ANGULAR 13 PROGRAMMING (5 DAYS)

By Dr. Vishwanath Rao

Session 1. Introducing Angular

- What is Angular?
- Central Features of the Angular Framework
- Appropriate Use Cases
- Building Blocks of an Angular Application
- Basic Architecture of an Angular Application
- Installing and Using Angular
- Anatomy of an Angular Application
- Running the Application
- Building and Deploying the Application
- Angular for Native Mobile Apps
- Summary

Session 2. Introduction to TypeScript

- Programming Languages for Use with Angular
- TypeScript Syntax
- Programming Editors
- The Type System – Defining Variables
- The Type System – Defining Arrays
- Basic Primitive Types
- Type in Functions
- Type Inference
- Defining Classes
- Class Methods
- Visibility Control
- Class Constructors
- Class Constructors – Alternate Form
- Uninitialized Fields
- Interfaces
- Working with ES6 Modules
- var vs let
- Arrow Functions
- Arrow Function Compact Syntax
- Template Strings
- Generics in Class
- Generics in Function
- Summary

Session 3. Components

- What is a Component?
- An Example Component
- Creating a Component Using Angular CLI
- The Component Class
- The @Component Decorator
- Registering a Component to Its Module
- Component Template
- Example: HelloComponent Template
- Example: The HelloComponent Class
- Using a Component
- Run the Application
- Component Hierarchy
- The Application Root Component
- The Bootstrap File
- Component Lifecycle Hooks
- Example Lifecycle Hooks
- CSS Styles
- Summary

Session 4. Component Templates

- Templates
- Template Location
- The Mustache {{ }} Syntax
- Setting DOM Element Properties
- Setting Element Body Text
- Event Binding
- Expression Event Handler
- Prevent Default Handling
- Attribute Directives
- Apply Styles by Changing CSS Classes
- Example: ngClass
- Applying Styles Directly
- Structural Directives
- Conditionally Execute Template
- Example: ngIf
- Looping Using ngFor
- ngFor Local Variables
- Manipulating the Collection
- Example - Deleting an Item
- Item Tracking with ngFor
- Swapping Elements with ngSwitch
- Grouping Elements

- Template Reference Variable
- Summary

Session 5. Inter Component Communication

- Communication Basics
- The Data Flow Architecture
- Preparing the Child to Receive Data
- Send Data from Parent
- More About Setting Properties
- Firing Event from a Component
- @Output() Example - Child Component
- @Output() Example - Parent Component
- Full Two Way Binding
- Setting up Two Way Data Binding in Parent
- Summary

Session 6. Template Driven Forms

- Template Driven Forms
- Importing Forms Module
- Basic Approach
- Setting Up a Form
- Getting User Input
- Omitting ngForm Attribute
- Initialize the Form
- Two Way Data Binding
- Form Validation
- Angular Validators
- Displaying Validation State Using Classes
- Additional Input Types
- Checkboxes
- Select (Drop Down) Fields
- Rendering Options for Select (Drop Down)
- Date fields
- Radio Buttons
- Summary

Session 7. Reactive Forms

- Reactive Forms Overview
- The Building Blocks
- Import ReactiveFormsModule
- Construct a Form
- Design the Template
- Getting Input Values

- Initializing the Input Fields
- Setting Form Values
- Subscribing to Input Changes
- Validation
- Built-In Validators
- Showing Validation Error
- Custom Validator
- Using a Custom Validator
- Supplying Configuration to Custom Validator
- FormArray - Dynamically Add Inputs
- FormArray - The Component Class
- FormArray - The Template
- FormArray - Values
- Sub FormGroups - Component Class
- Sub FormGroups - HTML Template
- Why Use Sub FormGroups
- Summary

Session 8. Services and Dependency Injection

- What is a Service?
- Creating a Basic Service
- The Service Class
- What is Dependency Injection?
- Injecting a Service Instance
- Injectors
- Injector Hierarchy
- Registering a Service with the Root Injector
- Registering a Service with a Component's Injector
- Register a Service with a Feature Module Injector
- Where to Register a Service?
- Dependency Injection in Other Artifacts
- Providing an Alternate Implementation
- Dependency Injection and @Host
- Dependency Injection and @Optional
- Summary

Session 9. HTTP Client

- The Angular HTTP Client
- Using The HTTP Client - Overview
- Importing HttpClientModule
- Service Using HttpClient
- Making a GET Request
- What does an Observable Object do?

- Using the Service in a Component
- The PeopleService Client Component
- Error Handling
- Customizing the Error Object
- Making a POST Request
- Making a PUT Request
- Making a DELETE Request
- Summary

Session 10. Pipes and Data Formatting

- What are Pipes?
- Built-In Pipes
- Using Pipes in HTML Template
- Chaining Pipes
- Internationalized Pipes (i18n)
- Loading Locale Data
- The date Pipe
- The number Pipe
- Currency Pipe
- Create a Custom Pipe
- Custom Pipe Example
- Using Custom Pipes
- Using a Pipe with ngFor
- A Filter Pipe
- Pipe Category: Pure and Impure
- Pure Pipe Example
- Impure Pipe Example
- Summary

Session 11. Introduction to Single Page Applications

- What is a Single Page Application (SPA)
- Traditional Web Application
- SPA Workflow
- Single Page Application Advantages
- HTML5 History API
- SPA Challenges
- Implementing SPA's Using Angular
- Summary

Session 12. The Angular Component Router

- The Component Router
- View Navigation
- The Angular Router API

- Creating a Router Enabled Application
- Hosting the Routed Components
- Navigation Using Links and Buttons
- Programmatic Navigation
- Passing Route Parameters
- Navigating with Route Parameters
- Obtaining the Route Parameter Values
- Retrieving the Route Parameter Synchronously
- Retrieving a Route Parameter Asynchronously
- Query Parameters
- Supplying Query Parameters
- Retrieving Query Parameters Asynchronously
- Problems with Manual URL entry and Bookmarking
- Summary

Session 13. Advanced HTTP Client

- Request Options
- Returning an HttpResponse Object
- Setting Request Headers
- Creating New Observables
- Creating a Simple Observable
- The Observable Constructor Method
- Observable Operators
- The map and filter Operators
- The flatMap() Operator
- The tap() Operator
- The zip() Combinator
- Caching HTTP Response
- Making Sequential HTTP Calls
- Making Parallel Calls
- Customizing Error Object with catchError()
- Error in Pipeline
- Error Recovery
- Summary

Session 14. Angular Modules

- Why Angular Modules?
- Anatomy of a Module Class
- @NgModule Properties
- Feature Modules
- Example Module Structure
- Create a Domain Module
- Create a Routed/Routing Module Pair

- Create a Service Module
- Creating Common Modules
- Using One Module From Another
- Summary

Session 15. Advanced Routing

- Routing Enabled Feature Module
- Using the Feature Module
- Lazy Loading the Feature Module
- Creating Links for the Feature Module Components
- More About Lazy Loading
- Preloading Modules
- routerLinkActive binding
- Default Route
- Wildcard Route Path
- redirectTo
- Child Routes
- Defining Child Routes
- for Child Routes
- Links for Child Routes
- Navigation Guards
- Creating Guard Implementations
- Using Guards in a Route
- Summary

Session 16. Advanced RxJS

- Observable Creator Functions
- Subscribing to DOM Events
- The interval Function
- forkJoin and zip Function
- The concat Function
- The merge Function
- Recap of Operators
- The debounceTime Operator
- The distinct Operator
- The delay Operator
- The reduce Operator
- The mergeMap Operator
- The switchMap Operator
- The retryWhen Operator
- Basic Example of retryWhen
- Retry for a Finite Time
- Creating a Custom Operator

- Advanced Example Custom Operator
- Intercepting All HTTP Calls
- Example Interceptor Service
- Registering the Interceptor
- Summary

Session 17. Unit Testing Angular Applications

- Unit Testing Angular Artifacts
- Testing Tools
- Typical Testing Steps
- Test Results
- Jasmine Test Suites
- Jasmine Specs (Unit Tests)
- Expectations (Assertions)
- Matchers
- Examples of Using Matchers
- Using the not Property
- Setup and Teardown in Unit Test Suites
- Example of beforeEach and afterEach Functions
- Angular Test Module
- Example Angular Test Module
- Testing a Service
- Injecting a Service Instance
- Test a Synchronous Method
- Test an Asynchronous Method
- Using Mock HTTP Client
- Supplying Canned Response
- Testing a Component
- Component Test Module
- Creating a Component Instance
- The ComponentFixture Class
- Basic Component Tests
- The DebugElement Class
- Simulating User Interaction
- Summary

Session 18. Debugging

- Overview of Angular Debugging
- Viewing TypeScript Code in Debugger
- Using the debugger Keyword
- Debug Logging
- What is Angular DevTools?
- Using Angular DevTools

- Angular DevTools - Component Structure
- Angular DevTools - Change Detection Execution
- Catching Syntax Errors
- Summary