accessing parameters using ActivatedRoute
G. Organizing your code into Modules

**XVI. Unit Testing**
A. Tools: Jasmine, Karma
B. Jasmine Syntax: describe, it, beforeEach, afterEach, matchers
C. Setup and your First Test
D. Testing Terminology: Mock, Stub, Spy, Fakse
E. Angular Testing Terminology: TestBed, ComponentFixture, debugElement, async, fakeAsync, tick, inject
F. Simple Component Test
G. Detecting Component Changes
H. Testing a Component with properties (inputs) and events (outputs)
I. Testing a Component that uses the Router
J. Testing a Component that depends on a Service using a Spy
K. Testing a Component that depends on a Service using a Fake
L. Testing a Service and Mocking its Http requests
M. Testing a Pipe

**XVII. Security**
A. How to Prevent Cross-site Scripting (XSS)
B. Trusting values with the DOMSanitizer
C. HTTP Attacks
D. Security Audits of Angular Applications

**XVIII. Advanced Components**
A. Component Styles
  1. using MetaData properties: Styles and StyleUrls
  2. Encapsulation Strategies
B. Change Detection Strategies
C. Component Lifecycle Hooks

**XIX. Advanced Routing**
A. Lazy-loading Angular Modules
B. Location Strategies
C. Nested or Child Routes
D. Route Guards

**XX. Advanced Dependency Injection**
A. Providers
B. Using the @Optional and @Host Decorators

**XXI. Attribute Directives**
A. Creating a custom Attribute Directive using ElementRef, Render

**XXII. Pipes**
A. Built-in Pipes: Using, Passing Parameters, Chaining

# Comprehensive Angular 5

## **Course Outline** (cont'd)

B. Creating a custom Pipe using PipeTransform
C. Understanding Pure and Impure Pipes

**XXIII. Creating, Building, and Deploying an Angular Application**
A. Manually
B. Using the Angular CLI
1. Overview
2. Features
3. Installation
4. Generating a New Project
5. Generating Code
6. Builds
7. Customizing Builds
8. Angular Material Setup
9. Eject

**XXIV. Template-driven Forms**
A. NgSubmit Directive
B. FormsModule
C. NgForm, NgModel, and NgModelGroup Directives
D. Validation Directives
1. Displaying validation messages
2. Styling validation messages

**XXV. Upgrade Strategies from AngularJS**
A. Preparing your AngularJS Project
1. Integrating a Module Loader
2. Start using TypeScript
3. Use Components instead of Controllers
B. Angular 5 and AngularJS together
1. Understanding the Upgrade Module
2. Angular (Angular 5) Components in AngularJS Code
3. AngularJS Directives in Angular Code
4. Injecting AngularJS Services into Angular
5. Injecting Angular Services into AngularJS
6. Upgrade from AngularJS Router to Angular Router

**XXVI. Redux**
A. Redux Basics
B. Debugging and Time Traveling with Redux DevTools

**XXVII. End-to-End Testing**
A. What is Protractor?
B. Why Protractor?
C. Using Locators
D. Page Objects
E. Debugging E2E Tests

**XXVIII. npm QuickStart**
A. Installing Dependencies Locally
B. Using npm as a Build Tool

**XXIX. Webpack Guide**
A. Installation
B. Building/Bundling
1. JavaScript
2. CSS
3. HTML
4. Images
C. Development Builds
D. Production Builds

**XXX. Conclusion**