## ­­­Duration:

5 days

## Prerequisites:

This course is explicitly for those who has just started working on Angular projects or complete beginners

## Objective:

After this course participants will be able to

1. Understand features of JavaScript and TypeScript languages, required for coding in Angular
2. Understand what is Client-side framework and Single Page Application (SPA)
3. Understand what Angular is and which features are offered by Angular and why should one choose Angular
4. How to build Angular application using CLI tool and he artifacts of the application
5. What is component and how to create an application based on components
6. What are directives
7. How to pass data between view and code of a component
8. How to use pipe for transformation of data in UI
9. How to write reusable piece of code through service and how the same is dependency injected into components and other assets of angular
10. How to fetch or pass data to RESTful API server using HttpClient through HTTP requests
11. How to use Forms in Angular ad validate the controls
12. How to use client-side routing in Angular
13. How to unit test the Angular components, services etc.
14. Learn about the new features of the latest version

## Content:

### Day-1:

## **JavaScript Basics**

* Introduction to JavaScript
* Understanding script tag
* Variables and data types
* Operators and control flow statements i.e. loops, if-else etc
* Strings and Arrays
* Function declaration
* Function parameters and arguments
* Scopes- local and global
* Function expression and anonymous functions
* Object literals
* Object constructors
* Behavior of ‘this’ keyword in constructors
* Best practices
* Functions in Depth
* Function Objects
* Function Literal
* Function Invocation Patterns
* Recursion & Closures
* Functions as Callbacks

## **JavaScript New Features**

* Classes
* Class Definition
* Class Inheritance
* Base Class Access
* Static Members
* Getter/Setter
* Scoping:
* let keyword: block scoped variables
* const keyword
* Extended Parameter Handling
* Default parameter values
* Rest Parameter
* Spread Operator
* arrow operator
* expression bodies
* statement bodies
* Lexical this
* Template Literals: String interpolation
* Property shorthand
* Destructuring object
* ES6 modules
* Value Export/Import
* Iterators
* Iterator & For-Of Operator
* Map/Set & WeakMap/WeakSet
* Promises
* Promise Usage

## **TypeScript Introduction**

* Why use TypeScript?
* TypeScript Features
* TypeScript Syntax, Keywords, and Code Hierarchy
* Tooling and Framework Options: TypeScript Playground, Visual Studio etc.
* Hello World Example

## **TYPING, VARIABLES, AND FUNCTIONS**

* Overview
* Grammar, Declarations, and Annotations
* Type Inference
* Grammar
* Static and Dynamic Typing
* Compile Time or Run Time
* Ambient Declarations and Type Definition Files
* The Any Type and Primitives
* Applying Types
* Objects
* Functions
* Arrow Functions and Debugging
* Functions and Interfaces
* Static Typing Recap
* Optional and Default Parameters
* Rest Parameters
* Demo: Defining Parameters
* Function Overloads
* Demo: Using Function Overloads

## **Modules**

* Overview
* Identifying a Module
* Creating an Internal Module
* Internal Module Accessibility and IIFE
* Named Modules
* Extending Modules and Importing Shortcuts
* Organizing Internal Modules
* Separating Internal Modules
* External Modules and Dependency Resolution
* Module Dependencies
* Importing External Modules Using AMD
* Importing 3rd Party Libraries Using AMD

## **Generics**

* Introduction and Overview
* What are Generics and Type Parameters?
* Using Array <T>
* Generic Functions
* Demo: Creating and Using Generic Functions
* Generic Interfaces and Classes
* Demo: Creating and Using a Generic Class
* Generic Constraints
* Demo: Applying Constraints to Generics

### Day-2:

## module-1: Introduction to Angular 15

* Introduction
* Anatomy of an Angular 15 Application
* High-Level architectural Overview of Angular 15
  + Code Simplification
  + Performance Improvements
* Sample Application

## module-2: Setup Environment and Package Requirement, Installtion Etc

* Introduction
* Selecting a Language
* Selecting an Editor
* Setting up Our Environment
* Setting up an Angular 15 Application
* Running an Angular 15 Application
* About Modules
* Loading Modules and Hosting our Application

## module-3: Introdduction to Component

* Introduction
* What Is a Component?
* Creating the Component Class
* Defining the Metadata with a Decorator
* Importing What We Need
* Demo: Creating the App Component
* Bootstrapping the App Component
* Demo: Bootstrapping the App Component

## module-4: Templates and Directives

* Introduction
* Building a Template
* Adding Logic with Directives: ngIf
* Adding Logic with Directives: ngFor
* Adding Logic with Directives: ngSwitch
* Using ngStyle and ngClass directives

## module-5: Data and event Binding

* Introduction
* Passing data between component code to view of the component
  + Interpolation
  + Property binding
* Passing data from component vie to code: Event binding
* Handling Input with Two-way Binding

### Day-3:

## module-6: More on Components and pipes

* Component lifecycle
* Important component lifecycle hooks
  + Init
  + Changes
  + Destroy
* Pipes
* Usage of pipes
* Built-in pipes in Angular
* Building Custom Pipes

## module-7: Building Nested Components and sharing data across components

* Introduction
* Building a nested component
* Passing Data to a nested component from parent component Using @Input
* Passing Data from a nested component to parent component using @Output
* Sharing data across components using RxJS Subjects
* Observables and Reactive Extensions
* Publishing data to the Subject
* Subscribing data from the subject

## module-8: Services and Dependency Injection

* Introduction
* How Does It Work?
* Creating a Service
* Registering the Service
* Injecting the Service
* 3 levels of injectors:
  + Component level
  + Feature module level
  + Root module level
* @Injectable() decorator for the service: is it mandatory and best practice
* Creating custom token-based provider using useClass and provide for service
* Inject the service instance using @Inject()

### Day-4:

## module-9: Retrieving Data using Http

* Introduction
* Setting Up
* Sending an Http Request using HttpClient in angular 15
* Importing HttpClientModule
* Different HttpClient methods
* Using generic methods like get<T>() etc. for automatic model mapping
* Model mapping using pipe() method of an observable
* Subscribing to an Observable

## module-10: Navigation and Routing

* Introduction
* Setting Up
* Configuring Routes: Route and Route types
* Creating routes in separate module file
* Registering the routes with RouterModule (forRoot method)
* Using routerLink, router-outlet directives
* Passing Parameters to a Route
* Fetching route data using ActivatedRoute and ActivatedRouteSnapshot
* Navigating a Route with Code using Router class

### Day-5:

## module-11: Working With Forms

* Types of Form in Angular
  + Template-Driven Forms
  + Reactive Forms
* Template driven forms
  + Using NgForm and NgModel directives
  + Validating a template driven form
  + FormsModule
* Reactive forms
  + Directly using Form Control, FrormGroup classes in component code
  + Using FormBuilder service to create forms and controls in code
  + FormGroup, FormGroupName, FormControl, FormControlName directives
  + ReactiveFormsModule
  + Validating Reactive Form
* Custom Validators
  + Custom Validation Directives

## Module-12: Testing Angular applications

* Unit Testing – Jasmine & Karma
* Test a Class
* Test a Component
* Test a Pipe

## Module-13: new features of angular 15

* Standalone Stable APIs
* Directive Composition API
* Improve Stack Traces for Debugging
* Http with provideHttpClient
* Stable Image Directives
* Dependency Injection
* Functional Route Guards